

КНИГА ЗА УЧИТЕЛЯ

ГАЛИНА МОМЧЕВА • ТОДОРКА ГЛУШКОВА • РОСИЦА МАРИНОВА

ИНФОРМАТИКА

8. КЛАС

ИЗДАТЕЛСКА
КЪЩА



Янубис

ИЗДАТЕЛСТВО
БУЛВЕСТ 2000



**КНИГА ЗА УЧИТЕЛЯ
ПО ИНФОРМАТИКА
за 8. клас**

Автори

- © Галина Димитрова Момчева-Гърдева, 2017
- © Тодорка Атанасова Глушкова, 2017
- © Росица Спасова Кузманова-Маринова, 2017

Графичен дизайн

- © Николай Йорданов Пекарев, 2017

Издателства

- © БУЛВЕСТ 2000, 2017
- © ИК Анубис, 2017

ISBN 978-954-18-1057-6

ISBN 978-619-215-100-3

Уважаеми колеги,

Нашият екип разработи уникално предложение за обучение по информатика в 8. клас, което е съобразено с новите държавни образователни стандарти (за съдържание и оценяване) и учебните програми за училища с природо-математически профил.

В настоящата книга искаме да изясним целите, инструментите и аргументите за предложените дейности и задачи, използвани в учебника. Вашият опит, методическа подготовка, особеностите на учениците и наличната база, съчетани с представените от нас ресурси, идеи, задачи и проекти, ще допринесат за създаване на уникално, но и успешно съвременно обучение по информатика.

Уважаеми колеги, изборът на учебник е избор на комплект ресурси и екип за подкрепа, който удовлетворява едновременно потребностите на ученика, на учителя и на родителя. В този смисъл в нашия комплект имате на разположение:

- специализиран сайт (<http://www.programmingatschool.com>) към учебника, разработен и поддържан от авторите, чрез който ще бъдем непрекъснато заедно, реализирайки това обучение;*
- електронни ресурси към учебника на платформата на издателството;*
- книга за учителя;*
- електронен вариант на учебника;*
- екип, който допълва комплекта с необходимото за вас в процеса на употреба на нашия продукт (обучителни семинари, видеоканал, нови продукти към комплекта).*

Във връзка с последното вече разработваме допълнителни примери: тестове, задачи, приложения, които да подпомогнат и улеснят вас и вашите ученици в учебния процес и извън училище.

От авторите

МЕТОДИЧЕСКИ БЕЛЕЖКИ

Уважаеми колеги, в този раздел от книгата за учителя ще представим нашата концепция/методика за обучението по учебния предмет информатика 8. клас, а също така ще направим и някои общи и конкретни методически бележки.

Според учебната програма и Държавните образователни стандарти (ДОС) учебното съдържание по предмета информатика 8. клас включва 4 различни по своя характер основни теми, които отговарят и на различни професии в ИТ бизнеса. Това предопределя както използването на различни методи и средства за обучение, така и необходимост от специфично оценяване. В настоящия текст с методически бележки ще намерите и готови за употреба препоръки за учене, които може да предоставяте на вашите ученици. Във връзка с това формулирането на общовалиден шаблон за всеки урок не би било адекватно.

Съществуват различни подходи/парадигми на обучение по информатика и компютърни науки. Организацията ACM предлага в своя стандарт Computing Curricula няколко подхода за това с какво започва обучението, сред които: първо процедурно програмиране (procedures-first), първо обектно програмиране (objects-first), първо функционално програмиране (functional-first), първо алгоритми (algorithms-first) и първо хардуер (hardware-first). Като обобщение от ACM и други практики и източници и в контекста на ДОС и учебната програма по учебния предмет информатика биха могли да се приложат следните подходи:

- Първо визуално програмиране (Visual Programming First);
- Първо функционално програмиране (Functional Programming First);
- Първо обектно програмиране (Object Programming First);
- Първо разработка на софтуер (Software Development First).

Досега в българското училище беше валиден

подходът procedures-first, който има недостатъка да не може да покаже ефект от наученото и води до липса на мотивация на обучение по предмета и на ученици, и на учители. При стартиране на обучението с обектно-ориентиран подход освен с обектно-ориентирано програмиране се акцентира и на обектно-ориентираното моделиране, което е по-сложна парадигма.

Друг важен методологичен въпрос е каква е целта/ролята/функцията на обучението: селективна или подкрепяща. Подготовката на учениците за състезания има селективна роля, обучението по предмета информатика е задължителен предмет в профилираната подготовка и методиката на обучение трябва да предоставя методи и средства, които да осигуряват подкрепяща роля.

Интересен вариант е въвеждането в програмирането на Java/C# чрез средство като Alice, на което ви обръщаме специално внимание.

Целите на обучението по този предмет са свързани с придобиване на знания и умения по информатика, а не само за конкретен език за програмиране. Аналогично е обучението по информационни технологии, където използването на софтуер е средство за решаване на проблеми с информационни технологии, а не самоцел. Както и в реалната софтуерна практика, езикът за програмиране е средство за решаване на определен проблем от практиката или науката. Ученикът ще знае какво се прави, за да избере професионално развитие.

Въпреки че учебникът е предназначен основно за ученици от класове с природоматематически профил, той може да бъде полезен на всеки, който се интересува от информатика.

За всички предложени от нас задачи са дадени препоръки за употребата им:

- Работа в час (РЧ);
- Домашна работа (ДР);
- Самостоятелна работа (СР).

Известни са различни стилове на преподаване на един учител като:

- Директен – учителят казва на учениците какво да правят;
- Дискусионен – учителят задава въпроси и слуша;
- Делегиращ – дава права/сила на ученика.

Опитният учител използва различни стилове на преподаване в зависимост от ситуацията.

Една от иновациите в учебника е, че в уроците за нови знания е добавена рубриката „Рефлексия“. Както е известно, рефлексията е механизъм за стимулиране на осъзнаването на наученото и развитие на умения за учене и подобряване на социалните умения.

Освен дейности, които подпомагат рефлексията, можем да говорим и за рефлексивни образователни практики. Рефлексивната практика е важен елемент от емоционалната интелигентност и от нагласите за учене през целия живот. Рефлексивни практики са следните дейности самонаблюдение, самооценка, рефлексивен диалог и др.

Основната цел да включим тази рубрика в урока,

е ученикът, да се научи системно да се само анализира, да си прави самооценка какво е научил днес, с какво повече може да се похвали днес.

Рефлексията тук се комбинира с допълнителните дейности, които подкрепят, а именно участието в система за мониторинг по темата и осъзнаване на мястото в развитието, сравнявайки се с учащите не само в класа, но и в страната.

Ситуация, която стимулира рефлексивните способности на обучавания, е например изискване за самооценяване, както и сравняване на резултатите от оценяването на себе си с оценяване от преподавателя и от други обучавани, което е честа практика в електронното обучение.

Класификация на задачите, препоръчани за обучение в настоящия учебен предмет, ще намерите в сайта към учебника.

Независимо от това дали изпълнява индивидуални, по двойки, групови, или екипни задания, ученикът се оценява индивидуално, т.е. получава индивидуална оценка.

Едно графично представяне на видовете уроци е следното:

нови знания и упражнения

практика

обобщение

контрол и оценка

ОСНОВИ НА ИНФОРМАТИКАТА



СРЕДА ЗА ВИЗУАЛНО ПРОГРАМИРАНЕ



ПРОГРАМИРАНЕ



СОФТУЕРЕН ПРОЕКТ



ИНФОРМАТИКА 8.

Разпределение на часовете по видове уроци

Вид уроци по учебна програма	Препоръчително разпределение по учебна програма	Разпределение в учебника	Брой часове в учебника
Нови знания и упражнения	30%	28%	20
Обобщение, контрол и оценка	20%	20%	14 = 5(O) + 9(KO)
Практически дейности	50%	52%	38
	100%	100%	72

Идеята за ученическо портфолио по този предмет е редно да се обясни и с използването на облачни пространства, където може да се публикуват сорсове и проекти github.

Препоръчваме на учителите преди всеки урок да преглеждат сайта за нови ресурси по темата. Ресурсите ще се обновяват непрекъснато, следвайки

развитието на използваните технологии.

Повече методически бележки търсете в секция методика на ресурсния сайт към учебника (<http://www.programmingatschool.com>) и по време на целогодишно провеждани семинари и обучения, организирани от авторите.

РЕСУРСЕН САЙТ КЪМ УЧЕБНИКА (www.programmingatschool.com) ЗА 2. КЛАС (УЧЕБНИК, ТЕТРАДКИ, ЕЛ. УЧЕБНИК, КНИГА ЗА УЧИТЕЛЯ)

Сайтът programmingatschool.com е създаден като неразделна част от учебника „Информатика за 8 клас“. Той е платформа и е насочен едновременно към учениците, учителите, експертите в областта и родители. Пълноценното използване на учебника предполага изпълнение на различни задачи и предизвикателства, които допълнително разширяват и затвърждават знанията по информатика на учениците. Авторите на учебника са създали различни ресурси (документи, ресурсни файлове, сорсове, тестове, справочници, софтуерни проекти, презентации, видеа и др.), които да бъдат използвани по време на учебния процес, подготовката и самоподготовката по информатика.

Сайтът има за цел да предостави на учениците възможност да се обучават и самообучават по начин, който отговаря на техните навици, очаквания и на съвременните тенденции за онлайн обучение. Основна цел на авторите е да предизвикат интереса и да задържат ангажираността на учителите и учениците, като им предоставят нови и допълнителни ресурси и задачи. Това е и една от причините, поради които е избрана web технологията пред предоставянето на CD със създадените ресурси като част от учебника. Учителите и учениците, имащи регистрация в сайта, имат възможност за достъп до различен тип информация. Учениците имат достъп до учебни материали, задачи и тестове, чрез които могат самостоятелно да установят и проследят нивото на усвояване на учебния материал. Учителите освен достъпа до всички ресурси имат на разположение годишно разпределение, методически разработки и възможност да станат част от един нов колектив, обединяващ учителите по информатика от страната.

Една от основните цели, поставени пред сайта, е **превръщането на учениците и учителите от пасивни ползватели на учебните ресурси в активна страна, която успешно да осигурява съвместна работа както помежду им, така и с**

авторите на учебника и привлечени експерти. Всички желаещи могат да създават допълнителни ресурси, които отговарят на отделните теми и уроците в учебника, с което да подпомогнат по-бързото осигуряване на увеличаващо се ресурсно съдържание. Създавайки допълнителни ресурси, авторите им – учениците и учителите, ще бъдат приобщени към колектива, грижещ се за осигуряването на високо ниво на преподаване на информатика в средното и висшето образование в страната.

УЧИТЕЛИТЕ, осигуряващи и създаващи ресурси, предназначени за programmingatschool.com, ще могат да използват сайта като допълнителен канал, легитимиращ и доказващ техните професионални умения. Чрез създадените собствени материали, задачи и методики на преподаване учителите могат да обменят опит и да споделят доказани и работещи практики, с което да бъдат полезни на своите колеги. **Добавянето на ресурси от учителите по информатика ще подпомогне за онлайн позиционирането на активните училища в страната, за които най-добрите ученици, желаещи да изучават информатика, ще се борят. Освен като принос за авторитета на училищата авторите на ресурсите ще градят своята лична репутация сред общността.**

УЧЕНИЦИТЕ, осигуряващи ресурси за programmingatschool.com; имат възможност да проверят уменията си и познанията си по информатика, ще тренират комуникативните си умения и ще предизвикват своята креативност. **Чрез създадените ресурси учениците могат да подпомогнат бъдещото си портфолио от проекти и CV. Чрез създадената от нас платформа ученикът ще види своето ниво спрямо учениците в страната.**

Сайтът дава възможност на потребителите да придобият лични впечатления от авторите, от техния опит и компетентност, като всеки желаещ може да влезе в пряк контакт с тях. За да бъде осигурена връзката между бизнеса и

образованието, в сайта ще бъдат публикувани различни материали и видеа, в които експерти, служители и собственици на ИТ компании ще споделят опита си, впечатления и съвети за успешна кариера, които ще бъдат в полза на учителите и учениците.

Обучението по информатика следва да изгради специфични навици и умения, които да осигурят възможности за развитие на умения за успешна ИТ кариера на учениците. За да поддържаме интереса на учениците, считаме, че е от особено значение те да бъдат адекватно предизвиквани да придобиват нови

знания и да демонстрират уменията и компетентностите си. В сайта ще намирате информация за организирани състезания, в които учениците от цялата страна могат да участват (включително и онлайн), за да проверят нивото и възможностите си. За учителите и учениците ще бъдат организирани уебинари и събития, които допълнително да ги мотивират, обединяват и подпомагат работата им в контекста на непрестанно изменящите се изисквания към работещите в най-бързо развиващата се сфера.

УРОЦИ ПО ТЕМИ

I. ОСНОВИ НА ИНФОРМАТИКАТА

ТЕМА НА УРОКА:

1. ОСНОВИ НА ИНФОРМАТИКАТА

Брой учебни часове: 2

Вид на урока: Нови знания и упражнения (НЗУ)

КОМПЕТЕНТНОСТИ КАТО ОЧАКВАНИ РЕЗУЛТАТИ ОТ ОБУЧЕНИЕТО

1.1. Информатика и компютърни науки

Изброява и описва основните информационни дейности събиране, съхраняване, преработка и разпространение и общата схема на информационните потоци. Описва понятието „информационен процес“ и дава примери на информационни процеси, свързани с решаване на житейски задачи. Различава понятията „информация“ и „данни“. Обяснява и илюстрира с примери връзката между информация и данни. Обяснява и илюстрира с примери същността на дискретното представяне на информацията за трансформирането ѝ в данни.

1.2. Числата и техните представяния

Разширява и обобщава знанията, свързани с числата и техните представяния:

- непозиционни бройни системи;
- същност на позиционните бройни системи;
- формат на числата в десетична, двоична и шестнадесетична бройна система.

Превръща числа от десетична в двоична бройна система и обратно.

Извършва събиране, изваждане и умножение на две числа в двоична бройна система. Дава примери за използване на двоична и шестнадесетична бройна система.

1.3. Алгоритми и езици за програмиране

Дефинира понятието „алгоритъм“ и описва основните му характеристики (результатност, крайност, детерминираност, масовост).

Описва и проследява несложни, линейни и разклонени алгоритми с различни средства. Описва

същността, структурата и разновидностите на цикличните алгоритмични конструкции. Обяснява същността и функционалното предназначение на език за програмиране. Обяснява същността и предназначението на транслятор (интерпретатор, компилатор).

Представя исторически факти, свързани със създаването и развитието на съвременните езици и среди за програмиране.

НОВИ ПОНЯТИЯ

1.1. Информатика и компютърни науки

основни информационни дейности; информация, данни; дискретно представяне на информацията

1.2. Числата и техните представяния

непозиционни бройни системи; позиционни бройни системи; двоична бройна система; шестнадесетична бройна система; експоненциален формат

1.3. Алгоритми и езици за програмиране

алгоритъм; език за програмиране; транслятор

Дидактически материали: учебник/електронен учебник, мултимедиен проектор, компютър, ресурсен сайт към учебника,

Предварителни изисквания, очаквани трудности и предложения за преодоляването им: учебното съдържание в този урок е по-обемно, поради което препоръчваме учениците да се запознаят с идеите и концепциите в него само с основни задачи, а да се упражняват и влизат в детайли в следващия урок, който е за практика. Считаме работата по трите подтеми в този следващият урок за подходяща поради теоретичния характер на някои от подтемите. Така ученикът ще изпълнява в двата часа за нови знания и упражнения различни по вид дейности: задачи за моделиране, изчислителни задачи и представяне на резултати от проучавания.

Междупредметни връзки: ИТ, математика, биология и здравно образование

Ход на урока: Обемът на основни понятия в този урок е значим, поради което препоръчваме в урока за нови знания и упражненията разучаване на понятията чрез примери.

Стр. 10/Задача 1

Диаграма на основните направления в компютърните науки (класификацията на ACM)

За създаването на диаграмата може да се използва класификацията от сайта <http://dl.acm.org/ccs/ccs.cfm> или <http://books.acm.org/subjects>. Използвайки терминологията от двата източника, сме предложили примерна схема. Учениците чертаят проектираната от тях диаграма с познат софтуер.

Задачата може да бъде зададена за самостоятелна работа. В нея се предвижда търсене на информация при указани ключови думи за търсене и дискусия за връзките на информационното общество, умните градове (smart cities) и дигиталните пазари. Дискусия по глобални проблеми на сигурността. Засилват ли се заплахите от ИТ с разширяване на влиянието на ИТ? Каква област от компютърните науки се грижи за това?

Стр.	Зад.	Отговори	РЧ/СР/ДР
12	1	а) 66; б) DCLXXXI; в) CVIII	СР
12	2	31; 266; 11	ДР
14	3	Задачата изисква търсене по изображение (например в Google) и цели предизвикване на интерес към известни личности в историята на компютърните науки.	ДР
14	4	19; 85	ДР
15	5	10010011; 100111110	РЧ
15	6	128	РЧ
15	7	9	РЧ
15	8	8	ДР
15	9	3644; 4405	СР
16	10	8A; 19B	РЧ
17	11	а) 1011000; б) 100000; в) 100000010; г) 100111	РЧ, ДР
17	12	а) 10010; б) 111; в) 11101010; г) 00010110	РЧ, ДР
17	13	а) 10000000000; б) 1100100; в) 110010101011; г) 11101000111001	РЧ, ДР
17	14	а) 11111010; б) 100011000; в) 100011; г) 101000000101	РЧ, ДР

Стр. 18/Задача 14

Етапи на информационен процес	Въпроси
избиране	Каква информация ми е наистина необходима?
презентиране	Как мога да представя информацията?
дефиниране	Какво наистина искам да открия?
оценяване	Какво научих от това?
локализиране	Къде мога да открия информацията, която търся?
организиране	Как мога да използвам тази информация?

Допълнителна информация за учителя

Умения за управление на информационния процес	
умения	въпроси
1. Определяне	Каква е целта? Какво трябва да направя? Кои са ключовите думи и какво е тяхното значение? Кои са предходните знания? Има ли необходимост от допълнителна информация?
2. Локализиране	Какви източници мога да използвам? Кои източници са подходящи? Точни и достоверни ли са източниците?
3. Избор/Подбор	Доколко е подходяща тази информация? Как ще я запиша? Има ли противоречия в информацията от различните източници?
4. Организиране	Има ли достатъчно информация? Трябва ли да използвам цялата информация? Как да подреда информацията логически? Има ли доказателства, от които мога да направя обобщения?
5. Представяне	Как ще представя тази информация? Коя е моята публика? Какъв стил да използвам? Какви възможности за презентиране да използвам спрямо аудиторията?
6. Оценяване	Изпълних ли задачата си? Как оценявам количеството и качеството на информацията спрямо поставената задача? Прегледах ли повторно етапите на процеса, за да се гарантира, че всички аспекти на задачата са изпълнени? Мога ли да идентифицирам стъпките на информационния процес, които могат да бъдат подобрени?

ТЕМА НА УРОКА:

2. ОСНОВИ НА ИНФОРМАТИКАТА. ПРАКТИКА

Брой учебни часове: 2

Вид на урока: Практически дейности (П)

Опорни знания и умения (ОЗУ): информационни дейности; информационен процес; бройни системи; преобразуване на числа (10), (2), (16), (8); аритметични действия с числа в (2); основни характеристики на алгоритми; описание на линейни и разклонени алгоритми

Основна цел на урока: усъвършенстване на уменията за информационен процес, бройни системи, преобразуване на числата и свойства и представяне на алгоритми

Дидактически материали: учебник, мултимедиен проектор, електронен учебник, ресурсен сайт към учебника, компютър

Очаквани трудности и предложения за преодоляването им: различна скорост на изпълнение на задачите от учениците. Проблемът може да се преодолее с използване на допълнителни задачи от отделените за самостоятелна работа.

Междупредметни връзки: ИТ, математика, биология и здравно образование

Ход на урока:

- ИНФОРМАЦИЯ. ИНФОРМАЦИОННИ ДЕЙНОСТИ. ИНФОРМАЦИОНЕН ПРОЦЕС

Стр.	Зад.	Отговори	РЧ/СР/ДР
28	1	Целта на задачата е осъзнаването на различната степен на важност на една информация за всеки от нас.	РЧ
28	2	Задача за групиране на дейности спрямо видовете информационни дейности, които са свързани с тях.	РЧ
28	4	Съответствие на информационни дейности с етапи на информационен процес.	СР

- ЧИСЛАТА И ТЕХНИТЕ ПРЕДСТАВЯНИЯ

Стр.	Зад.	Отговори	РЧ/ДР																																																
29	1	Задачата е от изследователски тип и е подходяща за изпълнение в малки групи, след което да се представят публично резултатите от работата им. http://www.dcode.fr/babylonian-numbers http://www.dcode.fr/mayan-numbers	СР																																																
29	2	Етикетите на стоки показват употребата на римска бройна система.	СР																																																
29	3	Задачата е от изследователски тип и е подходяща за изпълнение по двойки за реализация на кодиране/декодиране на съобщение. В темите по програмиране може да се опише алгоритъмът за кодиране с програма.	ДР																																																
29	4	Това е забавна задача, свързана с научната тема криптоанализ, където даден шифър или код се анализира, за да се открие начин за дешифриране. Тук забелязваме следното: Думата CODE е четирибуквена, а кодът е от 12 символа, много е вероятно да се използват три символа за кодиране на буква от думата. 143 155 144 145 След това забелязваме, че са използвани поредни числа, а в думата са използвани букви, които са поредни в азбуката. Ако ги разположим в таблица и довършим редицата с поредни стойности, забелязваме, че разликата им от началото на азбуката е едно и също число. <table border="1" style="margin-left: 20px;"> <tbody> <tr> <td>141</td><td>142</td><td>143</td><td>144</td><td>145</td><td>146</td><td>147</td><td>148</td><td>149</td><td>150</td><td>151</td><td>152</td><td>153</td><td>154</td><td>155</td><td>156</td> </tr> <tr> <td>О</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td> </tr> <tr> <td>А</td><td>В</td><td>С</td><td>Д</td><td>Е</td><td>Ф</td><td>Г</td><td>Н</td><td>И</td><td>Ј</td><td>К</td><td>Л</td><td>М</td><td>Н</td><td>О</td><td>Р</td> </tr> </tbody> </table>	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	О	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	А	В	С	Д	Е	Ф	Г	Н	И	Ј	К	Л	М	Н	О	Р	РЧ
141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156																																				
О	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15																																				
А	В	С	Д	Е	Ф	Г	Н	И	Ј	К	Л	М	Н	О	Р																																				
29	5	Кирилицата се намира в разширената осембитова ASCII кодова таблица. Кодовете на символи са съответно: 'к' с ASCII код 170 'о' с ASCII код 174 'д' с ASCII код 164 Общо думата заема 3 байта и 24 бита.	РЧ																																																
29	6	20	ДР																																																
29	7	11	ДР																																																
29	8	10000110□; D80□; 336256□; 348F□; 337□	РЧ																																																
29	9	1A; 1B; 1C	СР																																																
29	10	а) [22; 160]	РЧ																																																
29	11	б) г)	РЧ																																																

Стр.	Зад.	Отговори	РЧ/ДР																																																																
30	12	00110010 (2); 1212(3); 302(4); 200(5); 122(6); 101(7), 62□(8) и 20(25)	СР																																																																
30	13	7 A C 17 7A 7C 17C 7AC 7CA 7CC	ДР																																																																
30	14	7 - 1, 10 - 1, 15 - 1, 16 - 2, 26 - 2, 31 - 2, 32 - 2, 255 - 2, 256 - 3	СР																																																																
30	15	1111 - 1, 1010 - 1, 11111111 - 2, 10000010 - 2, 101110 - 2, 10101 - 2, 100 - 1	СР																																																																
30	16	<table border="1"> <tr> <td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td><td>32</td><td>33</td><td>34</td><td>35</td><td>36</td><td>37</td><td>38</td><td>39</td><td>40</td><td>41</td><td>42</td><td>43</td><td>44</td><td>45</td><td>46</td><td>47</td> </tr> <tr> <td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td> </tr> </table>	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47																																				
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15																																				
30	17	01010001	СР																																																																
30	18	<p>С кое от понятията в урок 1 е свързана темата за парадокса с котката на Шрьодингер (Schrödinger cat)?</p> <p>Задачата е подходяща за групова/екипна работа. Добре е да се изпълни за самостоятелна работа и най-успешните да бъдат представени пред класа, а също и изпратени на нас за публикуване в сайта към учебника.</p>	СР РЧ																																																																

Допълнителна информация за учителя

СТИЛОВЕ НА УЧЕНЕ



ТЕМА НА УРОКА:

3. ОСНОВИ НА ИНФОРМАТИКАТА. ОБОБЩЕНИЕ

Брой учебни часове: 1

Вид на урока: Обобщение (О)

Опорни знания и умения: информационни дейности и процеси; числата и техните представяния; бройни системи; алгоритми; видове алгоритми; езици за програмиране

Основна цел на урока: Обобщение на знанията за информационен процес, бройни системи, преобразуване на числата и свойства и представяне на алгоритми

Дидактически материали: учебник, компютър, работни листове, ресурсен сайт към учебника

Очаквани трудности и предложения за преодоляването им: различна скорост на изпълнение на задачите от учениците. Проблемът може да се преодолее с използване на допълнителни задачи от отделените за самостоятелна работа.

Междупредметни връзки: ИТ, математика

Ход на урока:

- **ИНФОРМАЦИЯ. ИНФОРМАЦИОННИ ДЕЙНОСТИ. ИНФОРМАЦИОНЕН ПРОЦЕС**

Въпроси

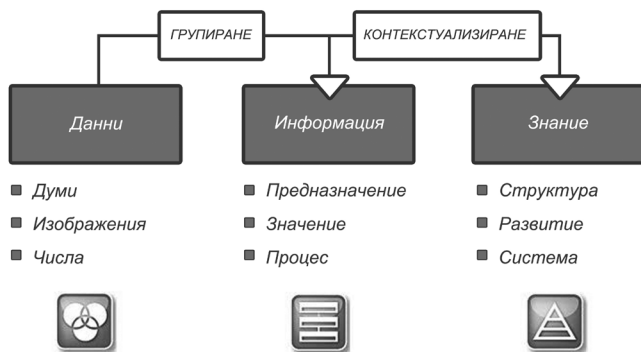
Въпросите служат за затвърждаване на наученото. Отговорите им могат да се открият в урок 1 и урок 2.

Тестови задачи

1 б; 2 а; 3 б

Практически задачи

Задача 1. Учениците могат да потърсят примери, които да представят в следващия учебен час.



Пример

Данните са факти. Например теглото или ръстът на всеки от нас са данни. Публикуваните данни от НСИ са факти, те също са данни.

Да се обърне внимание на учениците, че може да има и неструктурирани данни, които са широко разпространени.

Информацията е някаква новина за нас, някакви нови факти. Промяната на средния ръст или тегло на човек от нашия клас е информация за здравната служба на училището.

Знанието е комбинация от информация, опит и интуиция, която може да донесе полза на индивида или на организация.

• **ЧИСЛАТА И ТЕХНИТЕ ПРЕДСТАВЯНИЯ**

Въпроси

Въпросите служат за затвърждаване на наученото. Отговорите им могат да се открият в урок 1 и урок 2.

Стр.	Зад.	Отговори	РЧ/СР/ДР																																																							
Тестови задачи																																																										
34	1	Всяка позиционна бройна система има основа. Тя задава броя на символите, които се използват за представянето на число. Те са множеството от допустими символи. Ако основата на бройна система е n . Множеството от използваните символи са остатъци при делението на n , т.е. числата са от 0 до $n-1$.	ДР																																																							
34	2	в)	РЧ																																																							
34	3	а)	РЧ																																																							
Изчислителни задачи																																																										
34	1	10100011; 10111011001; 1010101110000011	СР																																																							
34	2	51; 2527; 2695	ДР																																																							
34	3	15 (10) = 17 (8) 173 (8) + 35 (8) = 10011000 (2) 9 (10) = 14 (x), защото $1 \cdot x + 4 \cdot 1 = 9$, $x = 9 - 4$, $x = 5$ 9 (10) = 14 (5)	РЧ																																																							
Практически задачи																																																										
34	1	За по-къса гривна трябва да използваме бройна система с друга основа.	ДР																																																							
35	2	а) 5; б) 8	СР																																																							
35	3	а) Необходими и достатъчни са 7 претегляния: 1, 2, 4, 8, 16, 32, 64. б) Необходими и достатъчни са 5 претегляния: 1, 3, 9, 27, 81. ? Претеглянията са местата за стойностите (бройна система, с основа 2 и 3).	РЧ																																																							
35	4	<table border="1" style="border-collapse: collapse; text-align: center; width: 100%;"> <tr> <td>¹ 1</td><td>0</td><td>0</td><td>0</td><td>² 1</td><td></td><td>³ 1</td><td>1</td><td>0</td><td>0</td><td>⁴ 1</td> </tr> <tr> <td>1</td><td></td><td></td><td></td><td>⁵ 1</td><td>1</td><td>0</td><td></td><td></td><td></td><td>1</td> </tr> <tr> <td>1</td><td></td><td>⁶ 1</td><td>0</td><td>1</td><td>0</td><td>⁷ 0</td><td>0</td><td>⁸ 1</td><td></td><td>1</td> </tr> <tr> <td>1</td><td></td><td>1</td><td></td><td>¹⁰ 1</td><td>1</td><td>1</td><td></td><td>0</td><td></td><td>0</td> </tr> <tr> <td>⁹ 1</td><td>0</td><td>1</td><td>1</td><td>1</td><td></td><td>¹¹ 1</td><td>1</td><td>1</td><td>1</td><td>1</td> </tr> </table>	¹ 1	0	0	0	² 1		³ 1	1	0	0	⁴ 1	1				⁵ 1	1	0				1	1		⁶ 1	0	1	0	⁷ 0	0	⁸ 1		1	1		1		¹⁰ 1	1	1		0		0	⁹ 1	0	1	1	1		¹¹ 1	1	1	1	1	ДР
¹ 1	0	0	0	² 1		³ 1	1	0	0	⁴ 1																																																
1				⁵ 1	1	0				1																																																
1		⁶ 1	0	1	0	⁷ 0	0	⁸ 1		1																																																
1		1		¹⁰ 1	1	1		0		0																																																
⁹ 1	0	1	1	1		¹¹ 1	1	1	1	1																																																

• **АЛГОРИТМИ И ЕЗИЦИ ЗА ПРОГРАМИРАНЕ**

Въпроси

Въпросите служат за затвърждаване на наученото.

Тестови задачи

1. б; 2. в; 3. б

Практически задачи

Зад. 2. Варианти на изпълнение на задачата са:

1. По дадена диаграма да се изработи с подходящ софтуер.
2. По няколко дадени диаграми по темата да се направи обобщение, което да отразява визуално обединяването на концепциите в отделните диаграми.
3. Ученикът създава диаграма с подходящ софтуер, като използва информация от учебника и допълнителните ресурси в сайта.

ТЕМА НА УРОКА:

4. ОСНОВИ НА ИНФОРМАТИКАТА. КОНТРОЛ И ОЦЕНКА

Брой учебни часове: 1

Вид на урока: Контрол и оценка (КО)

Опорни знания и умения: информационни дейности и процеси; числата и техните представяния; бройни системи; алгоритми; видове алгоритми; езици за програмиране

Основна цел на урока: контрол и оценка на знанията и уменията на учениците

Дидактически материали: работни листове; приложението Calculator, ресурсен сайт към учебника

Дейности за контрол и оценка: Проверката се реализира чрез въпроси и задачи със затворен и отворен отговор. Подборът на тестовите задачи е съобразен с формулираните в учебната програма очаквани резултати. Тестовата проверка дава възможност да се обхване по-голям обем от учебното съдържание за по-кратко време. Практическите задания се решават на работните листове.

Верни отговори на въпросите от теста:

Въпрос 1: в

Въпрос 2: б

Въпрос 3: б

Въпрос 4: а

Въпрос 5: в

Задача 1: DCLXXXI

Задача 2: б – преобразувани в десетична бройна система, всяко следващо число е с 3 повече от предходното.

Задача 3: 100001(2)

Задача 4: б

Задача 5: 69(10)

Задача 6: в

Задача 7: *Който се учи, ще сполучи!*

Задача 8: Намиране и отпечатване на абсолютната стойност на числото.

Практическа задача 1: Среща утре в кафето

Практическа задача 2:

Словесно:

1. Въведи а.

2. Въведи б.

3. Въведи с.

4. Ако $a > b$, премини към б.

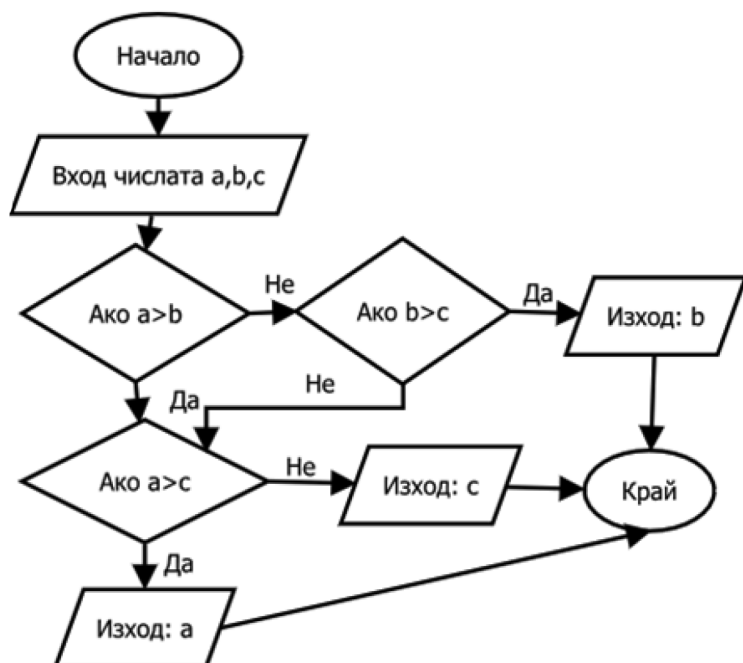
5. Ако $b > c$, отпечатай „Най-голямото число е б“ и премини към 8.

6. Ако $a > c$, отпечатай „Най-голямото число е а“ и премини към 8.

7. Отпечатай „Най-голямото число е с“.

8. Край

Блок-схема:



Оценяване																
Въпроси	1	2	3	4	5										Общо	
точки	1	1	1	1	1										5	
Задачи						1	2	3	4	5	6	7	8			
точки						1	2	1	2	2	2	3	2		15	
Практическа задача														1	2	
точки														10	10	20
скала	33 – 40 точки					Справям се отлично.										
	25 – 32 точки					Имам незначителни пропуски.										
	17 – 24 точки					Мога и по-добре.										
	10 – 16 точки					Трябва да положа още усилия.										
	0 – 9 точки					Не се справям.										

II. СРЕДА ЗА ВИЗУАЛНО ПРОГРАМИРАНЕ

ТЕМА НА УРОКА:

5. СРЕДА ЗА ВИЗУАЛНО ПРОГРАМИРАНЕ

Брой учебни часове: 2

Вид на урока: Нови знания и упражнения (НЗУ)

КОМПЕТЕНТНОСТИ КАТО ОЧАКВАНИ РЕЗУЛТАТИ ОТ ОБУЧЕНИЕТО

Отваря проект в интегрирана среда за визуално програмиране.

Редактира дизайна на графичния потребителски интерфейс на приложение в интегрирана среда за програмиране.

Запазва проект на приложение чрез средствата на интегрирана среда за програмиране.

Стартира приложение с графичен потребителски интерфейс чрез средствата на интегрирана среда за програмиране.

Разпознава основни компоненти на интегрирана среда за програмиране – графичен и текстов редактор, панел с контроли, панел за свойства на обект, панел за съобщения, панел за преглед на структурата на приложението.

Дидактически материали: учебник/електронен учебник, мултимедиен проектор, компютър, ресурсен сайт към учебника

НОВИ ПОНЯТИЯ

интегрирана среда за програмиране; свързващ редактор (linker); програма за откриване и отстраняване на грешки (debugger); редактор за проектиране на дизайн на графичен потребителски интерфейс

Междупредметни връзки: математика, английски език

Предварителни изисквания, очаквани трудности и предложения за преодоляването им: необходимо е предварително да са инсталирани средите за разработка в компютърните зали. На ученика е предоставена информация за процеса на инсталация, който е качен в сайта към учебника, и като последователност от екранни снимки, и като видео.

Ход на урока:

Представяне на различни онлайн среди за програмиране с материали от ресурсния сайт или други, подбрани или подготвени от учителя. Материалите в сайта ще се актуализират при обновяване на версиите на средите, както и при възникване на нови среди. В ресурсния сайт ще се публикува информация и за употребата на облачни среди за разработка на приложения с цел обща информация за ученика и за учителя.

Създаден е пример Triangle, който трябва да се отвори в средата за програмиране и да се стартира.

Проследяват се основни етапи на създаване и изпълнение на програма чрез дадения пример. Създават се тестови случаи и се проверява за различни по вид грешки.

Всички задачи се изпълняват заедно, за да може да се дискутира.

Допълнителни въпроси, задачи и упражнения:

Разучаване на облачните среди за програмиране. Изследване на техните възможности.

ТЕМА НА УРОКА:

6. ИНТЕГРИРАНА СРЕДА ЗА ВИЗУАЛНО ПРОГРАМИРАНЕ. ПРАКТИКА

Брой учебни часове: 2

Вид на урока: Практически дейности (П)

Опорни знания и умения: интегрирана среда за програмиране, редактор за проектиране на дизайн на графичен потребителски интерфейс

Основна цел на урока: Редактира дизайна на графичния потребителски интерфейс на приложение в интегрирана среда за програмиране.

Запазва проект на приложение чрез средствата на интегрирана среда за програмиране.

Дидактически материали: учебник/електронен учебник, компютър, ресурсен сайт към учебника

Очаквани трудности и предложения за преодоляване: различна скорост на работа на ученици, предлага се изпълнение на допълнителни задачи при необходимост.

Междупредметни връзки: химия и опазване на околната среда

Ход на урока:

По зададено приложение, свързано с молекулната маса на веществата, са описани първо сценарии за редактиране на приложението с цел промяна на свойствата, след което се стартира, съхранява и тества приложението.

Дейности за контрол и оценка: не се предвиждат.

Допълнителни въпроси, задачи и упражнения:

Допълнителни ресурси търсете в ресурсния сайт към учебника.

ТЕМА НА УРОКА:

7. ОСНОВНИ ЕТАПИ НА СЪЗДАВАНЕ И ИЗПЪЛНЕНИЕ НА КОМПЮТЪРНА ПРОГРАМА. ПРАКТИКА

Брой учебни часове: 2

Вид на урока: Практически дейности (II)

Опорни знания и умения: интегрирана среда за програмиране, редактор за проектиране на дизайн на графичен потребителски интерфейс, синтактични грешки в компютърна програма, логически грешки в компютърна програма, грешки по време на изпълнение на програмата

Основна цел на урока: Анализира задача с несложен математически модел. Създава математически модел за решаване на несложна задача.

Стартира чрез средствата на интегрирана среда предварително подготвена компютърна програма с графичен потребителски интерфейс.

Тества предварително подготвен несложен проект.

– Разпознава видовете грешки при програмиране.

– Разчита и прави предположение за естеството на синтактична грешка в даден проект.

– Открива и прави предположение за причината за логическа грешка в дадено приложение.

– Открива и прави предположение за причината за грешка по време на изпълнение на приложение.

Дидактически материали: учебник/електронен учебник, компютър, ресурсен сайт към учебника

Очаквани трудности и предложения за преодоляване: различна скорост на работа на ученици, предлага се изпълнение на допълнителни задачи при необходимост.

Междупредметни връзки: математика, БЕЛ, английски език

Ход на урока:

В урока се изпълняват дейности по реализация на етапи на създаване и изпълнение на компютърна програма с проект Grapez. В сайта към ученика е публикуван проект и на Java. Етапите, през които разясняваме условието на задачата, създаваме математическия модел и алгоритъм по него, проверяваме с тестове дали моделът работи вярно.

Следват задачи, в които учениците отварят създадено приложение, след което го тестват с дадени примери.

Провежда се дискусия за проблеми с тестовете и се дава задача на учениците да направят сами тестове.

Дейности за контрол и оценка: не се предвиждат.

Допълнителни въпроси, задачи и упражнения:

В сайта към учебника непрекъснато ще бъдат допълвани проекти, с които трябва да се решат подобни задания, с цел ученикът да има възможност да изпълнява задачи за домашна работа.

ТЕМА НА УРОКА:

8. ПРОЕКТИРАНЕ НА ГРАФИЧЕН ПОТРЕБИТЕЛСКИ ИНТЕРФЕЙС

Брой учебни часове: 2

Вид на урока: Нови знания и упражнения (НЗУ)

КОМПЕТЕНТНОСТИ КАТО ОЧАКВАНИ РЕЗУЛТАТИ ОТ ОБУЧЕНИЕТО

Знае предназначението на основни контейнери и контроли – форма, етикет, текстово поле, бутон, диалогова кутия.

Разпознава основни свойства на графични обекти контроли – име, състояние, етикет, фон, настройка на шрифт и др.

Проектира несложна форма, съдържаща етикет, текстово поле, бутон.

Настройва основни свойства на форма, етикет, текстово поле и бутон.

Именува обекти контроли съгласно общоприетата конвенция.

Задава функционалност на бутон, свързана с извеждането на статично съобщение в диалогова кутия.

НОВИ ПОНЯТИЯ

графичен потребителски интерфейс; интерфейсен компонент (контрола); контейнер на контроли; свойство на обект; метод на обект

Дидактически материали: учебник/електронен учебник, мултимедиен проектор, компютър, ресурсен сайт към учебника

Очаквани трудности и предложения за преодоляване: различна скорост на работа на ученици, предлага се изпълнение на допълнителни задачи при необходимост.

Междупредметни връзки: ИТ, БЕЛ, английски език

Ход на урока:

Изяснява се понятието „потребителски интерфейс“ с беседа.

Предлагаме за дискусия темата за еволюция на потребителските интерфейси.

Обясняват се елементите на графичния потребителски интерфейс, за разбирането на които препоръчваме задача за откриването им в съществуващи приложения.

Представени са принципите за проектиране на потребителски интерфейси и в серия от задачи са дадени примерни интерфейси, бланки за оценка на интерфейси с цел придобиване на практически умения по темата.

Важен елемент от обучението е осъзнаването на целта на употреба на всяка контрола.

Ученикът трябва да може да определя в какво задание кои контроли са подходящи и необходими за употреба.

Интересно е да отбележим, че ученикът може да упражнява тази тема без да трябва да знае всеки детайл от реализацията на приложението с използване впоследствие на тази контрола.

ТЕМА НА УРОКА:

9. ПРОЕКТИРАНЕ НА ГРАФИЧЕН ПОТРЕБИТЕЛСКИ ИНТЕРФЕЙС. ПРАКТИКА

Брой учебни часове: 2

Вид на урока: Практически дейности (П)

Опорни знания и умения: графичен потребителски интерфейс; интерфейсен компонент (контрола); контейнер на контроли; свойство на обект; метод на обект

Основна цел на урока: Отваря проект в интегрирана среда за визуално програмиране.

Настройва основни свойства на форма, етикет, текстово поле и бутон.

Проектира потребителски интерфейс на приложение.

Дидактически материали: учебник/електронен учебник, компютър, ресурсен сайт към учебника

Очаквани трудности и предложения за преодоляването им: различна скорост на работа на учениците, предлага се изпълнение на допълнителни задачи при необходимост

Междупредметни връзки: ИТ, БЕЛ, английски език, технологии и предприемачество

Ход на урока:

В този урок за практически дейности са предложени задачи за определяне на свойствата на обекти контроли. В някои от задачите ученикът

извлича стойностите на свойствата от реални приложения, в други създава контроли с определени свойства, в трети модифицира свойства.

С цел подобряване на стила на оформление са предложени обяснения за подравняването на контроли, което е аналогично с подравняването на векторни обекти в слайд от презентация.

Последната задача от урока е за самостоятелно проектиране на графичния потребителски интерфейс на приложение.

Дейности за контрол и оценка: урокът позволява да се провеждат дейности по оценяване на ученика.

ТЕМА НА УРОКА:

10. ПРОЕКТИРАНЕ И ОЦЕНКА НА ПОТРЕБИТЕЛСКИ ИНТЕРФЕЙСИ. ПРАКТИКА

Брой учебни часове: 2

Вид на урока: Практически дейности (П)

Опорни знания и умения: създаване, редактиране и откриване на грешки при проектиране на потребителски интерфейс; оценява потребителски интерфейс на приложение

Основна цел на урока: Усъвършенстване на умения при създаване, редактиране и откриване на грешки при проектиране на потребителски интерфейс и тестване на приложение за грешки.

Дидактически материали: учебник/електронен учебник, мултимедиен проектор, компютър, ресурсен сайт към учебника

Очаквани трудности и предложения за преодоляване: различна скорост на работа на ученици, предлага се изпълнение на допълнителни задачи при необходимост.

Междупредметни връзки: ИТ, БЕЛ, английски език

Ход на урока:

В урока се предлагат за изпълнение практически задачи, в които се оценява потребителският интерфейс на приложения със зададени шаблони за оценка. Препоръчва се ученикът да на-

прави проучване за видовете и целите на употреба на eye-tracking устройства.

Дейности за контрол и оценка: подходящи за оценка са практически и индивидуални, по двойки и екипни задачи.

ТЕМА НА УРОКА:

11. ПРОЕКТИРАНЕ НА ГРАФИЧЕН ПОТРЕБИТЕЛСКИ ИНТЕРФЕЙС. КОНТРОЛ И ОЦЕНКА

Брой учебни часове: 1

Опорни знания и умения: интегрирана среда за визуално програмиране; основни етапи на създаване и изпълнение на компютърна програма; проектиране на графичен потребителски интерфейс

Основна цел на урока:

Контрол и оценка на знанията и уменията на учениците, свързани със среди за визуално програмиране.

Дидактически материали:

- работни листове
- компютри с интернет достъп

Дейности за контрол и оценка:

- Въпросите са от затворен тип с един верен отговор и са свързани със знанията на учениците върху разглежданата тема.
- Задачите са със свободен отговор и са насочени към проверка на уменията на учениците да прилагат получените знания. За решаването им е необходим компютър с интернет достъп и възможност да ползват ресурсите от сайта към учебника.
- Решаването на практическата задача се реализира на компютър в час, като учениците могат да използват визуална среда за програмиране по техен избор.

Верни отговори на въпросите от теста:

Въпрос 1: б

Въпрос 2: в

Въпрос 3: б

Въпрос 4: в

- Въпрос 5: а, в
 Въпрос 6: а
 Въпрос 7: а, в
 Въпрос 8: г
 Въпрос 9: г
 Въпрос 10: а

Задача 1. Примерен интерфейс с три раздела – Разписание; Местоположение и Пътуване.

– При избор на раздел „Разписание“ се очаква избор на автобусна линия от списъчна кутия и се визуализира файл с разписанието.

– При избор на „Местоположение“ се очаква избор на линия от списъчна кутия и се визуализира карта на маршрут на тази линия и местоположението на всички автобуси по нея.

– При избор на „Пътуване“ се очаква избор на линия и спирка от списъчни кутии и се визуализира търсената информация.

Задача 2. Дадено е приложението Судоку и протокол за оценка на интерфейса му.

Задача 3. Примерно приложение: Избор на клас и ден от седмицата от списъчни кутии и визуализиране на съответната учебна програма, съдържаща учебния предмет и кабинета.

Задача 4. Презентацията се сваля на ученическият компютър. На трети и четвърти слайд контролите се подреждат според изискванията. Презентацията може да се представи пред класа.

Практическа задача – Примерен интерфейс
 – Три етикета, които да визуализират път, скорост и време.

– Под тях етикет за изгорените калории и етикет за необходимата консумация на вода.

– В горния десен ъгъл бутон с високоговорител.

– Карта, която да показва маршрута.

Оценяване																
Въпроси	1	2	3	4	5	6	7	8	9	10					Общо	
точки	1	1	1	1	1	1	1	1	1	1					10	
Задачи											1	2	3	4		
точки											4	3	4	4	15	
Практическа задача														1		
точки														15	20	
скала	33 – 40 точки					Справям се отлично.										
	25 – 32 точки					Имам незначителни пропуски.										
	17 – 24 точки					Мога и по-добре.										
	10 – 16 точки					Трябва да положа още усилия.										
	0 – 9 точки					Не се справям.										

III. ПРОГРАМИРАНЕ

ТЕМА НА УРОКА:

12. ОСНОВНИ ТИПОВЕ ДАННИ

Брой учебни часове: 2

Вид на урока: Нови знания и упражнения (НЗУ)

КОМПЕТЕНТНОСТИ КАТО ОЧАКВАНИ РЕЗУЛТАТИ ОТ ОБУЧЕНИЕТО

основни типове данни и операции с тях (низ (стринг), цели числа, реални числа)

НОВИ ПОНЯТИЯ

константа, променлива, символ, низ, множество на допустимите данни, множество на допустимите операции, име, тип стойност на променлива, име, тип и стойност на константа, присвояване на стойност, конкатенация, целочислен тип данни, конвенция за именуване на константи и променливи; реален тип данни; аритметичен израз в език за програмиране; вградени математически функции в език за програмиране; съвместимост на типове данни; форматиран изход

Дидактически материали: учебник/електронен учебник, мултимедиен проектор, компютър, интернет, ресурсен сайт към учебника

Междупредметни връзки: химия и опазване на околната среда, физика и астрономия, математика, информационни технологии, български език и литература, чужди езици

Ход на урока:

Разглеждат се различни типове езици за програмиране, които най-общо са разделени на типизирани и нетипизирани. При първите явно декларираме и типа ѝ, а при вторите типът се подразбира от стойност, която сме задали.

В урока се изяснява понятието „константа“ като стойност, която не се променя, и се прави връзка на константите в програмирането с константите по други учебни предмети от природните науки като физика, химия и математика.

Препоръчваме в обучението да се насърчават учениците да използват средствата на самия език за откриване на специфична информация като гранични стойности на съответните типове данни и др. (например документацията към езика).

Недопустимо е такъв тип информация да участва в изпитни и оценъчни материали, в които се изисква възпроизвеждане на верен отговор. (Пример: Кое е най-голямото число, което може да се запише в съответен тип?) Такъв тип задача може да бъде дадена, ако на ученика се даде възможност да си извлече тази информация чрез самия език за програмиране или чрез документацията към него.

Следва въвеждането на понятието „променлива“, и основните ѝ характеристики: име и стойност. Изяснява се какво се случва в паметта на компютъра при деклариране на променлива, при присвояване на стойности, а също и при присвояване на нова стойност на променлива. Още в тази тема се изяснява смисълът на понятията „синтаксис“ и „семантика“ като важни в обучението по програмиране и се представя връзката между езиците за програмиране и чуждите езици (естествените езици). Кой език е по-ясен, програмният или естественият, и защо?

Със значението на понятието „грешка“ ученикът вече се е запознал по времето на тестването на готово приложение в уроците в първи раздел и какво съобщение показва средата. Умението за разчитане на проблема: „При какви и грешки какви съобщения извежда средата, в която работим“, е важно с цел при настъпването им практиката да сме готови и да знаем как да реагираме. В урока е даден един пример на Java.

При правилата за именуване на константи и променливи са посочени правила, свързани с избора на символи, правила за употребата, чувствителност към регистъра, роля на специалните символи. Обяснени са употребата на коментари и тяхната роля в програмирането.

Относно символите са изяснени съхраняването на символи и използването на кодови таблици.

В учебника се въвежда понятието „стринг“ по нов начин за учебната литература по програмиране, а именно връзката му с получаването на вход в приложението чрез контроли като текстово поле за разлика от типичното въвеждане след темата масиви като масив от символи. В използваните от нас езици за програмиране стрингът

е обект. Достъпът до съответните стойности в текстовите полета се извършва чрез използването на методи за достъп.

Понятията за обект, метод, функция и клас умишлено не са изясняват теоретично, а се употребяват само на емпирично ниво. Те се използват при редактиране на свойства на контроли (компоненти/елементи от потребителския интерфейс).

Допустимите операции над стрингове са определяне на дължината, задаване на стойност, слепване на стрингове, сравняване, търсене, преобразуване на текст в число и обратно.

Направена е връзка на изчисляването при обучението по програмиране със съответните изчисления с използване на табличен редактор поради факта, че учениците имат опит от употребата на табличен редактор.

Приоритетно в урока се изясняват правила за именуване на константи и променливи и се разучават основните действия със стрингове, целочислени типове и реални типове. При числовите типове се разглежда използването на абсолютна стойност, повдигане на степен и определянето на минимум и максимум, както и извличането на цяла част от реално число. В ресурсния сайт и предложените тук задачи се решават със зададените функции и се прави сравнение с реализация с алгоритъм за изчисление и използване на функции от библиотеки.

Рубриката „Рефлексия“ и към този урок е дадена в сайта поради големия списък на новите понятия. Те могат да бъдат персонализирани, като се селектира подмножество от тях, което да се използва.

В този урок е подходящо за проверката на различни резултати от изрази да се използват онлайн среди за програмиране. Така се осигурява възможност бързо да се стартира всеки резултат и да се проверят конкретното му действие и неговата вярност.

Очаквани трудности и предложения за преодоляване: една такава трудност е обемът на тези обяснения в урока. Тя се преодолява с разглеждане на ключови примери с онлайн среда за програмиране и директно разглеждане на резултата от изпълнението. Друга трудност е непрекъснатото рефериране към учебното съдържание в целия раздел по време на обучението.

ТЕМА НА УРОКА:

13. ОСНОВНИ ТИПОВЕ ДАННИ.

ПРАКТИКА

Брой учебни часове: 2

Вид на урока: Практически дейности (II)

Опорни знания и умения: компоненти на графичен потребителски интерфейс, текстови полета, променливи, типове данни, моделиране на изчисления, откриване на грешки, тестване и създаване на тестове

Основна цел на урока: Усъвършенстване на уменията за използване на елементи на потребителски интерфейс и работа със стрингове.

Дидактически материали: учебник, мултимедия проектор, електронен учебник, ресурсен сайт към учебника, компютър

Ход на урока:

В урока се създава приложение, в което освен на проектирането на потребителския интерфейс е заложено проектиране и на логиката му. Определя се какви променливи се използват за описание на алгоритъма на това приложение, а също се създават и тестове за проверка за грешки и тяхното отстраняване. Предложени са интересни варианти на това приложение, които могат да бъдат зададени за самостоятелна работа.

За предложеното комбиниране на изчисление в една задача с изчисление и получаване на резултата в друга бройна система е предложена реализация с използване на метод, с който се получава преобразуването, без да се налага да се изчислява. Считаме за подходящо ученикът да види и двата варианта.

Дейности за контрол и оценка: този урок не е подходящ за оценяване.

Очаквани трудности и предложения за преодоляването им: различна скорост на изпълнение на задачите от учениците. Проблемът може да се преодолее с използване на допълнителни задачи от отделените за самостоятелна работа.

Междупредметни връзки: математика; ИТ; технологии и предприемачество

ТЕМА НА УРОКА:

14. ОСНОВНИ ТИПОВЕ ДАННИ.

ОБОБЩЕНИЕ

Брой учебни часове: 1

Вид на урока: Обобщение (О)

Опорни знания и умения: основни типове данни; определяне на подходящ тип данни за определена задача

Основна цел на урока: обобщение на знанията за именуване на константи и променливи, деклариране и инициализиране на променливи, вградени функции за преобразуване на типове.

Дидактически материали: учебник, компютър, работни листове, ресурсен сайт към учебника

Очаквани трудности и предложения за преодоляването им: различна скорост на изпълнение на задачите от учениците. Проблемът може да се преодолее с използване на допълнителни задачи от отделените за самостоятелна работа.

Междупредметни връзки: ИТ, математика, физика и астрономия

Ход на урока:

ИМЕНУВАНЕ НА КОНСТАНТИ И ПРОМЕНЛИВИ

Задачите служат за затвърждаване на знанията за правилата за именуване на променливи и константи. Отговорите им могат да се открият в предходните уроци от част III.

ТИПОВЕ ДАННИ (ЧИСЛА И НИЗОВЕ)

Задача 4 изисква попълването на таблицата.

Задача 5 проверява знанията за декларирането на типове данни и откриването на грешки.

Задача 6 – трябва да се открият допуснатите грешки в кода на съответния програмен език.

Задача 7 проверява знанията на учениците за избор на тип данни, съответстващ на зададените стойности.

ВГРАДЕНИ ФУНКЦИИ

Преобразуване на числа в стрингове (низове)

Задача 8. Попълване на таблицата с методите и функциите `toString()`, `ToString()`, `str()` за различните програмни езици. Отговорите могат да се намерят в предходните уроци.

Задача 9. Откриване на грешките в програм-

ни фрагменти на различни езици, свързани с преобразуване на стрингове в числа.

МАТЕМАТИЧЕСКИ ФУНКЦИИ

Задача 10 се решава в текстов редактор, като се очаква от учениците да систематизират предоставената информация за математически функции. Може да се остави за самостоятелна работа вкъщи.

Задача 11, 12 и 13 имат практическа насоченост. Чрез тях трябва да се запишат математически изрази чрез вградените функции и аритметичните операции.

ВЪВЕЖДАНЕ И ИЗВЕЖДАНЕ НА ДАННИ

За решаването на задача 12 се използват ресурсите в сайта към учебника. Необходимо е да се редактира таблицата и да се отстранят грешките.

Задача 15 може да се остави за домашна работа. Необходимо е учениците да създадат диаграма на използваните начини за вход и изход и да я представят на съучениците си за обсъждане.

Задача 16. В тази задача отново се работи с ресурсите от сайта към учебника. Може да бъде поставена за домашна или за самостоятелна работа.

ФОРМАТИРАН ИЗХОД

Разглеждат се начините за форматиране на изхода при текстове, при дати, време и парични знаци, при форматиране на цели и реални числа. Разглеждат се примерите в дадената таблица.

ЗАДАЧИ ЗА СЪЗДАВАНЕ НА ПРОГРАМА

Задача 17 цели да провери практическите умения на учениците за създаване на програмни фрагменти. Може да се използват ресурсите от сайта към учебника.

Задача 18 е насочена към затвърждаване на знанията и уменията за именуване на константи и променливи.

Задача 19 дава възможност за придобиване на практически умения за форматиране на изхода като стринг.

ЗАДАЧИ ЗА СЪЗДАВАНЕ НА ПРИЛОЖЕНИЕ

Задачи 20, 21, 22 и 23 решават несложни алгоритмични проблеми, но чрез тях се проверяват уменията за практическо приложение на получените знания. Решават се в избраната среда за програмиране, като се изисква и създаването на решение с графичен потребителски интерфейс. Някои от задачите могат да се предложат на учениците за самостоятелна работа.

ТЕМА НА УРОКА:

15. ОСНОВНИ ТИПОВЕ ДАННИ.

КОНТРОЛ И ОЦЕНКА

Брой учебни часове: 1

Опорни знания и умения:

- основни типове данни;
- именуване на константи и променливи;
- форматиране на изходните данни;
- аритметични изрази и вградени функции.

Приоритет на операциите

Основна цел на урока:

Контрол и оценка на знанията и уменията на учениците при работа с основните типове данни.

Дидактически материали:

- работни листове
- компютри с интернет достъп

Дейности за контрол и оценка:

- Тестовите **въпроси** са със свободен отговор.

Чрез тях се проверява нивото на знанията и разбиранията на учениците по темата.

• **Задачите** се изпълняват на компютър с интернет достъп за всеки ученик.

• **Практическата задача** се решава в часа на ученическия компютър в програмна среда по негов избор.

Верни отговори на въпросите от теста:

Въпрос 1. По отношение на типовете данни езиците за програмиране се делят на типизирани и нетипизирани. Данните в типизираните езици са от определен тип. Java и C# са типизирани езици.

Въпрос 2. Реален тип без знак, форматиран до два знака след десетичната запетая.

Въпрос 3. Синтактични – при присвояване на променливи от един тип на данни от друг тип; използване като параметри на функции данни от друг тип и т.н. Логически – при въвеждане и обработка на стойности от типове, които в реалния свят нямат смисъл. Например отрицателни числа за дължини на страни и др.

Оценяване													
Въпроси	1	2	3										Общо
точки	2	2	2										6
Задачи				1	2	3	4						
точки				4	4	4	2						14
Практическа задача								1					
точки								20					20
скала	33 – 40 точки			Справям се отлично.									
	25 – 32 точки			Имам незначителни пропуски.									
	17 – 24 точки			Мога и по-добре.									
	10 – 16 точки			Трябва да положа още усилия.									
	0 – 9 точки			Не се справям.									

ТЕМА НА УРОКА:

16. СЪЗДАВАНЕ НА КОМПЮТЪРНА ПРОГРАМА ЗА РЕШАВАНЕ НА КОНКРЕТНА ЗАДАЧА

Брой учебни часове: 2

Вид на урока: Нови знания и упражнения

Опорни знания и умения: компоненти на графичен потребителски интерфейс, текстови полета, променливи, типове данни, моделиране на изчисления, откриване на грешки, тестване и създаване на тестове

Основна цел на урока: Изясняване на понятията „компютърна програма“ и „софтуер“, етапите на създаване на компютърна програма, упражнения за осъзнаване на концепциите.

Дидактически материали: учебник, проектор, електронен учебник, компютър със среда за програмиране.

Ход на урока:

1. Беседа с учениците за определенията на компютърна програма и софтуер, разликата между програмист и софтуерен инженер. Следва упражнение за изслушване на аудио, след което учениците попълват таблица за предполагаемите длъжности на служителите.

2. Задача 1 може да се обсъди заедно с учениците чрез дискусия за основните етапи на създаването на компютърна програма, които се състоят от разрешаване на проблема, последван от разработка на програмата, и накрая – поддръжка на програмата. Всеки етап/фаза се състои от подетапи. Тези етапи представляват жизнения цикъл на програмата. Представени са две описания, текстово и със диаграма, съответствията на които могат да се обсъдят с учениците.

3. Задача 2 показва още един начин за представяне на това как се създава програма. Тази схема изобразява действията при разработка на програма, включващи писане и коригиране на кода на програмата, компилиране, след което, ако има синтактични грешки, кодът се коригира и отново се компилира. Ако няма синтактични грешки, следва създаване на изпълнимия код, последвано от изпълнение, при което може да се въведат данни.

4. ЗАДАНИЕ 1 представлява проблем, който цели да илюстрира етапите при създаването на компютърна програма. Тук е препоръчително учениците да се оставят да помислят за решението, след което да се проведе беседа. Задачи 3 – 6 са свързани със заданието и преминават през различните етапи на създаване на програма с команден интерфейс (зад. 5) и графичен интерфейс (зад. 6). Зад. 3 изисква решението да се опише с блок-схема или псевдокод, зад. 4 е помощ за изясняване на ограниченията и типовете данни за всяка от променливите в заданието.

5. Зад. 7 се състои в откриване на грешки в проект. Тази задача е предвидена за групово обсъждане.

Очаквани трудности и предложения за преодоляването им: Решаването на всяка задача включва индивидуален подход. Важно е да се осъзнае, че първият етап е изключително важен. Също така тестването и откриването на грешки трябва да се извършва внимателно, една по една грешките трябва да се отстраняват, след което да се проверява дали има още грешки.

Междупредметни връзки: английски език, български език и литература, математика

Предварителна подготовка: алгоритъм, начини за описание на решение на проблем, променливи от различен тип, последователни операции, среда за програмиране, грешки

ТЕМА НА УРОКА:

17. СЪЗДАВАНЕ НА КОМПЮТЪРНА ПРОГРАМА ЗА РЕШАВАНЕ НА КОНКРЕТНА ЗАДАЧА. ПРАКТИКА

Брой учебни часове: 2

Вид на урока: Практически дейности (II)

Опорни знания и умения: коментари; основни етапи на разработка, оформяне на програмния код; моделиране на изчисления; откриване на грешки

Основна цел на урока: Знае основните етапи при създаване на компютърна програма; анализира и проектира решението на конкретна задача

ча; създава математическия модел за решаване на задачата; разработва алгоритъм за решаване на задачата; определя входно-изходни данни и техните типове; структурира и разработва графичен интерфейс, като използва обекти и декларира променливи; създава и описва програмния код; коментари; оформяне на програмния код; стартира, тества и валидира готовия проект; открива синтактични и логически грешки в програмата; отстранява синтактични и логически грешки при програмиране; спазва изисквания за оформяне на програмния код

Дидактически материали: учебник, проектор, електронен учебник, компютър със среда за програмиране

Ход на урока:

В урока се разглеждат всички етапи по създаването на програма, която решава конкретната задача, свързана с информация за планетите от Слънчевата система.

Проучването за създаването на приложението се извършва по групи.

Следват проектирането на приложението и създаването на модела на изчисление.

Дискутират се контролите, необходими за това приложение, както и реализацията на програмната логика на бутони за изчисления.

На учениците се предлага да участват дискусивно в дейности по оценка на текущо предложени интерфейс на приложението с цел подобряването му.

Предвидени са въпроси за проучване за други специфични контроли, които могат да се използват.

Учениците могат да направят вариант на изпълнение на задачата с допълнение на функционалността.

Дейности за контрол и оценка: по преценка на учителя.

Очаквани трудности и предложения за преодоляването им: Практическата работа с различни концепции предполага индивидуален подход със затрудняващите се ученици.

Междупредметни връзки: физика и астрономия, математика

ТЕМА НА УРОКА:

18. ПРОГРАМНИ КОНСТРУКЦИИ ЗА РЕАЛИЗАЦИЯ НА РАЗКЛОНЕН АЛГОРИТЪМ

Брой учебни часове: 2

Вид на урока: Нови знания и упражнения (НЗУ)

КОМПЕТЕНТНОСТИ КАТО ОЧАКВАНИ РЕЗУЛТАТИ ОТ ОБУЧЕНИЕТО

Булев тип данни

– Обосновава необходимостта от разклоняване на алгоритмичния процес.

– Използва константите от булев тип данни.

– Дава примери, в които се използва булев тип данни.

– Декларира булева променлива.

– Присвоява стойност на булева променлива.

– Записва на език за програмиране булев израз, съдържащ операция за сравнение.

– Изписва синтактично правилно на езика за програмиране основните логически операции – логическо отрицание, дизюнкция, конюнкция.

– Познава приоритета на логическите операции.

– Пресмята без използване на компютър стойността на булев израз.

– Съставя със средствата на език за програмиране сложен булев израз, отговарящ на дадена логическа ситуация.

Условен оператор

– Описва синтаксиса и семантиката на кратка и пълна форма на условен оператор.

– Описва разклонен алгоритъм с помощта на условен оператор.

– Използва условен оператор за проверка на коректността на входните данни за програма.

– Използва условен оператор за обработка на свойства на радиобутон и поле за отметка.

Вложени условни оператори

– Обяснява семантиката на вложени условни оператори в кратка и пълна форма.

– Проиграва изпълнението на фрагмент на програма, съдържаща вложени условни оператори.

– Записва синтактично и логически правилно вложени условни оператори.

– Създава модел и алгоритъм за решаване на задача чрез използване на вложени условни конструкции.

– Реализира модел за решаване на задача чрез използване на вложени условни оператори.

– Заменя вложен условен оператор с единичен и обратно.

НОВИ ПОНЯТИЯ

булев тип данни; булеви константи; логически операции; приоритет на логическите операции; булев израз; условен оператор; съставен оператор; вложен условен оператор

Дидактически материали: учебник, проектор, електронен учебник, компютър, среда за програмиране

Ход на урока:

В урока се разглеждат програмни конструкции за реализация на разклонен алгоритъм и приложни аспекти на темата, свързани с вземането на решения. Разглеждат се също междупредметните връзки с математиката от гледна точка на различните логически булеви изрази и функции. Открива се връзката между тях и полезността от интерпретирането и прилагането им.

В урока са представени различни транспортни схеми, визуализиращи основните видове алгоритми. Потребността от тези на разклонен алгоритъм се въвежда с пример за теглене на пари от банкомат.

Представят се булевите изрази и логическите операции и връзката им с вземането на решения, разклонените алгоритми. Първоначално се въвежда понятието „съждение“ и как то се оценява.

Представени са и някои нови примери като връзката с изкуствен интелект и хуманоидните роботи и употребата на логика в тях, видео за програмирането на техните действия. Демонстрационно видео може да намерите в сайта към учебника. Разучават се програмни конструкции за реализация на разклонен алгоритъм с практически задачи, законите на Де Морган и ролята им при отрицание на сложно условие, а също и използването на верностни на таблици, с които можем да направим преобразуване или доказване на верността.

В задачите следва представянето на кратка и пълна форма на оператор за разклонен алгоритъм и особеностите в синтаксиса.

Представени са варианти на пресмятане на математически функции чрез условни оператори, например абсолютна стойност.

След урока има няколко задачи за самостоятелна работа. Урокът завършва с рефлексия.

Очаквани трудности и предложения за преодоляването им: Учениците често изпитват трудности с: конструирането на условия, приоритет на операциите, подреждане на условните оператори, използвани при влагане.

Междупредметни връзки: математика, логика, ИТ, английски език, български език и литература

Предварителна подготовка: познаване на средата за програмиране, основните операции при конструиране на алгоритми, етапите на създаване на програма, познаване на някои елементарни математически операции и функции.

ТЕМА НА УРОКА:

19. ПРОГРАМНИ КОНСТРУКЦИИ ЗА РЕАЛИЗАЦИЯ НА РАЗКЛОНЕН АЛГОРИТЪМ. ПРАКТИКА

Брой учебни часове: 2

Опорни знания и умения: програмни конструкции за реализация на разклонен алгоритъм; булев тип данни; условен оператор; вложени условни оператори

Основна цел на урока: Усъвършенстване на уменията за прилагане на разклонените алгоритми при решаване на практически задачи.

Дидактически материали: учебник, компютри, програмни среди, ресурсен сайт към учебника

Ход на урока:

Преговор на програмните конструкции за реализация на разклонен алгоритъм, булев тип данни, условен оператор, влагане на условни оператори.

Въпросите целят проверка на разбирането на основните условни конструкции при създаване на компютърна програма, реализираща алгоритъм за решаване на дадена задача. Линеиното изпълнение на програмата може да се промени с използването на условни оператори, при които

след резултата от проверката на дадено условие се изпълняват различни части на програмата. Въпросите проверяват и уменията за конструиране на условия, използващи сравнения и булеви операции.

Отговори на въпросите, коментари, кратки решения:

Въпрос 1: а) логически

Въпрос 2: а) булев

Въпрос 3: б) логически

Въпрос 4: а) $a < b$ and $a < c$

Въпрос 5: г) е правилният отговор, защото $3 > 4$ е лъжа (false), поради което ще се изпълни „в противен случай“ или

Отпечатай В

Отпечатай С

Въпрос 6: в) false

Стойността на израза $(x == 8)$ or $(x > y)$, при условие че $x = 10$ и $y = 10$ е:

$(10 == 8)$ or $(10 > 10)$

false or false

false

Въпрос 7: г) А. 1 Б. 1

Решение: Изразът е (not a and b) or c

А. $a = 0$; $b = 1$; $c = 1$

not a = not 0 = true

$b = 0 = false$

$c = 1 = true$

(true and false) or true = (false)

or true = true = 1

Б. $a = 1$; $b = 1$; $c = 1$

not a = not 1 = false

$b = 1 = true$

$c = 1 = true$

(false and true) or true = (false)

or true = true = 1

Може да се забележи, че стойността на израза винаги е 1, ако $c = 1$, което спестява време за намирането на резултата.

Въпрос 8: Сложете фигурни скоби на местата, където прецените, така че след изпълнението на описания алгоритъм да се отпечата 9:

1. Първоначално $x = 2$.

2. Проверяваме дали x е по-малко от 4.

3. Увеличаваме x 5 пъти.

4. Отпечатаваме предното число преди x .

5. В противен случай.

6. Намаляваме x с 3.

7. Отпечатаваме следващото след x число.

1. $x = 2$

2. if $x < 4$

3. $x = x * 5$

4. print($x-1$)

5. else

6. $x = x - 3$

7. print($x+1$)

Например:

C#

```
x = 2;
```

```
if (x < 4) {
```

```
    x = x * 5;
```

```
    Console.WriteLine(x-1);
```

```
}
```

```
else {
```

```
    x = x - 3;
```

```
    Console.WriteLine(x+1);
```

```
}
```

JavaScript

```
var x = 2;
```

```
if (x < 4) {
```

```
    x = x * 5;
```

```
    document.write(x-1);
```

```
}
```

```
else {
```

```
    x = x - 3;
```

```
    document.write(x+1);
```

```
}
```

Python

```
x = 2
```

```
if x < 4:
```

```
    x = x * 5
```

```
    print(x-1)
```

```
else:
```

```
    x = x - 3
```

```
    print(x+1)
```

Следват задачи с логически изрази и условни оператори в програмни фрагменти, които проверяват уменията за прилагане на наученото.

Задача 1: Прилага се законът на Де Морган

а) not ($x > 0$ and $y > 0$)

$x <= 0$ or $y <= 0$

б) not (country = "BG" and city = "Varna")

country != "BG" or city != "Varna")

Има населено място Варна в САЩ **Varna is a village in Marshall County, Illinois, United States).**

Задача 2:

a) $(a - b = 1) \text{ or } (a - b = -1)$

$\text{abs}(a-b) = 1$

Ако $a = 5$ и $b = 4$, тогава $a - b = 1$,

от което следва, че изразът е истина.

б) Ако координатите на точките са $X(x1, x2)$, $Y(y1, y2)$:

$(x2 == 0) \text{ and } (y2 == 0) \text{ and } ((x1 - y1 == 0)$

or $(x1 + y1 == 0))$

$(x2 == 0) \text{ and } (y2 == 0) \text{ and } (\text{abs}(x1) == \text{abs}(y1))$

Ако $X = (3.5, 0)$, $Y = (-3.5, 1)$, тогава стойността на израза е лъжа, защото $y2 == 0$ не е истина.

Ако $X = (-3.5, 0)$, $Y = (3.5, 0)$, тогава стойността на израза е истина.

Стандартната математическа функция, която може да се използва, е тази за намирането на абсолютна стойност.

Задача 3:

a) Ако условието $1 = 2$ в първата проверка се промени на $3 = 1$ or $3 = 2$, тогава местата на „да“ и „не“ ще се сменят.

Ако условието $1 > 2$ във втората проверка се промени на $1 < 2$, тогава местата на „да“ и „не“ ще се сменят.

РАДИОБУТОНИ И ПОЛЕТА ЗА ОТМЕТКИ

Учениците разглеждат видеата и отговарят на поставените въпроси.

ОСНОВНИ ЕТАПИ ПРИ ИЗПОЛЗВАНЕ НА РАДИОБУТОНИ В ПРИЛОЖЕНИЯ

Защо се прилага групиране на радиобутоните?

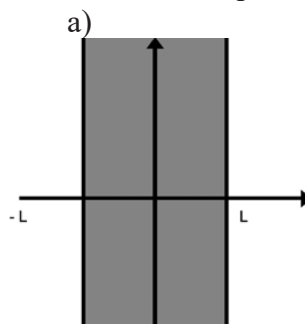
Само един от радиобутоните може да се избере в даден момент.

Какво е различното при полетата с отметки? А те групират ли се, или не?

Различното при полетата с отметки в сравнение с радиобутоните е, че може да се избират повече от едно поле. Те също се групират.

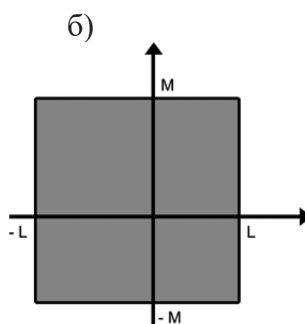
САМОСТОЯТЕЛНА РАБОТА

Задача 1. Напишете логически изрази, с които се проверява дали точка с координати (x, y) принадлежи на заштрихованата област:

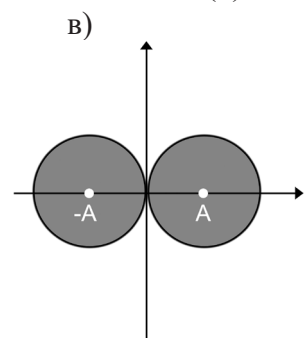


$x \leq L \text{ and } x \geq -L$

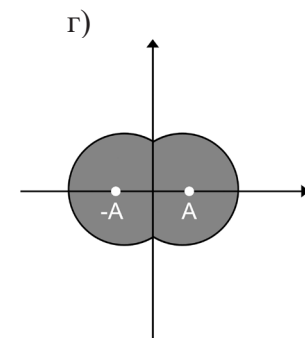
или $\text{abs}(x) \leq L$



$\text{abs}(x) \leq L \text{ and } \text{abs}(y) \leq M$



$((x-A)*(x-A) + y*y \leq A*A) \text{ or } ((x+A)*(x+A) + y*y \leq A*A)$



$((x-A)*(x-A) + y*y \leq R*R) \text{ or } ((x+A)*(x+A) + y*y \leq R*R)$

Тук сме предположили, че радиусът е един и същ, равен на R.

Задача 2. Съставете програма, която определя дали полето с координати (x, y) от шахматна дъска n x n (n ≤ 8) се бие от фигура, разположена в поле с координати (p, q). Изберете една от фигурите: цар, топ, царица и кон.

Останалите шахматни фигури биха ударили следните маркирани позиции:

Цар

```
abs(x-p) <= 1 and abs(y-q) <= 1
```

Топ

```
x == p or y == q
```

Царица

```
x == p or y == q or abs(x-p) == abs(y-q)
```

Кон

```
(abs(x-p)==1 and abs(y-q)==2) or (abs(x-p)==2 and abs(y-q)==1)
```

Задача 3. Дадени са две точки в равнината (x1, y1) и (x2, y2). Напишете програма, която извежда колко от тях (0, 1 или 2) се намират във вътрешността на окръжност с център т. (0, 0) и радиус r.

```
d1 = math.sqrt(x1*x1+y1*y1)
```

```
d2 = math.sqrt(x2*x2+y2*y2)
```

```
count = 0
```

```
if d1 < r:
```

```
    count = count + 1
```

```
if d2 < r:
```

```
    count = count + 1
```

```
print(count)
```

Задача 4. Напишете програма, която пресмята и извежда разстоянието между две точки в равнината A(xA, yA) и B(yB, yB). В програмата да се предвиди валидация на входа, ако точките съвпадат, и да се изведе съобщение „no distance“. Изведете разстоянието с точност до два знака след десетичната точка.

```
d = math.sqrt((x1-x2)*(x1-x2)+(y1-y2)*(y1-y2))
```

Дейности за контрол и оценка: урокът е подходящ и за оценяване по преценка на учителя.

Междупредметни връзки: БЕЛ, английски език, математика

Допълнителни въпроси, задачи и упражнения:

1. Примери за условни операции са всички освен:

a. Докато има още чекове за обработване, изпълнете следните стъпки.

b. Ако сумата на чека е по-малка или равна на текущия баланс на сметката, тогава осребрете чека; в противен случай кажете на клиента, че в сметката няма достатъчно пари.

c. Ако x не е равно на нула, разделете y на x и запишете резултата в z.

d. Ако температурата е по-ниска от 15 градуса, включете отоплението.

e. Ако сместта е твърде суха, тогава добавете половин чаша вода в купата.

Отговор: a. е цикъл

2. Определете стойността на следния булев израз, ако x е 10 и y е 10: (x == 8) or (x > y).

a. 8 b. 10 c. False d. null e .

True

Отговор: c. False

ТЕМА НА УРОКА:

20. ПРОГРАМНИ КОНСТРУКЦИИ ЗА РЕАЛИЗАЦИЯ НА РАЗКЛОНЕН АЛГОРИТЪМ. ПРАКТИКА

Брой учебни часове: 2

Вид на урока: Практически дейности (П)

Опорни знания и умения: условни оператори, графичен потребителски интерфейс, контроли

Основна цел на урока: Затвърждаване на наученото за условни оператори, продължено от програмни конструкции за реализиране на разклонени алгоритми за избиране на повече от две възможности, операторите if... else if и switch.

Дидактически материали: учебник, проектор, компютър, среда за програмиране на избрания език

Ход на урока:

В устна беседа се обсъжда начинът за обновяване на графичното приложение за суши. Представен е прототип с грешки, които, след като се намерят, се иска да се направи промяна в приложението. Този пример цели да се упражни избор на входни данни и очаквани резултати от приложението. Друга важна тема, която се реализира

най-напред с условни оператори, е проверка за валидност на входните данни. Валидацията на данни ще се затвърди по-късно и ще се реализира с помощта на цикли.

Дейности за затвърждаване на НЗУ:

- Започва се със задача 1, именно софтуерният проект „Меню за суши“, в който са добавени няколко специфични изисквания с предложен прототип, в който има грешки за откриване. Това упражнение цели да покаже как се обновява вече създадено приложение.

- Задача 2 цели да се моделира новият интерфейс с обсъждане на решението, последвано от реализирането му с програма.

- Чрез задача 3 се цели да се направи описание при какви входни данни са очакваните резултати от приложението.

- Задача 4 поставя проблема за избор на един от 3 варианта, решен в 4 програмни фрагмента, реализирани с кратката (if) и пълната форма (if... else) на условен оператор. След откриването на фрагмента, който правилно решава задачата, препоръчваме да се обсъди защо е така и кое е грешно в неверните фрагменти.

- Зад. 5 усложнява зад. 4 като поставя повече възможности за ценообразуване. Заданието е да се напише конзолно приложение, което по дадени стойности, които се обновяват, извежда таксата за съответната държава.

- Следва въвеждане на if... else if и switch конструкции за избор на една от много възможности. Добре е да се обсъди разликата и кога използваме всеки от операторите, също така да се разгледа код решаващ един и същи проблем с помощта на if... else if и switch.

- Зад. 6 упражнява създаването и използването на switch.

- Следното задание предлага да се види програмата с име NumberOfDays, в която са допуснати няколко грешки, които учениците да потърсят, намерят и коригират.

- Зад. 7 не казва кой оператор трябва да се използва, ученикът трябва да вземе решение, като може да му се помогне, че това е if...else if или if... elif, ако програмният език предлага такъв. Подчертайте, че условието не е израз, в който променлива се проверява дали е равна на

дадени стойности. Променливата има стойности които са в интервал, и проверката изисква конструиране на по-сложен израз.

- Следват примери за използване на условен оператор за проверка на коректността на входните данни, които учениците могат да разгледат сами или заедно с учителя.

- Задача 8 цели да затвърди наученото за условни оператори чрез създаване на конзолно приложение по описание на заданието. Подходящо е да се направи групово, тъй като някои ученици може да се затруднят.

- Предложили сме две самостоятелни задачи за създаване на приложения.

Дейности за контрол и оценка: по преценка на учителя. Подходящи за оценка са повечето задачи.

Очаквани трудности и предложения за преодоляване: в този урок може да се очакват трудности, свързани с конструиране на условните оператори, условията за проверка, подход за решаване на задачите.

Междупредметни връзки: БЕЛ, английски език, география и икономика, технологии и предприемачество

ТЕМА НА УРОКА:

21. УСЛОВНИ КОНСТРУКЦИИ И РАЗКЛОНЕНИ АЛГОРИТМИ. ОБОБЩЕНИЕ

Брой учебни часове: 1

Вид на урока: Обобщение (О)

Опорни знания и умения: условни конструкции и разклонени алгоритми; булеви изрази

Основна цел на урока: Обобщение на знанията за условни конструкции и разклонени алгоритми.

Дидактически материали: учебник, компютър, работни листове, ресурсен сайт към учебника

Очаквани трудности и предложения за преодоляване: различна скорост на изпълнение на задачите от учениците. Може да се използват допълнителни задачи за учениците, които работят по-бързо.

Междупредметни връзки: ИТ, математика

Ход на урока:

ОПЕРАЦИИ ЗА СРАВНЕНИЕ

Коментират се булевите данни и логическите операции между тях.

Задача 1

а) 9; б) true; в) false

Задача 2 а) и г)

Задача 3 се разглежда с различни стойности на А и В и се проверява какъв е изходът за всеки набор от входни данни.

УСЛОВНИ КОНСТРУКЦИИ

Задача 4 може да се използва за затвърждаване на знанията за импликация. Преди този урок може да се поиска от учениците да потърсят повече информация за логическата операция импликация.

Задача 5 има за цел да затвърди знанията за пълна и кратка форма на оператора if... then

Задача 6 изисква от учениците да напишат програма, която по зададени координати на точка да проверява в кой квадрант е точката. Необходимо е да се използва кратката форма на условния оператор.

Задача 7. Тази задача може да се постави за домашна работа, като само се коментира математическото условие, при което точката ще лежи вътре в кръга.

Задача 8 се решава в час с цел да се коментират възможностите за практическо приложение на пълната форма на оператор if... then... else.

Задача 9. Задачата изисква случаен избор на число в определен интервал от компютъра и възможност потребителят да направи два опита да го познае, като след всеки опит получава информация дали неговото число е по-голямо, или по-малко от избраното. Задачата може да се остави за самостоятелна работа.

Задача 10 проверява знанията за използване на съставни условия. Може да се коментира, а решението да остане за домашна работа.

Оператор за избор на вариант – коментират се операторът и възможностите за неговото използване.

Задача 11. Решава се в клас, като се коментират различни варианти за решение – с оператор за многовариантен избор или с пълната форма на условния оператор.

ТЕМА НА УРОКА:

22. УСЛОВНИ КОНСТРУКЦИИ И РАЗКЛОНЕНИ АЛГОРИТМИ. ПРАКТИКА

Брой учебни часове: 2

Вид на урока: Практически дейности (II)

Опорни знания и умения: компоненти на графичен потребителски интерфейс, текстови полета, променливи, типове данни, моделиране на изчисления, откриване на грешки, тестване и създаване на тестови случаи; разклонени алгоритми и условен оператор; графично описание на алгоритмите

Основна цел на урока: Усъвършенстване на уменията за използване на елементи на потребителски интерфейс и работа с условен оператор и стрингове

Дидактически материали: учебник, мултимедия проектор, ресурсен сайт към учебника, компютър

Ход на урока:

Задача 1 цели да се създаде приложение с графичен потребителски интерфейс за сравняване на две числа. Коментират се различни варианти за създаване на приложението и възможните контроли за въвеждане и извеждане на информация. Реализацията на различни програмни езици може да се разгледа от ресурсите в сайта към учебника.

Задача 2 е типичен пример за създаване на алгоритъм за решаване на математически проблем – намиране на реалните корени на квадратно уравнение. Необходимо е да се обърне внимание на използването на математически функции от подходяща библиотека. Считаме, че при тази задача е удачно да се припомнят и затвърдят знанията за описание на алгоритъма и е предложен един вариант с блок-схема. Добра идея е да се разгледат решенията като конзолно приложение и като приложение с графичен потребителски интерфейс.

```

JavaScript
function solve(args) {
    var a = +args[0],
        b = +args[1],
        c = +args[2],
        disc = Math.pow(b, 2) -
(4 * a * c);
    if (disc < 0) {
        console.log('no real
roots');
    }
    else if (disc === 0) {
        var solve = -b / (2 * a);
        console.log('x1=x2=' +
solve.toFixed(2));
    }
    else if (disc > 0) {
        var sqrt = Math.sqrt(disc);
        var solve1 = (-b + sqrt)
/ (2 * a);
        var solve2 = (-b - sqrt)
/ (2 * a);
        if (solve2 < solve1) {
            var temp = solve1;
            solve1 = solve2;
            solve2 = temp;
        }
        console.log('x1=' + solve1.
toFixed(2) + "; x2=" + solve2.
toFixed(2));
    }
}

```

С# (Визуално програмиране)

Елементи на графичния потребителски интерфейс:

Компоненти на GUI	брой
Етикети за надписи	3
Етикет за отпечатване на резултата	1
Текстово поле/кутия за въвеждане	3
Бутон „Реша уравнението“	1

```

//Събитието е на бутона:
private void button1_Click(object
sender, EventArgs e)
{
    double a, b, c, D,x1,x2;
    a = Convert.
ToDouble(txt_a.Text);
    b = Convert.
ToDouble(txt_b.Text);
    c = Convert.
ToDouble(txt_c.Text);
    D = b *b - 4 * a * c;
    if (D < 0) lblRezult.Text
= "Няма реални корени";
    else if (D > 0)
    {
        x1 = (-b + Math.
Sqrt(D)) / (2 * a);
        x2 = (-b - Math.
Sqrt(D)) / (2 * a);
        lblRezult.Text =
"x1=" + x1 + "; x2=" + x2;
    }
    else lblRezult.Text =
"x1=x2=" + (-b / (2 * a));
}

```

Допълнителна информация и идеи за решение на различни програмни езици можете да откриете в ресурсите към учебника.

Задача 3 изисква да се създадат тестови примери и да се тества създаденото приложение. Предложени са възможности за решаване на квадратно уравнение и чрез друга среда (напр. Microsoft Mathematics 3.0.). Добра идея е резултатите от тестването да се запишат в таблица и да се анализират причините за некоректните резултати.

За самостоятелна работа може да се предложи на учениците да създадат приложение за решаване на линейно уравнение, като преминават през същите стъпки: създаване и описание на алгоритъма, разработка на графичен интерфейс, програмиране и тестване.

Тази задача може да бъде поставена и за домашна работа по преценка на учителя.

Към темата са предложени няколко интересни пъзела и игри, чрез които да се провокират интересът на учениците и желанието за създава-

не на подобно приложение. Част от тях могат да се разгледат в час и да се обсъдят възможностите за тяхната програмна реализация. Останалите могат да се оставят за самостоятелна работа.

Дейности за контрол и оценка: този урок не е подходящ за оценяване. Може да се оцени самостоятелната работа на учениците или тяхната активност при решаването на поставените учебни задачи.

Очаквани трудности и предложения за преодоляването им: различна скорост на изпълнение на задачите от учениците. Проблемът може да се преодолее, като се задават допълнителни задачи от пзелите и игрите на учениците, които по-бързо решават поставените задачи.

Междупредметни връзки: математика, ИТ

ТЕМА НА УРОКА:

23. УСЛОВНИ КОНСТРУКЦИИ И РАЗКЛОНЕНИ АЛГОРИТМИ. КОНТРОЛ И ОЦЕНКА

Брой учебни часове: 1

Опорни знания и умения:

- програмни конструкции за реализация на разклонени алгоритми
- булеви данни
- условен оператор
- многовариантен избор
- вложени условни оператори

Основна цел на урока:

Контрол и оценка на знанията и уменията на учениците при работа с условни конструкции и разклонени алгоритми

Дидактически материали:

- работни листове
- компютри с интернет достъп

Дейности за контрол и оценка:

- **Въпросите** са с избираем и свободен отго-

вор. С тях се проверяват знанията на учениците по темата.

- **Задачите** имат за цел да проверят практическата подготовка на учениците.

- **Практическата задача** проверява уменията на учениците да създават самостоятелно компютърно приложение за решаване на определена задача.

Верни отговори на въпросите от теста:

Въпрос 1 – г

Въпрос 2 – б

Въпрос 3 – б

Въпрос 4 – д

Въпрос 5 – в

Въпрос 6 – б

Въпрос 7: M= – 1

Въпрос 8: N= – 2

Задача 1: `if(x<=y) Console.WriteLine(x);`
`else Console.WriteLine(y);`

Задача 2. `int a=3;` (променливата трябва да е от изброим тип)

Задача 3

```
int a=7;int b=6;
```

```
if(a<b) c=a;
```

```
else c=b;
```

Задача 4

```
switch(event)
```

```
{case 0: action="jump"; break;
```

```
case 1: action="run";break;
```

```
case 2:action="run";break;
```

```
default:action="walk";break;
```

```
}
```

Задача 5

```
if ((num>=1) && (num<=100) && (num%2==0))
```

```
Console.WriteLine("OK");
```

```
else Console.WriteLine("Not Valid");
```

Зад. 6.

```
if ((a+b>c) && (a+c>b) && (b+c>a) && (a>0) && (b>0) && (c>0))
```

Практическа задача: Примерно решение с вложени условни оператори:

```
int cod;
int N;
double C;
cod=int.Parse(Console.ReadLine());
N=int.Parse(Console.ReadLine());
C=double.Parse(Console.ReadLine());
if(cod==3)
{if((n>3)&&(n<10)) C=C*0.85;
else if(n>10)C=C*0.75;
}
else if((cod>1)&&(cod<=3))
{if(n>3)&&(n<10) C=C*0.85*0.5;
else if(n>10) C=C*0.75*0.5;
}
Console.WriteLine("Цената на двупосочен билет е {0:#.##}",C*0.90);
```

Заб. Може да се използва и операторът за многовариантен избор.

Оценяване														
Въпроси	1	2	3	4	5	6	7	8						Общо
точки	1	1	1	1	1	1	1	1						8
Задачи									1	2	3	4	5	6
точки									2	2	2	2	2	2
Практическа задача														1
точки														20
скала	33 – 40 точки					Справям се отлично.								
	25 – 32 точки					Имам незначителни пропуски.								
	17 – 24 точки					Мога и по-добре.								
	10 – 16 точки					Трябва да положа още усилия.								
	0 – 9 точки					Не се справям.								

ТЕМА НА УРОКА:

24. ПРОГРАМНИ КОНСТРУКЦИИ ЗА РЕАЛИЗАЦИЯ НА ЦИКЛИЧНИ АЛГОРИТМИ

Брой учебни часове: 2

Вид на урока: Нови знания и умения (НЗУ)

КОМПЕТЕНТНОСТИ КАТО ОЧАКВАНИ РЕЗУЛТАТИ ОТ ОБУЧЕНИЕТО

циклически алгоритъм и програмни конструкции за реализация на цикличните алгоритми

НОВИ ПОНЯТИЯ

цикъл, тяло на цикъла, итерация, брояч, структура на циклически алгоритмични конструкции, оператор за цикъл с предусловие, оператор за цикъл с постусловие, оператор за цикъл, управляван от брояч, списъчно поле

Дидактически материали: учебник/електронен учебник, мултимедия проектор, компютър, интернет, ресурсен сайт към учебника

Междупредметни връзки: физика и астрономия, математика, информационни технологии, чужди езици

Ход на урока:

Разглеждат се понятието „цикъл“ и различните програмни конструкции за реализацията на циклическите алгоритми. Обсъждат се особеностите, приликите и разликите между оператор за цикъл с предусловие, оператор за цикъл с постусловие и оператор за цикъл, управляван от брояч.

Дават се примери за приложението на всеки от тях. Смятаме, че е добра идея да се реши една и съща задача чрез използването на различните оператори за цикъл, като се обсъдят разликите между тях. В урока също са предложени по няколко решения на един и същи проблем.

За упражнение и затвърждаване на получените знания предлагаме да се използват задачи 4, 5, 6 и 7. По преценка на учителя може да се използва груповата работа, като всяка група работи по някой от тези примери, след което се представя работата и се провежда дискусия.

Обръща се внимание на това, че най-удобната графична компонента за обработка и отпечатване на стойности в цикъл е списъчната кутия.

Разглежда се задача 8, решението на която в различни програмни езици може да се открие в ресурсите към учебника.

Примерите в урока са свързани най-вече с учебното съдържание по математика и физика. Предлагат се и задачи игри, които целят повишаване на активността и интереса на учениците.

Задачите за самостоятелна работа могат да се използват за провокиране на интереса на учениците, които работят по-бързо. Част от тях могат да се поставят и за работа вкъщи.

Очаквани трудности и предложения за преодоляване: една очаквана трудност е обемът на новите знания в този урок, която се преодолява с разглеждане на ключови примери в онлайн среда за програмиране и използването на ресурсния сайт към учебника.

Предлагаме една реализация на задача 3 от секцията за самостоятелна работа.

Sample Run: (Inputs are in bold for the sample run only.)

```
Do you want to guess a number?  
y/n: y  
Enter the maximum size: 100  
Enter your guess: 50  
The number is larger.  
Do you want to make another guess?  
y/n: y  
Enter your guess: 80  
The number is smaller.  
Do you want to make another guess?  
y/n: y  
Enter your guess: 65  
That is correct.  
Total guesses: 3  
Do you want to guess a new number?  
y/n: y  
Enter the maximum size: 10  
Enter your guess: 5  
The number is larger.  
Do you want to make another guess?  
y/n: n  
Total guesses: 1  
Do you want to guess a new number?  
y/n: n  
>>>
```

Програмата (Python)

```
# Program allows for numbers to
be guessed.
import random
choice = input("Do you want to
guess a number? y/n: ")
while choice == 'y':
    max = int (input("Въведете мак-
симално число за отгатване: "))
    x = random.randrange(max+1);
    choice2 = 'y'
    attempts = 0
    found = 0
    while choice2 == 'y' and found
    == 0:
        guess = int(input("Познайте чис-
ло от 0 до ", max, ":"))
        attempts+=1
        if guess == x:
            print ("That is correct")
            found = 1
            if guess > x:
                print("The number is lower")
            if guess < x:
                print ("The number is higher")
            if guess != x:
                choice2 = input("Do you want to
make another guess? y/n: ")
            print("Number of guesses: ",
attempts)
            choice = input("Do you want to
guess a new number? y/n: ")
```

ТЕМА НА УРОКА:

25. ПРОГРАМНИ КОНСТРУКЦИИ ЗА РЕАЛИЗАЦИЯ НА ЦИКЛИЧНИ АЛГОРИТМИ. ПРАКТИКА

Брой учебни часове: 2

Опорни знания и умения: прехвърляне на информация (графична, аудио, видео) от и към мобилни устройства; инсталиране на приложения на мобилно устройство в зависимост от операционната система.

Основна цел на урока: Усъвършенстване на уменията за конструиране на цикли, прилагането им в различни ситуации, включително при работа с файлове.

Дидактически материали: учебник, електронен учебник, компютър, среда за програмиране на даден език, софтуер

Ход на урока:

1. Дейности за актуализиране на ОЗУ: Този урок е за усвояване на умения за използване на цикли в различни графични приложения. Започва се с въведение в компютърната графика. Представят се библиотеки за 2D графика.

2. Дейности за затвърждаване на ОЗУ:

- Задача 1 е за създаване на програма за въвеждане и записване на имената на учениците от класа в текстов файл. Реализирана е на различни програмни езици.

- Задача 2 е за програмиране на четене на записани имена от текстов файл и тяхното отпечатване на конзола.

- Задача 3 е за затвърждаване на уменията на учениците да използват и проследяват резултати от цикли. Задачата представя кредитен калкулатор с графичен потребителски интерфейс, който при въведени от потребителя данни за обща сума на кредита, годишен лихвен процент и брой месеци да изчислява и отпечатва погасителния план с месечната анюитетна вноска (лихва). Представени са два тестови примера за проверка на приложението за коректност. Очаква се да се създадат още няколко тестови примера, които може да се подберат, така че да се провери валидацията на входа, която трябва да изисква неотрицателни стойности за кредит-

на сума, брой месечни вноски, лихвен процент. Тестовите случаи могат да бъдат:

Кредит сума = 10000; Годишна лихва = 5;
Брой месечни вноски = 1.

Кредит сума = 11000; Годишна лихва = 5;
Брой месечни вноски = -10.

Кредит сума = 11000; Годишна лихва = -6.

Брой месечни вноски = 12.

- Задача 4 илюстрира програма за често прилаганите алгоритми за намиране на средни, минимални и максимални стойности на данни, въведени от файл. Програмният код за въвеждане на данните от файл е даден, включващ проверка за коректност на входните данни, като например успехът на ученика да е между 2 и 6. Тази задача показва как се използват цикли с предусловие и постусловие.

3. Дейности за контрол и оценка: по преценка на учителя. Подходящи за оценка са всички задачи за самостоятелна работа.

Очаквани трудности и предложения за преодоляването им: в този урок може да се очакват трудности, свързани с конструирането на цикли, както и работа с файлове. Помощта на учителя е необходима.

Междупредметни връзки: БЕЛ; английски език; технологии и предприемачество; математика

Допълнителни въпроси, задачи и упражнения:

Може да се потърси и предостави информация за използване на текстов файл, който е от формат csv (comma separand value).

Задачи за самостоятелна работа:

- Задача 1 изисква да се въведат цените на 5 стоки от магазин в текстов файл и да се отпечата въведените цени, средната стойност за цена на стока и общата сума. Кодът от предходните задачи може да се адаптира за решаването на тази задача.

- Задача 2 е продължение на задача 1 и е за намиране и извеждане на най-ниската и най-високата цена, подобно на задача 4 от предишния раздел.

- Задача 3 изисква създаване на приложение с графичен интерфейс за решаване на задача 1 и задача 2.

- Задача 4 е изследователски проект за Хейлстонови числа (Hailstone numbers). Хейлстоновите числа са дефинирани като редица от числа, които се получават по следното правило: ако числото е четно, се разделя на 2; ако числото е нечетно, увеличаваме го три пъти и добавяме 1. Например, ако започнем със 7, ще получим следната редица от числа: 7 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1. Интересно е, че без значение от кое число ще започнем, накрая се получава числото 1. Тези числа се наричат така, защото както при градушката те подскочат надолу-нагоре (увеличават се и намаляват), докато накрая се срещат със Земята (стават 1).

Подусловие а) се състои в създаване на програма, в която за дадено число се отпечата редицата от числа до получаването на 1. Това изисква цикъл, който да проверява дали последното пресметнато число на редицата е 1 и да спира, ако е така. В тялото на цикъла трябва да има условен оператор, който да проверява дали числото е четно, или не, което става, като се намери остатъкът при деление на 2 и се провери дали е 0 или 1. В зависимост от резултата се пресмята следващият член на редицата чрез деление на 2 или се умножава по 3 и се добавя 1.

Подусловие б) изисква създаване на 5 тестови примера, за да проверите хипотезата, че накрая винаги се получава числото 1. Трябва да се добави и променлива, която да преброи итерациите до получаването на 1.

Примерни тестови данни: 7, 1, 2, 10, 101.

С цел затвърждаване на наученото за файлове се иска резултатите от тестовете да се запишат в текстов файл, който е от формат csv (comma separand value). Въпреки че този файлов формат е текстов, той съдържа таблични данни. Програмата, която реализира това, е дадена в уебсайта с ресурси за учителя.

ТЕМА НА УРОКА:

26. ПРОГРАМНИ КОНСТРУКЦИИ ЗА РЕАЛИЗАЦИЯ НА ЦИКЛИЧНИ АЛГОРИТМИ В КОМПЮТЪРНА ГРАФИКА. ПРАКТИКА

Брой учебни часове: 2

Опорни знания и умения: циклични алгоритми, векторна и растерна графика, координатна система, познания за основните геометрични фигури, приложения с графичен интерфейс

Основна цел на урока: Използване на циклични алгоритми в двумерна компютърна графика. Запознаване с основните концепции в компютърната графика като графични примитиви, представяне на цветовете, векторна и пикселна графика, специализиран интерфейс за чертаене на графики.

Дидактически материали: учебник, проектор, компютър, ресурсен сайт към учебника

Ход на урока:

1. Дейности за актуализиране на ОЗУ: с устна беседа за основните графични примитиви и техните параметри, последвано от запознаване с библиотеки и модули, включващи функции за чертаене в зависимост от избрания език за програмиране. Важно е да се обясни координатната система, която се използва в компютърната графика. Представяне на програми, които чертаят графика.

2. Дейности за затвърждаване на НЗУ:

- Със задача 1 се упражнява (словесно или с псевдокод) чертаене на точка чрез отсечка, чертаене на квадрат чрез отсечки и на триъгълник чрез отсечки. Това помага да се изгради визуална представа за разположението на фигурите в координатната система.

- Със задача 2 се упражнява създаване на графично приложение за чертаене на окръжност, като се ползват различни параметри за дебелина на контура, цвят и радиус.

- Със задача 3 се затвърждават уменията на учениците да чертаят, като ползват случайни стойности за цвета и дебелината на контура на фигура.

- Задача 4 продължава задача 3, като добавя цикъл за чертаене на концентрични окръжности.

- Задача 5 упражнява създаване на приложе-

ние за чертаене на квадрати с помощта на линии, свързващи върховете.

- Задача 6 упражнява чертаене на квадрати с различен цвят и дебелина на линиите.

- Задача 7 упражнява създаване на цикъл за чертаене на 5 квадрата, които са начертани отделно.

- Задача 8 е за откриване на грешки в програма, така че да се начертае фигура в даден файл.

- Задача 9 е за създаване на програма за чертаене на зададена фигура, което включва определяне на примитиви и симетрии.

Накрая е даден пример за употреба на софтуер Processing, използван в дигиталните изкуства. Този софтуер може да се ползва заедно с други езици, като например Python и JavaScript.

3. Дейности за контрол и оценка: темата е подходяща и за оценяване.

Междупредметни връзки: БЕЛ; английски език; математика; изобразително изкуство

ТЕМА НА УРОКА:

27. ЦИКЛИЧНИ КОНСТРУКЦИИ И ЦИКЛИЧНИ АЛГОРИТМИ. ОБОБЩЕНИЕ

Брой учебни часове: 1

Вид на урока: Обобщение (О)

Опорни знания и умения: циклични алгоритми и циклични конструкции; видове цикли и тяхното приложение

Основна цел на урока: Обобщение на знанията за циклични конструкции, циклични алгоритми и тяхното приложение.

Дидактически материали: учебник, компютър, работни листове, ресурсен сайт към учебника

Очаквани трудности и предложения за преодоляването им: различна скорост на изпълнение на задачите от различните ученици. Проблемът може да се преодолее с използване на допълнителни задачи от отделените за самостоятелна работа.

Междупредметни връзки: ИТ, математика

Ход на урока:

• ОСНОВНИ ЦИКЛИЧНИ КОНСТРУКЦИИ

Обобщават се знанията посредством приложената таблица.

Задача 1 се препоръчва за работа в клас. Чрез нея се демонстрират различията в използването на различните циклични конструкции.

а) $i=5; num=11$

б) $i=6; num=7$

в) $i=5; num=1$

Задача 2 цели да се приложат знанията за цикличен оператор с брояч при намиране на произведението на първите n четни числа.

• ВЛОЖЕНИ ЦИКЛИ

След обобщение на знанията за вложени цикли се решава **задача 3**. Обсъжда се предложеното решение и се коментират възможностите за други начини за решаване на проблема.

Задача 4 предоставя възможност за прилагане на знанията в задача от практиката – тото 5 от 35. Предложено е едно решение, което може да се обсъди с учениците, като се изисква предложението за други възможности за решение.

Задача 5 се решава самостоятелно в час, като се очаква учениците да използват решените вече задачи. Изисква попълването на таблицата.

Задача 6 се решава в избраната програмна среда. Обсъжда се алгоритъмът и учениците работят самостоятелно в клас.

Обсъжда се възможността за приложение на операторите Break и Continue и се дават насоки за решение на задачите за самостоятелна работа в края на урока.

ТЕМА НА УРОКА:

28. ЦИКЛИЧНИ КОНСТРУКЦИИ И ЦИКЛИЧНИ АЛГОРИТМИ. КОНТРОЛ И ОЦЕНКА

Брой учебни часове: 1

Вид на урока: Контрол и оценка (КО)

Опорни знания и умения: циклични конструкции и алгоритми; видове цикли; приложение на цикличните алгоритми за решаване на задачи

Основна цел на урока: Контрол и оценка на знанията и уменията на учениците при работа с циклични конструкции и алгоритми.

Дидактически материали: учебник, работни листове, компютри

Дейности за контрол и оценка:

- Въпросите са от затворен тип с избираем отговор. Чрез тях се проверява нивото на знание и разбиране на учениците върху циклични алгоритми и конструкции.

- Подборът на задачите трябва да се съобрази с формулираните в учебната програма очаквани резултати.

- Решаването на практическа задача в този урок става на компютър в час. Този тип задачи може да съдържа отделни компоненти, които измерват усвояването на конкретни умения за работа с изучавания софтуер.

Верни отговори на въпросите от теста:

Въпрос 1 – б

Въпрос 2 – а

Въпрос 3 – в

Въпрос 4 – б

Въпрос 5 – б

Въпрос 6 – в

Въпрос 7 – б

Въпрос 8 – а

Въпрос 9 – б

Въпрос 10 – а

Задачите са със свободен отговор и се решават на работните листове.

Практическата задача се решава в избраната програмна среда.

Оценяване																
Въпроси	1	2	3	4	5	6	7	8	9	10						Общо
точки	1	1	1	1	1	1	1	1	1	1						10
Задачи											1	2	3	4		
точки											4	3	4	4		15
Практическа задача															1	
точки															15	20
скала	33 – 40 точки					Справям се отлично.										
	25 – 32 точки					Имам незначителни пропуски.										
	17 – 24 точки					Мога и по-добре.										
	10 – 16 точки					Трябва да положа още усилия.										
	0 – 9 точки					Не се справям.										

ТЕМА НА УРОКА:

29. ТЕСТВАНЕ И ВЕРИФИКАЦИЯ НА ПРОГРАМА

Брой учебни часове: 2

Вид на урока: Нови знания и упражнения (НЗУ)

Нови знания и умения: тестване и верификация; дефиниране на тестови данни и очаквани резултати от тестването при определени входни данни; използване на инструмент за откриване и отстраняване на грешки (debugger); приложение на процедури за тестване и верификация на вече създадени програми

Основна цел на урока: Усвояване на знания за тестване и верификация на програма. Въвеждане на понятията „тестване“, „тестови данни“, „debugger“, „процедури за тестване“ и „верификация“ и др.

КОМПЕТЕНТНОСТИ КАТО ОЧАКВАНИ РЕЗУЛТАТИ ОТ ОБУЧЕНИЕТО

- Обяснява и разграничава понятията „тестване“ и „верификация“.
- Дефинира тестови данни.
- Дефинира очаквани резултати от тестването при определени входни данни.
- Използва инструмент за откриване и отстраняване на грешки (debugger).

- Прилага процедури за тестване и верификация на вече създадени програми.

НОВИ ПОНЯТИЯ

тестване, верификация, тестови данни, debugger, процедури за тестване и верификация

Дидактически материали:

учебник, проектор, компютър с интернет достъп, сайт с ресурси към учебника.

План на урока:

1. Тестване. Дефиниране на тестови случаи.
2. Начини за търсене и намиране на грешки
3. Проследяване на фрагменти от изпълняван код. Debugger
4. Стратегии при търсене и отстраняване на грешки.

Ход на урока:

В урока се разглеждат действията в етапа на тестване и верификация на готов или създаден от учениците програмен продукт. Урокът започва с дефиниране на някои основни понятия, следван от информация, свързана с избор на входни данни, за които се очаква програмата да работи коректно, намиране на гранични случаи с входни данни, за които се очаква програмата да се справи при нормална работа, и включване на случаи с входни данни, за които се очаква програмата да не успее да се справи.

Предложени са задачи, които да помогнат на учениците да усвоят новите понятия, като избират и определят тестови случаи в познати за тях ситуации.

След дефиниране на входните данни, необходими за тестване на програмата, трябва да се прецени дали получените резултати са правилни. Понякога очакваният резултат може да се провери чрез пресмятане на ръка, но ако данните и случаите са много, се коментира с учениците, че е разумно да се напише програма, която проверява дали изходните стойности отговарят на определени свойства.

Следващата част на урока е свързан с начините за търсене и откриване на грешки. Дебъгване или откриване и отстраняване на грешки се прави след изпълнение на успешен тестов случай и се състои от две стъпки, първата е откриване на грешка в програмата, втората стъпка е отстраняване на грешката. Коментират се различните начини за търсене и откриване на грешки – с използване на Assert и проследяване на изпълнението на отделни фрагменти от програмата. В учебника са предоставени примери, които могат да се използват за изясняване на новите понятия в тази част на урока. В хода на работата с тестовите случаи и основните стъпки в процеса на тестване и верификация се изяснява понятието „debugger“. Използвайте програмната среда, с която работите, за да се демонстрира работата на debugger.

Изясняват се различните стратегии при търсене и откриване на грешки в кода и се коментират някои ограничения на дебъгъра.

В следващата част на урока се очаква от учениците да изберат програма от ресурсния сайт към учебника и да планират, проектират и извършат тестването ѝ.

Урокът завършва с рефлексия.

Очаквани трудности и предложения за преодоляването им: Учениците често изпитват трудности със създаването на тестови примери. Това може да се преодолее чрез работа по групи, задаване на подпомагащи въпроси и тестване на програми с познати за учениците ситуации (например математически задачи за откриване на максимален елемент, проверка за четност, нечетност и т.н.).

Междупредметни връзки: математика, логика, ИТ, английски език, български език и литература

Предварителна подготовка:

Познаване на средата за програмиране, основните операции при конструиране на алгоритми, етапите на създаване на програма, познаване на някои елементарни математически операции и функции.

ТЕМА НА УРОКА:

30. СЪСТАВНИ ТИПОВЕ ДАННИ.

ЕДНОМЕРЕН МАСИВ

Брой учебни часове: 2

Вид на урока: Нови знания и упражнения (НЗУ)

КОМПЕТЕНТНОСТИ КАТО ОЧАКВАНИ РЕЗУЛТАТИ ОТ ОБУЧЕНИЕТО

- Разбира необходимостта от използване на масиви.
- Идентифицира елементите на масив.
- Разпознава индекс и стойност на елемент на масив.
- Дефинира масив със средствата на език за програмиране.
- Създава и инициализира масив със средствата на език за програмиране.
- Осъществява достъп до елемент на масив.
- Обхожда, въвежда и извежда стойностите на елементите на масив.
- Използва списъчно поле за извеждане на стойностите на елементите на масив.
- Пресмята сбор и произведение на стойностите на елементите на едномерен масив.
- Търси елемент от масива с максимална и минимална стойност.
- Търси елементи от масива, отговарящи на дадено условие.

НОВИ ПОНЯТИЯ

логическо описание на масив; базов тип на масив, индекс и стойност на елемент от масив; последователно търсене

Дидактически материали: учебник, проектор, компютър с интернет достъп, сайт с ресурси към учебника

Ход на урока:

Темата се въвежда с две проблемни ситуации, в които се вижда потребността от съхраняването на редица от стойности с определена цел. В следващите уроци се решават задачите от тези предизвикателства и ученикът има възможност да ги реализира.

Разглежда се понятието „масив“, логическото му описание. Представят се чрез серия от примери деларация, задаване на стойности, обръщение, достъп до елементите, въвеждане и извеждане на масив.

Очаквани трудности и предложения за преодоляването им: не се очакват особени и специфични трудности по темата.

Междупредметни връзки: математика, ИТ

ТЕМА НА УРОКА:

31. СЪСТАВНИ ТИПОВЕ ДАННИ. ЕДНОМЕРЕН МАСИВ. ПРАКТИКА

Брой учебни часове: 2

Вид на урока: Практически дейности (П)

Опорни знания и умения:

– Дефинира масив със средствата на език за програмиране.

– Създава и инициализира масив със средствата на език за програмиране.

– Осъществява достъп до елемент на масив.

– Обхожда, въвежда и извежда стойностите на елементите на масив.

– Използва списъчно поле за извеждане на стойностите на елементите на масив.

Основна цел на урока: Усъвършенстване на умения за работа с едномерни масиви.

Дидактически материали: учебник, мултимедия проектор, електронен учебник, ресурсен сайт към учебника, компютър

Ход на урока:

За упражнение на темата едномерни масиви в този урок се създава приложение, обработващо данни за туристи. Върху данните се извършват статистически изчисления.

Представените данни са комбинирани с употре-

бата на контроли от вида на списъчна кутия, които са контейнери за данните, които се обработват. Съществен е и въпросът за визуализиране на елементите от списъчната кутия.

Следва задача, в която запълваме масив със случайни стойности. Необходимо е и провеждане на дискусия защо и в какви случаи е необходимо това.

От практическа гледна точка е необходимо да се научи и търсенето на елемент, отговарящ на дадено условие, което е следващият акцент в урока.

Представени са задачи, свързани с обръщане на елементите на масив, и различни реализации.

Задачи 7 и 8 могат да се изпълнят както в клас, така и за самостоятелна работа.

Дейности за контрол и оценка: по преценка на учителя.

Очаквани трудности и предложения за преодоляването им: темата всъщност обединява всички изучавани до момента теми и може евентуално да затрудни ученици с пропуски.

Междупредметни връзки: география и икономика, математика

ТЕМА НА УРОКА:

32. ОБРАБОТКА НА РЕДИЦИ ОТ ЕЛЕМЕНТИ. ОБОБЩЕНИЕ

Брой учебни часове: 1

Вид на урока: Обобщение (О)

Опорни знания и умения: едномерни масиви и редици от елементи; обработка на редица от елементи

Основна цел на урока: Обобщение на знанията за едномерни масиви и редица от елементи и тяхната обработка.

Дидактически материали: учебник, компютър, работни листове, ресурсен сайт към учебника

Очаквани трудности и предложения за преодоляването им: скоростта на работа е съобразена с основната група ученици. Има ученици, които се справят по-бавно, и други, които работят бързо. Предложеното решение е да се използват диференцирано задачите за самостоятелна работа.

Междупредметни връзки: ИТ, математика

Ход на урока:

• **ВЪВЕЖДАНЕ И ИЗВЕЖДАНЕ НА СТОЙНОСТИ НА ЕДНОМЕРЕН МАСИВ**

Задача 1. Работа по групи. Прави се проучване, което се систематизира и се представя на учениците от останалите групи.

• **ДОБАВЯНЕ И ИЗТРИВАНЕ НА ЕЛЕМЕНТИ ОТ ЕДНОМЕРЕН МАСИВ**

Коментират се възможностите за работа с елементите на едномерен масив. Разглежда се приложената таблица.

• **ПОДРЕЖДАНЕ И РАЗБЪРКВАНЕ**

Подреждането е необходимо, за да се търсят и намират елементите по-лесно или за да се извеждат данните по-читаеми за потребителя. Разбъркването е свързано с предизвикване на изненада. Дават се примери и се разглеждат приложените таблици.

Разчита се приложеният код, чрез който се реализира разбъркване на елементите.

Задача 2 се решава в час на ученическите компютри в избрана от тях програмна среда. Използват се приложените примерни вход и изход.

Задача 3 се решава самостоятелно, като се използва приложеният план на решение и примерните данни за вход и изход.

• **ТЪРСЕНЕ (ПОСЛЕДОВАТЕЛНО ИЛИ ДВОИЧНО)**

Задача 4. Решава се в клас. Обсъждат се различните предложения за тестови входни данни. Опишете особените случаи.

Самостоятелна и домашна работа: задачи 1, 2 и 3. Обсъждат се алгоритмите и особеностите при тяхното решаване и се поставят за самостоятелно решение.

ТЕМА НА УРОКА:

33. ОБРАБОТКА НА РЕДИЦИ ОТ ЕЛЕМЕНТИ. ПРАКТИКА

Брой учебни часове: 2

Вид на урока: Практически дейности (П)

Опорни знания и умения: едномерни масиви,

цикли, редици от елементи

Основна цел на урока: Усъвършенстване на умения по обработка на редица от елементи.

Дидактически материали: учебник, компютър, работни листове, ресурсен сайт към учебника

Ход на урока:

В урока се разглежда употребата на масива като помощна променлива.

Следва разглеждане на казуси, за които се дискутира необходимостта от използване на масиви.

Една особена цел в този урок е критично мислене по отношение на използване на структурата масив – осъзнаване на случаи на употреба.

Предложени са и няколко задачи, които може да се решат в клас или за самостоятелна работа.

Дейности за контрол и оценка: урокът е подходящ за оценяване.

Очаквани трудности и предложения за преодоляването им: заради аналитичния си характер задачите може да затрудняват някои ученици, предлага се на по-напредналите да работят по дадените задачи към урока.

Междупредметни връзки: математика, ИТ

ТЕМА НА УРОКА:

34. ОБЩИ ЗАДАЧИ ПО ПРОГРАМИРАНЕ. ПРАКТИКА

Брой учебни часове: 2

Вид на урока: Практически дейности (П)

Опорни знания и умения: условни и циклични конструкции, дебъгване, масиви, визуализиране на резултати

Основна цел на урока: Да илюстрира употребата на масиви при решаването на проблем за сортиране на стойности, използвайки линейни, условни и циклични оператори.

Дидактически материали: учебник, проектор, мултимедиен проектор, компютър, среда

за програмиране с дебъгър, интернет връзка за показване анимации на сортиращи алгоритми, като например <https://www.toptal.com/developers/sorting-algorithms>

Ход на урока:

С устна беседа за това как да се подходи към решаване на задача, използваща масиви и цикли за обхождане и манипулиране на елементите, показване на дебъгъра и неговите функции.

Дейности за затвърждаване на НЗУ:

- Учениците може да се накарат да изпълнят програмата, най-напред без дебъгър, после и с дебъгър. Може да пробват няколко различни сортирания, включително като подберат входните данни по различен начин.

- Програмата може да се промени така, че да сортира в обратен ред, да се сортира част от редицата.

- Предложили сме две задачи, едната за визуализиране на сортирането, другата за сортиране на стрингове и подредба по азбучен ред въведените думи.

Дейности за контрол и оценка: по преценка на учителя. Подходящи за оценка са задачите за модифициране на програмата за сортиране, която да сортира в обратен ред или да работи с различен тип данни.

Очаквани трудности и предложения за преодоляването им: различните опции на дебъгъра може да създадат известни трудности.

Междупредметни връзки: БЕЛ, английски език, математика

Допълнителни въпроси, задачи и упражнения:

По желание може да се отдели време за пъзели и игри, като например се направи задачата за играта на Симон.

ТЕМА НА УРОКА:

35. ОБРАБОТКА НА РЕДИЦИ ОТ ЕЛЕМЕНТИ. КОНТРОЛ И ОЦЕНКА

Брой учебни часове: 1

Опорни знания и умения: редици от елементи; едномерен масив; приложение

Основна цел на урока: Контрол и оценка на знанията и уменията на учениците при работа с редици от елементи.

Дидактически материали: работни листове; компютри с интернет достъп и възможност за работа в програмна среда

Дейности за контрол и оценка:

- Тестовите въпроси с избираем отговор и имат за цел да проверят базовите знания на учениците по темата.

- Задачите проверяват нивото на умения за прилагане на получените знания. Те трябва да се съобразят с формулираните в учебната програма очаквани резултати.

- Решаването на практическата задача в този урок става на компютър в час. Предложеният вариант съдържа отделни компоненти и подзадачи, които могат да бъдат оценявани по отделно.

Верни отговори на въпросите от теста:

Въпрос 1 – в

Въпрос 2 – б

Въпрос 3 – г

Въпрос 4 – б

Въпрос 5 – г

Въпрос 6 – г

Въпрос 7 – г

Въпрос 8 – а

Въпрос 9 – б

Въпрос 10 – а

Задача 1: br=2

Задача 2: Сборът на нечетните стойности на масива

Задачи 3 и 4 се решават на хартия, като про-

грамните фрагменти се записват на избрания от учениците програмен език.

Практическата задача се решава в избраната програмна среда на компютър.

Оценяване															
Въпроси	1	2	3	4	5	6	7	8	9	10					Общо
точки	1	1	1	1	1	1	1	1	1	1					10
Задачи												1	2	3	4
точки												4	3	4	4
Практическа задача															1
точки															15
скала	33 – 40 точки					Справям се отлично.									
	25 – 32 точки					Имам незначителни пропуски.									
	17 – 24 точки					Мога и по-добре.									
	10 – 16 точки					Трябва да положа още усилия.									
	0 – 9 точки					Не се справям.									

IV. СОФТУЕРЕН ПРОЕКТ

ТЕМА НА УРОКА:

36. СЪЗДАВАНЕ НА СОФТУЕРЕН ПРОЕКТ

Брой учебни часове: 2

Вид на урока: Нови знания и упражнения (НЗУ)

КОМПЕТЕНТНОСТИ КАТО ОЧАКВАНИ РЕЗУЛТАТИ ОТ ОБУЧЕНИЕТО

Описва етапите при реализиране на софтуерен проект; извършва проучване и анализ на решения за даден групов проект; създава модел за решаване на проблема, поставен в заданието на проекта; проектира графичен потребителски интерфейс; създава програмен код за реализация на модела; създава тестови примери с входни данни и очаквани резултати; изготвя документация за софтуерния проект; презентира и защитава готовия софтуерен проект.

Дидактически материали: учебник, проектор, електронен учебник, компютър

Ход на урока:

Представят се етапите при реализиране на софтуерен проект, които надграждат теоретично етапите на решаване на конкретна задача в предходния раздел.

Описват се две методологии на създаване на софтуерен проект и се дават прости примери за демонстрацията им.

В урока е подходящо да бъдат пуснати видеа с мнения на експерти, описващи дейността си.

Дава се пример за създаване на софтуерен проект за часовник, който следва да се създаде групово и да се дискутира всеки етап от създаването му. За целта е препоръчително предварителна подготовка на проекта и наличието му като файлове в отделните фази с цел по-ефективна демонстрация и реализация на задачата.

Очаквани трудности и предложения за преодоляването им: за изпълнение на повече видове дейности в урока се препоръчва демонстрация на видео по част от обясненията.

Междупредметни връзки: изобразително изкуство

ТЕМА НА УРОКА:

37. СЪЗДАВАНЕ НА СОФТУЕРЕН ПРОЕКТ. ПРАКТИКА

Брой учебни часове: 2

Вид на урока: Практически дейности (П)

Опорни знания и умения: софтуерен проект, етапи на реализация на софтуерен проект

Основна цел на урока: Описва етапите при реализиране на софтуерен проект; извършва проучване и анализ на решения за даден групов проект; създава модел за решаване на проблема, поставен в заданието на проекта; проектира графичен потребителски интерфейс; създава програмен код за реализация на модела; създава тестови примери с входни данни и очаквани резултати.

Дидактически материали: учебник, проектор, електронен учебник, компютър

Ход на урока:

Описват се целите на създаване на биоритмичен хороскоп и същността му.

Разясняват се отделните етапи, които се изпълняват в час заедно.

Препоръчва се на ученика да изпълни самостоятелно проекта.

Дейности за контрол и оценка: не се предвиждат в този урок.

Очаквани трудности и предложения за преодоляването им: необходимост от допълнителни задачи, при което се препоръчва проследяване на ресурсния сайт за ресурси.

Междупредметни връзки: ИТ

ТЕМА НА УРОКА:
38. СЪЗДАВАНЕ НА СОФТУЕРЕН ПРОЕКТ. ПРАКТИКА

Брой учебни часове: 2

Вид на урока: Практически дейности (П)

Опорни знания и умения: софтуерен проект, етапи на реализация на софтуерен проект

Основна цел на урока: Извършва проучване и анализ на решения за даден групов проект; създава модел за решаване на проблема, поставен в заданието на проекта; проектира графичен потребителски интерфейс; създава програмен код за реализация на модела; създава тестови примери с входни данни и очаквани резултати.

Дидактически материали: учебник, проектор, електронен учебник, компютър

Ход на урока:

В този урок се предлага на учениците работа по три проекта.

Първоначално може да се изпълнят, за да се види действието им, след което да се проследят част от етапите за изпълнението им, свързани с разработката на софтуерния проект, и накрая да се направят вариации и модификации на проекта. Ученикът отговаря и на поставени въпроси.

Дейности за контрол и оценка: по преценка на учителя.

Очаквани трудности и предложения за преодоляването им: с цел изясняване на детайлите препоръчваме използване на проект, създаден до отделните етапи, който да е готов след всяка стъпка, за да може да се гарантира ефективна работа в часа.

Междупредметни връзки: БЕЛ; ИТ

ТЕМА НА УРОКА:
39. СЪЗДАВАНЕ НА СОФТУЕРЕН ПРОЕКТ. ПРАКТИКА

Брой учебни часове: 2

Вид на урока: Практически дейности (П)

Опорни знания и умения: софтуерен проект, етапи на реализация на софтуерен проект

Основна цел на урока: Описва етапите при реализиране на софтуерен проект; извършва проучване и анализ на решения за даден групов проект; създава модел за решаване на проблема, поставен в заданието на проекта; създава програмен код за реализация на модела; създава тестови примери с входни данни и очаквани резултати; изготвя документация за софтуерния проект.

Дидактически материали: учебник, проектор, електронен учебник, компютър

Ход на урока:

В този урок се предлага на учениците създаването на един проект, при реализацията на който се използват файлове. Препоръчваме първоначалното му стартиране, за да се види действието им, след което да се проследят част от етапите за изпълнението им, свързани с разработката на софтуерния проект, и накрая да се направят вариации и модификации на проекта. Ученикът отговаря и на поставени въпроси.

Дейности за контрол и оценка: по преценка на учителя.

Очаквани трудности и предложения за преодоляването им: с цел изясняване на детайлите препоръчваме използване на проект, създаден до отделните етапи, който да е готов след всяка стъпка, за да може да се гарантира ефективна работа в часа.

Междупредметни връзки: БЕЛ; математика; география и икономика

ТЕМА НА УРОКА:

40. СЪЗДАВАНЕ НА СОФТУЕРЕН ПРОЕКТ. ПРАКТИКА

Брой учебни часове: 2

Вид на урока: Практически дейности (П)

Опорни знания и умения: софтуерен проект, етапи на реализация на софтуерен проект

Основна цел на урока: Описва етапите при реализиране на софтуерен проект; извършва проучване и анализ на решения за даден групов проект; създава модел за решаване на проблема, поставен в заданието на проекта; създава програмен код за реализация на модела; създава тестови примери с входни данни и очаквани резултати; изготвя документация за софтуерния проект; презентира и защитава готовия софтуерен проект.

Дидактически материали: учебник, проектор, електронен учебник, компютър

Ход на урока:

В този урок се предлага на учениците работа по три проекта.

Първоначално може да се изпълнят, за да се види действието им, след което да се проследят част от етапите за изпълнението им, свързани с разработката на софтуерния проект, и накрая да се направят вариации и модификации на проекта. Ученикът отговаря и на поставени въпроси.

Дейности за контрол и оценка: по преценка на учителя. Подходящи за оценка са практически екипни и индивидуални задачи.

Очаквани трудности и предложения за преодоляване им: различна скорост на работа на учениците, предвидени са няколко проекта, урокът позволява всеки ученик да реализира задачите със собствения си темп.

Междупредметни връзки: БЕЛ; Английски език

ТЕМА НА УРОКА:

41. СЪЗДАВАНЕ НА СОФТУЕРЕН ПРОЕКТ. КОНТРОЛ И ОЦЕНКА

Брой учебни часове: 1

Вид на урока: Контрол и оценка (КО)

Опорни знания и умения: описва етапите при реализиране на софтуерен проект; извършва проучване и анализ на решения за даден групов проект; създава модел за решаване на проблема, поставен в заданието на проекта; проектира графичен потребителски интерфейс; създава програмен код за реализация на модела; създава тестови примери с входни данни и очаквани резултати.

Основна цел на урока: Контрол и оценка на знанията и уменията на учениците, свързани със създаването на софтуерен проект.

Дидактически материали:

- работни листове
- компютри с интернет достъп и възможност за използване на програмна среда

Дейности за контрол и оценка:

- Тестовите въпроси са с избираем отговор и чрез тях се проверяват знанията на учениците за софтуерното моделиране.
- Задачите са с разглеждане на готов софтуерен продукт и дописване или коригиране на код, както и със създаване на програмни фрагменти.
- Решаването на практическата задача в този урок става на компютър с инсталирана програмна среда в час. Може да се раздели на отделни компоненти, които да се оценяват независимо от останалите.

Верни отговори на въпросите от теста:

Въпрос 1: а

Въпрос 2: а

Въпрос 3: б

Задача 1 се решава, като се отваря готов проект от интернет сайта към учебника с добавяне на код по аналогия с вече написания.

Задача 2 е аналогична на задача 1. В нея се очаква учениците да направят оценка на готов софтуерен продукт и да направят предложение за неговото оптимизиране и подобряване.

Задача 3 се решава на работните листове.

Практическата задача се решава на ученическия компютър в избрана от ученика програмна среда.

Оценяване															
Въпроси	1	2	3												Общо
точки	2	2	2												6
Задачи				1	2	3									
точки				4	5	5									14
Практическа задача							1								
точки							20								20
скала	33 – 40 точки			Справям се отлично.											
	25 – 32 точки			Имам незначителни пропуски.											
	17 – 24 точки			Мога и по-добре.											
	10 – 16 точки			Трябва да положа още усилия.											
	0 – 9 точки			Не се справям.											

42. ИЗХОДЯЩО РАВНИЩЕ

Брой учебни часове: 1

Опорни знания и умения:

- Представяне на информация във вид на данни
- Основни етапи при създаване и изпълнение на компютърна програма
- Обектно-ориентиран подход при създаване на несложна компютърна програма
- Библиотеки от готови компоненти
- Добър стил на програмиране
- Разбиране и използване на основни компоненти на среда за визуално програмиране при разработка на софтуер
- Проектиране на графичен потребителски интерфейс с визуални средства
- Избор на подходяща графична компонента в съответствие с необходимата функционалност на графичния интерфейс

- Настройка на свойствата на графичните компоненти
- Програмиране на подразбиращи се събития за основни компоненти от графичния интерфейс
- Разграничаване на различни типове данни и определяне на подходящ тип данни за определена задача
- Същност на алгоритмите и начини за описанието им
- Приложение на основни управляващи конструкции
- Структуриране на данни в едномерен масив и прилагане на основни алгоритми за намиране на сума, минимален/максимален елемент и средно аритметично
- Четене и запис на данни в текстов файл
- Създаване на програми за графично изобразяване на геометрични обекти със стандартни средства в езика

Основна цел на урока: Контрол и оценка на знанията, уменията и компетентностите на учениците по информатика в 8. клас.

Дидактически материали:

- работни листове
- компютри с интернет достъп и възможност за използване на програмна среда.

Дейности за контрол и оценка:

- Тестовите въпроси са с избираем отговор. При някои от тях се изисква предварително извършване на пресмятания или други дейности. Чрез тези въпроси се проверяват знанията на учениците върху учебното съдържание по информатика в 8. клас.
- Задачите в теста са съобразени с изучаваните алгоритми и чрез тях се проверява нивото на умения на учениците.
- Решаването на практическата задача става на компютър с инсталирана програмна среда в час. Тя съдържа отделни компоненти, които измерват усвояването на конкретни умения.

Верни отговори на въпросите от теста:

- Въпрос 1: г
- Въпрос 2: б
- Въпрос 3: г
- Въпрос 4: б
- Въпрос 5: б
- Въпрос 6: г
- Въпрос 7: б
- Въпрос 8: в
- Въпрос 9: б
- Въпрос 10: в
- Въпрос 11: а
- Въпрос 12: г

Задачи 1 и 2 – да се напишат програмни фрагменти на работните листове без използването на компютър.

Практическата задача се реализира на ученическите компютри в избрана от тях програмна среда. Оценка може да се реализира за всеки отделен модул по преценка на учителя.

Оценяване														
Въпроси	1	2	3	4	5	6	7	8	9	10	11	12		Общо
точка	1	1	1	1	1	1	1	1	1	1	1	1		12
Задачи													1	2
точка													6	6
Практическа задача														1
точка														16
скала	33 – 40 точки					Справям се отлично.								
	25 – 32 точки					Имам незначителни пропуски.								
	17 – 24 точки					Мога и по-добре.								
	10 – 16 точки					Трябва да положа още усилия.								
	0 – 9 точки					Не се справям.								

V. ДЕЙНОСТИ ЗА ПРИДОБИВАНЕ НА КЛЮЧОВИ КОМПЕТЕНТНОСТИ (МЕЖДУПРЕДМЕТНИ ВРЪЗКИ)

Дейностите за придобиване на ключови компетентности съпътстват целия процес на обучение по информатика в 8. клас, поради което в таблицата по-долу са описани само някои примерни уроци, отговарящи на съответните дейности.

Дейност 1	
Проучване и представяне на допълнителна информация, свързана с изучаваните теми. Учениците развиват умения за учене, социални и граждански компетентности, създават се междупредметни връзки с български език и литература.	
Урок 1	<ul style="list-style-type: none"> – представяне на проучване във вид на диаграма за различните направления в компютърните науки (БЕЛ); – проучване на връзката между информационното общество и умните градове (философия); – разучаване на съвременни интерпретации за Розетския камък (история и цивилизация, БЕЛ); – проучване за съвременни компютърни архитектури и връзка на изучавания материал. Например квантови компютри и квантови изчисления – bit срещу qubit (ИТ, физика и астрономия); – търсене по изображение – личности с принос в областта (ИТ); – представяне на информационни процеси от житейския ни опит и откриване на връзка и модел в тях (всички учебни предмети); – проучване за примери от практиката за откриване на основните алгоритми и техните описания – словесно, звуково, визуално, мултимедийно (БЕЛ); – търсене на специфична информация от история на компютрите.
Урок 2	<ul style="list-style-type: none"> – проучване и представяне на информация за стилове на учене и обработка на информация; – проучване и сравнение на представяне на числата на египтяни, май, шумери и вавилонци; – проучване на разликите между кодовите таблици ASCII и UNICODE; – проучване и представяне на информация за парадокса на котката на Шрьодингер (Schrodinger cat).
Дейност 2	
Структуриране на статистическа информация във вид на данни в контекста на конкретна информационна задача. На учениците се дава възможност да демонстрират социални и граждански компетентности, както и компетентности в областта на българския език.	
Урок 30	– статистики на игри при мотивиране за изучаване на масиви
Урок 31	– моделиране и реализация на приложение за обработка на данни и статистики за месечно посещение на туристи в туристически обект
Дейност 3	
Идентифициране и дефиниране на стандартите в информатиката и използване на конвенция за именуване. Учениците демонстрират дигитални компетентности и умения за общуване на чужди езици.	
Урок 12	– изучаване и прилагане на стилове на именуване на константи и променливи
Дейност 4	
Редактиране, компилиране и изпълнение на готова програма, създадена за различни сфери на обществения живот. Учениците развиват дигитални, математически компетентности и основни компетентности в областта на природните науки и на технологиите, социални и граждански компетентности. Осъществяват се междупредметни връзки с природни науки, информационни технологии и математика.	
Урок 5	– компилиране, изпълнение, редактиране, тестване на приложение за триъгълник (математика)
Урок 17	– проектиране на потребителския интерфейс на приложение планетите в Слънчевата система и реализация на логиката му

Дейност 5	
Редактиране на дизайна на приложение с графичен потребителски интерфейс във визуална среда за програмиране. Реализират се междупредметни връзки на информационните технологии и изобразителното изкуство – цвят, цветови схеми, елементи на интерфейса и др.	
Урок 6	– компилиране, изпълнение, редактиране, тестване на приложение за молекулна маса на веществата (химия и опазване на околната среда)
Урок 36	– проектиране на потребителски интерфейс и избор на цветови схеми
Дейност 6	
Създаване на математически модели на ситуации, изучавани в часовете по математика и природни науки, като се използват средствата на програмирането. Учениците демонстрират математически и дигитални компетентности в областта на природните науки и на технологиите.	
Урок 14	– моделиране на изчисления по теми от комбинаторика
Урок 28	– моделиране на алгоритъм за изчисляване на брой пермутации
Дейност 7	
Настройки на основните свойства на графичните компоненти на приложение. Създаване на програма за изчисляване на печалбата на дадено предприятие, общия приход (разход) на домакинство. Демонстрира се компетентността инициативност и предприемчивост.	
Урок 7	– приложение за изчисляване на разход за керемиди (математика, технологии и предприемачество)
Урок 25	– проектиране на графичен потребителски интерфейс и реализация на приложение за кредитен калкулатор
Дейност 8	
Описване на алгоритми за решаване на линейни уравнения, линейни неравенства и сравняване на отсечки и ъгли. Учениците развиват дигитални, математически компетентности и основни компетентности в областта на природните науки.	
Урок 22	– проектира и реализира приложение за намиране на корените на квадратно уравнение; – проектира и реализира самостоятелно приложение за намиране на решението на линейно уравнение.
Дейност 9	
Въвеждане и извеждане на данни в приложения, свързани в природо-математическите дисциплини, бита и бизнеса. Класифициране на типовете данни. Учениците развиват дигитални, математически компетентности и основни компетентности в областта на природните науки и на технологиите, социални и граждански компетентности. Осъществяват се междупредметни връзки с природните науки, информационните технологии и математиката.	
Урок 12	– запознаване с основните типове данни
Урок 20	– въвеждане и обработка на данни за фирми и лицензи
Урок 30	– моделира и реализира програмно съставни типове данни
Дейност 10	
Въвеждане и извеждане на данни от тип низ в текстово поле. Прилагат граматически правила в българския език за конструиране на изречение, работят с измерителни единици. Осъществяват се междупредметни връзки с български език и литература, математика, природни науки.	
Урок 12	– въвеждане на стойности в текстови полета в програма за конвертиране на стойности от една мерна единица в друга (математика, физика и астрономия)
Урок 40	– разработка на приложение за генериране на забавен текст (БЕЛ); – разработка на приложение за неправилни глаголи в английски език (ЧЕ)

Дейност 11	
Създаване на аритметични изрази, съдържащи различни типове данни, отчитайки съвместимостта им. Осъществяване на междупредметни връзки с математиката (пресмятане на числена стойност на израз, разлагане на множители), физиката (определяне на пътя и скоростта при равноускорително движение и свободно падане).	
Урок 12	– аритметични изрази за преобразуване на температура (Целзий, Фаренхайт, Келвин); – аритметичен израз за изчисляване на кинетична енергия.
Дейност 12	
Реализиране на несложни алгоритми със средствата на език за програмиране. Учениците развиват дигитални, математически компетентности и основни компетентности в областта на природните науки и на технологиите, социални и граждански компетентности.	
Урок 19	
Урок 23	– реализиране на разклонени алгоритми
Урок 24	
Урок 28	– реализиране на циклични алгоритми
Дейност 13	
Реализиране на програми с графично изобразяване на геометрични обекти и комбинация от обекти. Осъществяват междупредметна връзка с математика – създаване на програма за визуализация на задачи, свързани с пресмятане на точки на пресичане на геометрични обекти (отсечки, окръжности), и изобразяване на траектория, описана от точка, средна отсечка, триъгълник, симетрала.	
Урок 26	– изобразява геометрични фигури; – създава графични 2D композиции от графични промитиви с използване на език за програмиране.
Урок 37	– създаване на приложение за биоритмичен хороскоп
Дейност 14	
Пресмятане на стойността на булеви изрази чрез език за програмиране, обсъждане на съждения и техните отрицания. Осъществяват междупредметна връзка с математика. Учениците развиват дигитални, математически компетентности и основни компетентности в областта на природните науки и на технологиите, социални и граждански компетентности.	
Урок 18	– оценка на съждения; – действия с логически операции; – пресмятане на стойности на булеви изрази; – прилагане на закона на Де Морган; – връзка на булеви изрази и условен оператор.
Дейност 15	
Реализиране на алгоритми с използване на повторение, управлявано от условие. Осъществя се междупредметна връзка с математика – създаване на алгоритъм за намиране на най-голямо (най-малко) от N числа, пресмятане на данни, чието въвеждане се прекратява от условие. Реализиране на програми за определяне на вида на триъгълник, определяне на дата по зададено условие, определяне на процент, намиране на най-висока/най-ниска температура в определен период от време или други задачи от изучавани теми. Учениците развиват дигитални, математически компетентности и основни компетентности в областта на природните науки и на технологиите, социални и граждански компетентности.	
Урок 24	– реализация на алгоритми с повторения за изчисляване на среден, минимален/максимален успех; – изследователски проект Хейлстонови числа.

Урок 27	– отпечатване на комбинации от тотото
Дейност 16	
Събиране на информация от различни източници. Трансформиране на информацията във формат на данни. Въвеждане и използване на стойностите в структура масив. Използване на данни за валежи, температура, топлинно замърсяване, търсене на средна стойност, минимален и максимален елемент. Учениците развиват дигитални, математически компетентности и основни компетентности в областта на природните науки и на технологиите, социални и граждански компетентности.	
Урок 12	– минимум/максимум с функции/методи и изрази
Урок 32	– минимален/максимален елемент в масив
Урок 33	– минимален/максимален елемент в редица от елементи
Дейност 17	
Създаване на самостоятелно софтуерно приложение. Учениците разработват проект в екип по двама. Развиват се граждански компетентности – форми на поведение за успешно участие в социалния живот, разрешаване на конфликти и проблеми, умения за самостоятелно учене и събиране на информация. Разработват се общественополезни и екологични теми – топлинно и химично замърсяване, замърсяване на околната среда, влияние на различни фактори върху човешкото здраве.	
Урок 37	– самостоятелно създаване на приложение с използване на шестте мислещи шапки
КУ	– идеи за софтуерни проекти (от книга за учителя, публикувани и в сайта)

VI. ИДЕИ ЗА СОФТУЕРНИ ПРОЕКТИ

В тази част от книгата за учителя са посочени идеи за софтуерни проекти, които отговарят на спецификата на възрастта на обучаемите и на изучаваните теми по другите учебни предмети. Те са допълнение на вече използваните проекти с междудисциплинарни връзки в самите уроци и може да бъдат използвани за самостоятелна работа от ученика през цялата година, като може да се моделират изчисления, процеси, а някои от тях и да се реализират като софтуерни проекти.

	СОФТУЕРЕН ПРОЕКТ идеи за проекти	МЕЖДУПРЕДМЕТНИ ВРЪЗКИ	
		предмет/и	тема/понятие
1	Basal Metabolic Rate (BMR) количеството енергия, което човек изразходва, докато почива в неутрална среда и температура.	Биология и здравно образование	Обмяна на веществата
2	Тест за разпознаване на причини – признаци – превенция	Биология и здравно образование	Полова система Заболявания, предавани по полов път
3	Топлообмен Анимация/симулация	Физика и астрономия	Топлинни явления
4	Видове движения Разпознаване на видове движения, дадени със серия от изображения	Физика и астрономия	Дигитална компетентност
5	Видове вълни Разпознаване на видове вълни, дадени със серия от изображения	География и икономика	География на природата
6	Строеж на атома Анимация/симулация	Химия и опазване на околната среда	Строеж на веществото
7	Мотиватор за събиране на мотиви за избор на кариера	Технологии и предприемачество	Професия и кариера
8	Емпатия	Философия	Междупersonностни отношения
9	Тест за емоционална интелигентност	Философия	Междупersonностни отношения
10	Генеративно изкуство	Изобразително изкуство Музика	Дигитални изкуства в съвременното
11	Калории на ден Изчислява погълнатите с храни калории дневно. Използва информация за калориите от етикетите на стоките.	Биология и здравно образование	Здравословен начин на живот
12	Countdown до класното	Математика	Моделиране
13	Бързина на реакциите	Философия	Личност и себепознание
14	Тегло на друга планета	Физика и астрономия	Механика
15	Конвертор за час между две часови зони	География и икономика	Планетата Земя
16	Ренесанс Тест за познаване на автори от Ренесанса	Български език и литература Изобразително изкуство Музика	Ренесанс
17	Шестнадесетичното време за представянето на времето с помощта на шестнадесетични числа в интервала $[0, 1)$	Математика	Моделиране

ПРИМЕРНО ГОДИШНО РАЗПРЕДЕЛЕНИЕ НА ТЕМИТЕ ПО ИНФОРМАТИКА ЗА 8. КЛАС

Учебна седмица по ред	Тема на урочна единица	Урочна единица	Компетентности като очаквани резултати от обучението	Нови понятия	Контекст и дейности за всяка урочна единица	Методи и форми на оценяване по теми и/или раздели	Забележка
ОСНОВИ НА ИНФОРМАТИКА							
1	1. Основи на информатиката 1.1. Информатика и компютърни науки	НЗУ – 2	Изброява и описва основните информационни дейности събиране, съхраняване, преработка и разпространение на информация и общата схема на информационните потоци. Описва понятието информационен процес и дава примери на информационни процеси, свързани с решаване на житейски задачи. Различава понятията информация и данни. Обяснява и илюстрира с примери връзката между информация и данни. Обяснява и илюстрира с примери същността на дискретното представяне на информацията за трансформирането ѝ в данни.	– основни информационни дейности – информация, данни – дискретно представяне на информацията	Актуализиране на знания за основните информационни дейности, информация, данни. Мотивиране и въвеждане на дискретното представяне на информацията за трансформирането ѝ в данни.	– писмено изпитване – тест (изчислителни задачи, избираем отговор и др.); – устно изпитване (описва алгоритъм с думи, посочване на очаквани грешки (синтактични, логически) проследяване на алгоритъм и др.); – практическо изпитване (задачи за пресмятане, моделиране, практическа задача).	
	1.2. Числата и техните представяния	НЗ	Разширява и обобщава знанията, свързани с числата и техните представяния: – непозиционни бройни системи; – същност на позиционните бройни системи; – формат на числата в десетична, двоична и шестнадесетична бройна система. Превърща числа от десетична в двоична бройна система и обратно. Извършва събиране, изваждане и умножение на две числа в двоична бройна система. Дава примери за използване на двоична и шестнадесетична бройна система.	– непозиционни бройни системи – позиционни бройни системи – двоична бройна система – шестнадесетична бройна система – експоненциален формат	Актуализиране на знания, свързани с числата и техните представяния. Въвеждане на понятията: – непозиционни бройни системи; – позиционните бройни системи; – формат на числата в десетична, двоична и шестнадесетична бройна система; – експоненциален формат. Формиране на умения за превръщане на числа от десетична в двоична бройна система и обратно. Формиране на умения за събиране, изваждане и умножение на две числа в двоична бройна система.		

Учебна седмица по ред	Тема на урочна единица	Урочна единица	Компетентности като очаквани резултати от обучението	Нови понятия	Контекст и дейности за всяка урочна единица	Методи и форми на оценяване по теми и/или раздели	Забележка
	1.3. Алгоритми и езици за програмиране		Дефинира понятието алгоритъм и описва основните му характеристики (результатност, крайност, детерминираност, масовост). Описва и проследява несложни, линейни и разклонени алгоритми с различни средства. Описва същността, структурата и разновидностите на цикличните алгоритмични конструкции. Обяснява същността и функционалното предназначение на език за програмиране. Обяснява същността и предназначението на транслятор (интерпретатор, компилатор). Представа исторически факти, свързани със създаването и развитието на съвременните езици и среди за програмиране.	– алгоритъм – език за програмиране – транслятор	Ввеждат се на понятието алгоритъм, основните му видове и характеристики. Формиране на умения за описание на алгоритми с различни средства. Запознаване с историята, същността и функционалното предназначение на съвременни езици за програмиране.		
2	2. Основи на информатиката. Практика	П – 2	Усъвършенстване на уменията за информационен процес, бройни системи, преобразуване на числата и свойства и представяне на алгоритми.		Затвърждаване на понятията информационен процес, бройни системи и информационни дейности. Усъвършенстване на умения за представяне на числа; преобразуване на числа от една бройна система в друга; описание на алгоритми.		
3	3. Основи на информатиката. Обобщение	0 – 1	Обобщение на знанията за информационен процес, бройни системи, преобразуване на числата и свойства и представяне на алгоритми.		Обобщаване на знания и умения за информационен процес и информационни дейности; представяне на числата; бройни системи; преобразуване на числа от една бройна система в друга; описание на алгоритми.		
3	4. Основи на информатиката. Контрол и оценка	К0 – 1	Контрол и оценка на знанията и уменията на учениците.		Затвърждаване на знания и умения. Оценяване на знания и умения. Рефлексия.		

Учебна седмица по ред	Тема на урочна единица	Урочна единица	Компетентности като очаквани резултати от обучението	Нови понятия	Контекст и дейности за всяка урочна единица	Методи и форми на оценяване по теми и/или раздели	Забележка
СРЕДА ЗА ВИЗУАЛНО ПРОГРАМИРАНЕ							
4	5. Среда за визуално програмиране	НЗУ – 2	Отваря проект в интегрирана среда за визуално програмиране. Стартира приложение с графичен потребителски интерфейс чрез средствата на интегрирана среда за програмиране. Разпознава основните компоненти на интегрирана среда за програмиране – графичен и текстов редактор, панел с контроли, панел за свойства на обект, панел за съобщения, панел за преглед на структурата на приложението.	– интегрирана среда за програмиране – свързващ редактор (linker) – програма за откриване и отстраняване на грешки (debugger) – редактор за проектиране на дизайн на графичен потребителски интерфейс – синтактични грешки в компютърна програма – логически грешки в компютърна програма – грешки по време на изпълнение на програмата	Въведяне на понятието интегрирана среда за програмиране и основните ѝ компоненти.	– писмено изпитване – тест (открива грешки в проектиране, задава свойства на обекти от потребителски интерфейс, описва тестови случаи; прилага и създава протоколи от оценка на потребителски интерфейс на приложение); – усстно изпитване (обяснява типичната употреба на контроли в приложение, обяснява проблем, обяснява проблем, за да потърси помощ); – практически изпитване (изпълнява практически задачи по проектиране на потребителски интерфейс, редактира потребителския интерфейс на приложение с цел подобряването му).	
5	6. Интегрирана среда за визуално програмиране. Практика	П – 2	Редактира дизайна на графичния потребителски интерфейс на приложение в интегрирана среда за програмиране. Запазва проект на приложение чрез средствата на интегрирана среда за програмиране.		Актуализиране на умения за анализ на задача с несложен математически модел. Формиране на умения за редактиране на дизайна на графичния потребителски интерфейс на приложение в интегрирана среда за разработка. Усъвършенстване на умения за определяне на подбор на компоненти и определяне на свойствата ѝ.		

Учебна седмица по ред	Тема на урочна единица	Урочна единица	Компетентности като очаквани резултати от обучението	Нови понятия	Контекст и дейности за всяка урочна единица	Методи и форми на оценяване по теми и/или раздели	Забележка
6	7. Основни етапи на създаване и изпълнение на компютърна програма. Практика	П – 2	<p>Анализира задача с несложен математически модел.</p> <p>Създава математически модел за решаване на несложна задача.</p> <p>Съпоставя математически модел с програмно решение на даден проблем.</p> <p>Открива основните компоненти на математически модел в демонстрирано програмно решение на даден проблем.</p> <p>Стартира чрез средствата на интегрирана среда предварително подготвена компютърна програма с графичен потребителски интерфейс.</p> <p>Тества предварително подготвен несложен проект.</p> <p>– Разпознава видовете грешки при програмиране.</p> <p>– Разчита и прави предположение за естеството на синтактична грешка в даден проект.</p> <p>– Открива и прави предположение за причината за логическа грешка в дадено приложение.</p> <p>– Открива и прави предположение за причината за грешка по време на изпълнение на приложението.</p>	<p>– графичен потребителски интерфейс</p> <p>– интерфейсен компонент (контрола)</p> <p>– контрол</p> <p>– свойство на обект</p> <p>– метод на обект</p>	<p>Актуализиране на умения за анализ на задача с несложен математически модел.</p> <p>Затвърждава понятията синтактична и логическа грешка, грешка по време на изпълнение на програмата.</p> <p>Формиране на умения за стартиране и тестване на готов проект.</p> <p>Затвърждаване на умения за разпознаване на грешки и причините за появяването им.</p>		
7	8. Проектиране на графичен потребителски интерфейс	НЗУ – 2	<p>Знае предназначението на основни контейнери и контроли – форма, етикет, текстово поле, бутон, диалогова кутия. Разпознава основни свойства на графични обекти – контроли – име, състояние, етикет, фон, настройка на шрифт и гр.</p> <p>Проектира несложна форма, съдържаща етикет, текстово поле, бутон.</p> <p>Настройва основни свойства на форма, етикет, текстово поле и бутон.</p> <p>Именува обекти – контроли съгласно общоприета конвенция.</p> <p>Задава функционалност на бутон, свързана с извеждането на статично съобщение в диалогова кутия.</p>	<p>– графичен потребителски интерфейс</p> <p>– интерфейсен компонент (контрола)</p> <p>– контрол</p> <p>– свойство на обект</p> <p>– метод на обект</p>	<p>Ввеждат се на понятията основни елементи на потребителски интерфейс (контейнери и контроли) и основни свойства на графични обекти.</p> <p>Формиране на умения за: проектиране на несложна форма, основни настройки, именуване на обекти – контроли, задаване на функционалност на бутон.</p> <p>Оценява дизайн на потребителски интерфейс и открива грешки.</p>		

Учебна седмица по ред	Тема на урочна единица	Урочна единица	Компетентности като очаквани резултати от обучението	Нови понятия	Контекст и дейности за всяка урочна единица	Методи и форми на оценяване по теми и/или раздели	Забележка
8	9. Проектиране на графичен потребителски интерфейс. Практика	П – 2	Отваря проект в интегрирана среда за визуално програмиране. Настройва основни свойства на форма, етикет, текстово поле и бутон. Проектира потребителски интерфейс на приложение.		Усъвършенстване на умения за основни елементи на потребителски интерфейс (контейнери и контроли); основни свойства на обекти – контроли, задаване на функционалност на бутон. Срянява възможности. Проектира самостоятелно потребителски интерфейс на приложение.		
9	10. Проектиране и оценка на потребителски интерфейс. Практика	П – 2	Оценява потребителски интерфейс на приложение.		Усъвършенстване на умения при създаване, редактиране и откриване на грешки при проектиране на потребителски интерфейс и тестване на приложение за грешки.		
10	11. Проектиране на графичен потребителски интерфейс. Контрол и оценка	К0 – 2	Контрол и оценка на знанията и уменията на учениците.		Затвърждаване на знания и умения. Оценяване на знания и умения. Рефлексия.		
ПРОГРАМИРАНЕ							
11	12. Основни типове данни	НЗУ – 2	Тип низ (string) – Познава правила за именуване на константи и променливи. – Декларира, описва и инициализира променливи и константи от тип низ. – Присвоява стойност на променлива от тип низ. – Въвежда и извежда данни от тип низ в/от текстово поле. – Извежда данни от тип низ в/от етикет. – Извършва конкатенация на низове. – Използва стандартни методи на интерфейсни компоненти за формиране на текст.	– символ – низ – множество на допустимите данни – множество на допустимите операции – име, тип стойност на променлива – име, тип и стойност на константа – присвояване на стойност – конкатенация	Въвеждане на понятията низ, множество на допустими данни, множество на допустими операции, име, тип стойност на променлива/константа, присвояване на стойност, конкатенация. Формиране на умения за именуване, деклариране, описване, инициализиране на променливи и константи от тип низ. Формиране на умения за присвояване на стойност от тип низ. Формиране на умения за въвеждане и извеждане на данни от тип низ. Формиране на умения за извършване на конкатенация на низове. Формиране на умения за използване на стандартни методи на интерфейсни компоненти за формиране на текст.		

Учебна седмица по ред	Тема на урочна единица	Урочна единица	Компетентности като очаквани резултати от обучението	Нови понятия	Контекст и дейности за всяка урочна единица	Методи и форми на оценяване по теми и/или раздели	Забележка
			<p>Целочислени типове данни</p> <ul style="list-style-type: none"> – Декларира, описва и инициализира променливи и константи от целочислен тип данни. – Използва въградени функции за преобразуване на низ в цяло число и обратното. – Въвежда и извежда данни от целочислен тип. – Използва различни целочислени типове данни. – Познава целочислените аритметични операции и техния приоритет. – Конструира аритметични изрази, съдържащи само целочислени данни, спазвайки синтаксиса и семантиката на конкретния език за програмиране. – Прилага и анализира резултатите от операцията – събиране, изваждане, умножение, деление, цяла част и остатък от целочислено деление. – Реализира модел за решаване на задачи, базиран на целочислени типове данни. 	<ul style="list-style-type: none"> – целочислен тип данни – конвенция за именуване на константи и променливи 	<p>Въвеждане на понятието целочислен тип данни и конвенция за именуване на константи и променливи.</p> <p>Формиране на умения за деклариране, описване и инициализиране на променливи и константи от целочислен тип данни.</p> <p>Формиране на умения за използване на въградени функции за преобразуване на низ в цяло число и обратното.</p> <p>Формиране на умения за въвеждане и извеждане на данни от целочислен тип.</p> <p>Актуализиране на знанията за приоритет на аритметичните операции.</p> <p>Формиране на умения за конструирание на аритметични изрази, съдържащи само целочислени данни.</p> <p>Формиране на умения за реализиране на модел за решаване на задачи с целочислен тип данни.</p>		
			<p>Реални типове данни</p> <ul style="list-style-type: none"> – Декларира, описва и инициализира променливи и константи от реален тип. – Използва въградени функции за преобразуване на низ в реално число и обратното. – Въвежда и извежда данни от реален тип данни. – Използва различни реални типове данни. – Познава приоритетите на аритметичните операции при реални типове данни. – Конструира аритметични изрази, съдържащи реални типове данни, спазвайки синтаксиса и семантиката на конкретния език за програмиране. – Прилага и анализира резултатите от операцията – събиране, изваждане, умножение, деление. – Реализира модел за решаване на задачи, базиран на реални типове данни. 	<ul style="list-style-type: none"> – реален тип данни 	<p>Въвеждане на понятието реален тип данни за именуване на константи и променливи.</p> <p>Формиране на умения за деклариране, описване и инициализиране на променливи и константи от реален тип данни.</p> <p>Формиране на умения за използване на въградени функции за преобразуване на низ в реално число и обратното.</p> <p>Формиране на умения за въвеждане и извеждане на данни от реален тип.</p> <p>Актуализиране на знанията за приоритет на аритметичните операции с реален тип данни.</p> <p>Формиране на умения за конструирание на аритметични изрази, съдържащи само реални данни.</p> <p>Формиране на умения за реализиране на модел за решаване на задачи с целочислен тип данни.</p>		

Учебна седмица по ред	Тема на урочна единица	Урочна единица	Компетентности като очаквани резултати от обучението	Нови понятия	Контекст и дейности за всяка урочна единица	Методи и форми на оценяване по теми и/или раздели	Забележка
			<p>Аритметични изрази и въградени математически функции. Приоритет на операциите.</p> <ul style="list-style-type: none"> – Оценява числената стойност на аритметичен израз, записан на език за програмиране. – Записва аритметичен израз със средствата на език за програмиране. – Прилага и използва въградени в езика за програмиране математически функции – абсолютна стойност, повдигане на степен, закръгляване, извличане на цялата част на реално число. – Използва приоритет на операциите в аритметични изрази, съдържащи въградени функции. – Създава аритметични изрази, съдържащи различни типове данни, като се съобразява със съвместимостта им. – Описва синтаксис и семантика на оператор за присвояване. 	<ul style="list-style-type: none"> – аритметичен израз в език за програмиране – въградени математически функции в език за програмиране – съвместимост на типове данни – форматирани изход 	<p>Въвеждане на понятието аритметичен израз в език за програмиране; въградени математически функции в език за програмиране, съвместимост на типове данни, форматирани изход.</p> <p>Формиране на умения за оценяване на числената стойност на аритметичен израз, записан на език за програмиране.</p> <p>Формиране на умения за запис на аритметичен израз.</p> <p>Формиране на умения за прилагане на въградени в езика за програмиране математически функции.</p> <p>Формиране на умения за създаване на аритметични изрази, съдържащи различни типове данни, като се съобразява със съвместимостта им.</p> <p>Въвеждане на понятията синтаксис и семантика на оператор за присвояване.</p> <p>Формиране на умения за форматирани на изхода на реално число.</p>		
12	13. Основни типове данни. Практика	П – 2	<ul style="list-style-type: none"> – Декларира, описва и инициализира променливи и константи от тип низ. – Присвоява стойност на променлива от тип низ. – Въвежда и извежда данни от тип низ в/от текстово поле. – Извежда данни от тип низ в/от етикет. – Извършва конкатенация на низове. – Използва стандартни методи на интерфейсни компоненти за форматиране на текст. – Реализира модел за решаване на задачи, базиран на реални типове данни. 	<p>Компоненти на графичен потребителски интерфейс, текстови полета, променливи, типове данни, моделиране на изчисления, откриване на грешки, тестване и създаване на тестове.</p>	<p>Усъвършенстване на уменията за използване на елементи на потребителски интерфейс и работа със стрингове.</p> <p>Затвърждаване на умения за проектиране.</p>		

Учебна седмица по ред	Тема на урочна единица	Урочна единица	Компетентности като очаквани резултати от обучението	Нови понятия	Контекст и дейности за всяка урочна единица	Методи и форми на оценяване по теми и/или раздели	Забележка
13	14. Основни типове данни. Обобщение	0 – 1	<ul style="list-style-type: none"> – Въвежда и извежда данни от тип низ в/от текстово поле. – Извежда данни от тип низ в/от етикет. – Извършва конкатенация на низове. – Използва стандартни методи на интерфейсни компоненти за форматиране на текст. – Използва стандартни методи на интерфейсни компоненти за форматиране на текст. – Реализира модел за решаване на задачи, базиран на целочислени и реални типове данни. – Форматира изхода на цяло, реално число. 		Усъвършенстване на умения за типове данни: тип низ (string), целочислени типове, реални типове, аритметични изрази и въградени математически функции. Приоритет на операциите.		
13	15. Основни типове данни. Контрол и оценка	К0 – 1	Контрол и оценка на знанията и уменията на учениците.		Контрол и оценка на знанията и уменията за типове данни.		
14	16. Създаване на компютърна програма за решаване на конкретна задача	НЗУ – 2	<ul style="list-style-type: none"> – Знае основните етапи при създаване на компютърна програма. – Анализира и проектира решението на конкретна задача. – Създава математическия модел за решаване на задачата. – Разработва алгоритъм за решаване на задачата. – Определя входно – изходни данни и техните типове. – Структурира и разработва графичен интерфейс, като използва обекти и декларира променливи. – Създава и описва програмния код; – Коментари; – Оформяне на програмния код. – Стартира, тества и валидира готовия проект. – Открива синтактични и логически грешки в програмата. – Отстранява синтактични и логически грешки при програмиране. – Спазва изисквания за оформяне на програмния код, включващи подравняване, коментари, именуване на програмните единици. 	<ul style="list-style-type: none"> – Коментари – оформяне на програмния код 	<p>Въвеждане на етапите за създаване на компютърна програма.</p> <p>Формиране на умения за анализиране и проектиране на решението на задача.</p> <p>Усъвършенстване на умения за създаване на математически модел на задача.</p> <p>Формиране на умения за разработка на алгоритъм за решаване на задача.</p> <p>Въвеждане на понятията входно-изходни данни и типовете им.</p> <p>Формиране на умения за разработване на графичен интерфейс с деклариране на променливи.</p> <p>Формиране на умения за стартиране, тестване и валидиране на готов проект.</p> <p>Формиране на умения за откриване на грешки при програмиране.</p> <p>Формиране на умения за спазване на изискванията за оформяне на програмния код.</p>		

Учебна седмица по ред	Тема на урочна единица	Урочна единица	Компетентности като очаквани резултати от обучението	Нови понятия	Контекст и дейности за всяка урочна единица	Методи и форми на оценяване по теми и/или раздели	Забележка
15	17. Създаване на компютърна програма за решаване на конкретна задача. Практика	П – 2	<ul style="list-style-type: none"> – Знае основните етапи при създаване на компютърна програма. – Анализира и проектира решението на конкретна задача. – Създава математическия модел за решаване на задачата. – Разработва алгоритъм за решаване на задачата. – Определя входно-изходни данни и техните типове. – Структурира и разработва графичен интерфейс, като използва обекти и декларира променливи. – Създава и описва програмния код: <ul style="list-style-type: none"> – коментари; – оформяне на програмния код. – Стартира, тества и валидира готовия проект. – Открива синтактични и логически грешки в програмата. – Отстранява синтактични и логически грешки при програмиране. – Спазва изисквания за оформяне на програмния код, включващи подравняване, коментари, именувание на програмните единици. 		Актуализира знания за основните етапи при създаване на компютърна програма, анализ на проблема и проектиране на решението. Усъвършенстване на умения за създаване на компютърна програма за решаване на конкретна задача.		
16	18. Програмни конструкции за реализация на разклонен алгоритъм	НЗУ – 2	<ul style="list-style-type: none"> Булев тип данни – Обосновава необходимостта от разклоняване на алгоритмичния процес. – Използва константите от булев тип данни. – Дава примери, в които се използва булев тип данни. – Декларира булева променлива. – Присвоява стойност на булева променлива. – Записва на език за програмиране булев израз, съвръщащ операция за сравнение. – Изписва синтактично правилно на езика за програмиране основните логически операции – логическо отрицание, дизюнкция, конюнкция. – Познава приоритета на логическите операции. – Пресмята без използване на компютър стойността на булев израз. – Съставя със средствата на език за програмиране сложен булев израз, отговарящ на дадена логическа ситуация. 	<ul style="list-style-type: none"> – булев тип данни – булеви константи – false, true – логически операции – приоритет на логическите операции – булев израз 	<p>Въвеждане на понятието булев тип данни, булеви константи, логически операции, приоритет, булев израз.</p> <p>Формиране на умения за използване на константи и променливи от булев тип.</p> <p>Формиране на умения за запис на булев израз.</p> <p>Формиране на умения за пресмятане на стойности на булев израз.</p> <p>Формиране на умения за съставяне на програма със сложен булев израз; отговарящ на логическа ситуация.</p>		

Учебна седмица по ред	Тема на урочна единица	Урочна единица	Компетентности като очаквани резултати от обучението	Нови понятия	Контекст и дейности за всяка урочна единица	Методи и форми на оценяване по теми и/или раздели	Забележка
			<p>Условен оператор Описва синтаксиса и семантиката на кратка и пълна форма на условен оператор. Описва разклонен алгоритъм с помощта на условен оператор. Използва условен оператор за проверка на коректността на входните данни за програма. Използва условен оператор за обработка на свойства на радиобутон и поле за отмятка.</p>	<ul style="list-style-type: none"> – условен оператор – съставен оператор 	<p>Ввеждани на понятието условен оператор и съставен оператор. Формиране на умения за използване на условен оператор за проверка на коректността на входните данни. Формиране на умения за използване на условен оператор за обработка на радиобутон и поле за отмятка.</p>		
			<p>Вложени условни оператори – Обяснява семантиката на вложени условни оператори в кратка и пълна форма. – Проиграва изпълнението на фрагмент на програма, съдържаща вложени условни оператори. – Записва синтактично и логически правилно вложени условни оператори. – Създава модел и алгоритъм за решаване на задача чрез използване на вложени условни конструкции. – Реализира модел за решаване на задача чрез използване на вложени условни оператори. – Заменя вложен условен оператор с единичен и обратно.</p>	<ul style="list-style-type: none"> – вложен условен оператор 	<p>Ввеждани на понятието вложен условен оператор. Формиране на умения за програмиране на фрагмент от програма, съдържаща вложени условни оператори. Формиране на умения за създаване на модел и алгоритъм за решаване на задача с използване на вложен условен оператор. Формиране на умения за реализиране на модел за решаване на задача с вложен условен оператор. Формиране на умения за замяна на вложен условен оператор с единичен и обратно.</p>		
17	19. Програмни конструкции за реализация на разклонен алгоритъм. Практика	П – 2	<ul style="list-style-type: none"> – Записва на език за програмиране булев израз, съдържащ операция за сравнение. – Пресмята без използване на компютър стойността на булев израз. – Съставя със средствата на език за програмиране сложен булев израз, отговарящ на дадена логическа ситуация. – Описва синтаксиса и семантиката на кратка и пълна форма на условен оператор. – Описва разклонен алгоритъм с помощта на условен оператор. 		<p>Усовършенстване на умения за реализиране на модел за решаване на задача с вложен условен оператор.</p>		

Учебна седмица по ред	Тема на урочна единица	Урочна единица	Компетентности като очаквани резултати от обучението	Нови понятия	Контекст и дейности за всяка урочна единица	Методи и форми на оценяване по теми и/или раздели	Забележка
18	20. Програмни конструкции за реализация на разклонен алгоритъм. Практика	П – 2	<ul style="list-style-type: none"> – Използва условен оператор за проверка на коректността на входните данни за програма. – Използва условен оператор за обработка на свойства на радиобутона и поле за отметка. – Записва синтактично и логически правилно вложени условни оператори. – Създава модел и алгоритъм за решаване на задача чрез използване на вложени условни конструкции. – Реализира модел за решаване на задача чрез използване на вложени условни оператори. – Заменя вложен условен оператор с единичен и обратно. 		Усъвършенстване на умения за логически изрази, условни конструкции, разклонени алгоритми		
19	21. Условни конструкции и разклонени алгоритми. Обобщение	0 – 1	<ul style="list-style-type: none"> – Дава примери, в които се използва булев тип данни. – Записва на език за програмиране булев израз, съдържащ операция за сравнение. – Изписва синтактично правилно на езика за програмиране основните логически операции – логическо отрицание, дизюнкция, конюнкция. – Пресмята без използване на компютър стойността на булев израз. – Съставя със средствата на език за програмиране сложен булев израз, оговарящ на дадена логическа ситуация. Описва синтаксиса и семантиката на кратка и пълна форма на условен оператор. Описва разклонен алгоритъм с помощта на условен оператор. – Проиграва изпълнението на фрагмент на програма, свързваща вложени условни оператори. – Записва синтактично и логически правилно вложени условни оператори. – Създава модел и алгоритъм за решаване на задача чрез използване на вложени условни конструкции. – Реализира модел за решаване на задача чрез използване на вложени условни оператори. – Заменя вложен условен оператор с единичен и обратно. 		Усъвършенстване на умения за условни конструкции и разклонени алгоритми.		

Учебна седмица по ред	Тема на урочна единица	Урочна единица	Компетентности като очаквани резултати от обучението	Нови понятия	Контекст и дейности за всяка урочна единица	Методи и форми на оценяване по теми и/или раздели	Забележка
19 – 20	22. Условни конструкции и разклонени алгоритми. Практика	П – 2	<ul style="list-style-type: none"> – Обосновава необходимостта от разклоняване на алгоритмичния процес. – Записва на език за програмиране булев израз, съдържащ операция за сравнение. – Използва условен оператор за проверка на коректността на входните данни за програма. – Използва условен оператор за обработка на свойства на радиобутон и поле за отметка. – Създава модел и алгоритъм за решаване на задача чрез използване на вложени условни конструкции. – Реализира модел за решаване на задача чрез използване на вложени условни оператори. 		Усъвършенстване на умения за условни конструкции и разклонени алгоритми.		
20	23. Условни конструкции и разклонени алгоритми. Контрол и оценка	К0 – 1	<ul style="list-style-type: none"> – Контрол и оценка на знанията и уменията на учениците. 		Контрол и оценка на знанията и уменията за условни конструкции и разклонени алгоритми.		
21	24. Програмни конструкции за реализация на циклични алгоритми	НЗУ – 2	<ul style="list-style-type: none"> – Посочва елементите на циклична конструкция – инициализация, тяло на цикъла и условие на цикъла. – Оценява необходимостта от използването на алгоритми с циклични конструкции с предусловие и предусловие. – Записва синтактично и логически правилно оператори за цикъл с предусловие, предусловие и управлявани от брояч. – Прилага алгоритми с циклични конструкции за проверка на входни данни. – Оценява необходимостта от използване на алгоритми с циклични конструкции с условие или управлявани от брояч. 	<ul style="list-style-type: none"> – структура на циклични алгоритмични конструкции – оператор за цикъл с предусловие – оператор за цикъл с предусловие – оператор за цикъл, управляван от брояч – списъчно поле 	<ul style="list-style-type: none"> – Въвеждане на понятията циклична алгоритмична конструкция и видове. – Формиране на умения за оценяване на необходимостта от използването на алгоритми с циклични конструкции с предусловие и предусловие. – Формиране на умения за записване на синтактично и логически правилно оператор за цикъл с предусловие, предусловие и с брояч. – Формиране на умения за прилагане на алгоритми с циклични конструкции за проверка на входни данни. 		

Учебна седмица по ред	Тема на урочна единица	Урочна единица	Компетентности като очаквани резултати от обучението	Нови понятия	Контекст и дейности за всяка урочна единица	Методи и форми на оценяване по теми и/или раздели	Забележка
21			<ul style="list-style-type: none"> – Преобразува програмен код, съдържащ циклична конструкция, управлявана от брояч, в циклична или управлявана от условие. – Открива синтактични и логически грешки в програмния код на алгоритми с циклична конструкция. – Прилага циклични алгоритми за управление на графичен потребителски интерфейс. – Прилага програмни конструкции за реализация на алгоритми за намиране на сума, минимален/максимален елемент, средно аритметично и др. в редици от числа, въвеждани от потребителския интерфейс/клавиатурата. 		<p>Формиране на умения за преобразуване на програмен код, съдържащ циклична конструкция, управлявана от брояч, в циклична конструкция или управлявана от условие.</p> <p>Формиране на умения за откриване на синтактични и логически грешки в програмен код на алгоритми циклична конструкция.</p> <p>Формиране на умения за прилагане на циклични алгоритми за управление на графичен потребителски интерфейс.</p>		
22	25. Програмни конструкции за реализация на циклични алгоритми. Практика	П – 2	<ul style="list-style-type: none"> – Прилага циклични алгоритми за управление на графичен потребителски интерфейс. – Използва циклични алгоритми за въвеждане и извеждане на данни от файл. 		<p>Усъвършенстване на умения за използване на програмни конструкции за реализация на алгоритми за намиране на сума минимален или максимален елемент, средно аритметично и др. в редици от числа.</p>		
23	26. Програмни конструкции за реализация на циклични алгоритми в компютърна графика. Практика	П – 2	<ul style="list-style-type: none"> – Прилага циклични алгоритми за изчертаване на графични примитиви. 	<ul style="list-style-type: none"> – графичен примитив – кутия за изображение – текстов файл 	<p>Въвеждане на понятията графичен примитив, кутия за изображение в текстов файл.</p> <p>Формиране на умения за прилагане на циклични алгоритми за изчертаване на графични примитиви.</p> <p>Формиране на умения за използване на циклични алгоритми за въвеждане и извеждане на данни от файл.</p>		
24	27. Циклични конструкции и циклични алгоритми. Обобщение	0 – 1	<ul style="list-style-type: none"> – Записва синтактично и логически правилно оператори за цикъл с предусловие, постусловие и управлявани от брояч. – Прилага алгоритми с циклични конструкции за проверка на входни данни. – Прилага програмни конструкции за реализация на алгоритми за намиране на сума, минимален/максимален елемент, средно аритметично и др. в редици от числа, въвеждани от потребителския интерфейс/клавиатурата. 		<p>Усъвършенстване на умения за циклични конструкции и алгоритми.</p>		

Учебна седмица по ред	Тема на урочна единица	Урочна единица	Компетентности като очаквани резултати от обучението	Нови понятия	Контекст и дейности за всяка урочна единица	Методи и форми на оценяване по теми и/или раздели	Забележка
24			– Прилага циклични алгоритми за управление на графичен потребителски интерфейс.				
24	28. Циклични конструкции и циклични алгоритми. Контрол и оценка	К0 – 1	Контрол и оценка на знанията и уменията на учениците.		Контрол и оценка на знанията уменията за за циклични конструкции и алгоритми.		
25	29. Тестване и верификация на програма	НЗУ – 2	<ul style="list-style-type: none"> – Обяснява и разграничава понятията тестване и верификация. – Дефинира тествови данни – Дефинира очаквани резултати от тестването при определени входни данни. – Използва инструмент за откриване и отстраняване на грешки (debugger). – Прилага процедури за тестване и верификация на вече създадени програми. 	<ul style="list-style-type: none"> – тестване – верификация – тествови данни 	<p>Въвеждане на понятията тестване, верификация, тествови данни.</p> <p>Формиране на умения за дефиниране на тествови данни и очаквани резултати от тестването при определени входни данни.</p> <p>Формиране на умения за използване на инструмент за откриване и отстраняване на грешки.</p> <p>Формиране на умения за прилагане на процедури за тестване и верификация на вече създадени програми.</p>		
26	30. Съставни типове данни. Егномерен масив	НЗУ – 2	<ul style="list-style-type: none"> – Разбира необходимостта от използване на масиви. – Идентифицира елементите на масив. – Разпознава индекс и стойност на елемент на масив. – Дефинира масив със средствата на език за програмиране. – Създава и инициализира масив със средствата на език за програмиране. – Осъществява достъп до елемент на масив. – Обхожда, въвежда и извежда стойности на елементите на масив. – Използва списъчно поле за извеждане на стойностите на елементите на масив. 	<ul style="list-style-type: none"> – логическо описание на масив – базов тип на масив – индекс и стойност на елемент от масив 	<p>Въвеждане на понятието масив, логическо описание на масив, базов тип на масив, индекс и стойност на елемент от масив.</p> <p>Затвърждаване на знанията за елементи на масив индекс и стойност на елемент на масив.</p> <p>Формиране на умения за дефиниране на масив със средствата на език за програмиране.</p> <p>Формиране на умения за създаване и инициализиране на масив със средствата на език за програмиране.</p> <p>Формиране на умения за осъществяване на достъп до елемент на масив.</p> <p>Формиране на умения за обхождане, въвеждане и извеждане на стойности на елементите на масив.</p> <p>Формиране на умения за използване на списъчно поле за извеждане на стойностите на елементите на масив.</p>		

Учебна седмица по ред	Тема на урочна единица	Урочна единица	Компетентности като очаквани резултати от обучението	Нови понятия	Контекст и дейности за всяка урочна единица	Методи и форми на оценяване по теми и/или раздели	Забележка
27	31. Съставни типове данни. Едномерен масив. Практика	П – 2	<p>Основни алгоритми за работа с едномерен масив</p> <ul style="list-style-type: none"> – Пресмята сбор и произведение на стойностите на елементите на едномерен масив. – Търси елемент от масива с максимална и минимална стойност. – Търси елементи от масива, отговарящи на дадено условие. <p>– Дефинира масив със средствата на език за програмиране.</p> <ul style="list-style-type: none"> – Създава и инициализира масив със средствата на език за програмиране. – Осъществява достъп до елемент на масив. – Обхожда, въвежда и извежда стойности на елементите на масив. – Използва списъчно поле за извеждане на стойностите на елементите на масив. 	– последователно търсене	<p>Формиране на умения за пресмятане на сбор и произведение на стойностите на елементите на едномерен масив.</p> <p>Формиране на умения за търсене на елемент от масив с максимална и минимална стойност.</p> <p>Формиране на умения за търсене на елементи от масив, отговарящи на дадено условие.</p> <p>Усъвършенстване на умения за работа с едномерни масиви.</p>		
28	32. Обработка на редици от елементи. Обобщение	0 – 1	<p>Разбира необходимостта от използване на масиви.</p> <ul style="list-style-type: none"> – Дефинира масив със средствата на език за програмиране. – Създава и инициализира масив със средствата на език за програмиране. – Осъществява достъп до елемент на масив. – Обхожда, въвежда и извежда стойности на елементите на масив. – Използва списъчно поле за извеждане на стойностите на елементите на масив. – Пресмята сбор и произведение на стойностите на елементите на едномерен масив. – Търси елемент от масива с максимална и минимална стойност. – Търси елементи от масива, отговарящи на дадено условие. 		<p>Усъвършенстване на умения за работа с редици от елементи.</p>		
28–29	33. Обработка на редици от елементи. Практика	П – 2			<p>Усъвършенстване на умения по обработка на редица от елементи.</p>		

Учебна седмица по ред	Тема на урочна единица	Урочна единица	Компетентности като очаквани резултати от обучението	Нови понятия	Контекст и дейности за всяка урочна единица	Методи и форми на оценяване по теми и/или раздели	Забележка
29–30	34. Общи задачи по програмиране. Практика	П – 2			Усъвършенстване на умения по типове данни, условни конструкции, циклични алгоритми и едномерни масиви.		
30	35. Обработка на редици от елементи. Контрол и оценка	К0 – 1	Контрол и оценка на знанията и уменията на учениците		Контрол и оценка на знанията и уменията за редици от елементи.		
СОФТУЕРЕН ПРОЕКТ							
31	36. Създаване на софтуерен проект	НЗУ – 2	<ul style="list-style-type: none"> – Описва етапите при реализиране на софтуерен проект. – Извършва проучване и анализ на решения за даден групов проект. – Създава модел за решаване на проблема, поставен в заданието на проекта. – Проектира графичен потребителски интерфейс. – Създава програмен код за реализация на модела. – Създава тестови примери с входни данни и очаквани резултати. – Изготвя документация за софтуерния проект. – Презентира и защитава готовия софтуерен проект. 		<p>Въвеждане на етапите при реализиране на софтуерен проект.</p> <p>Запознаване с начини за проучване и анализ на решения за даден групов проект.</p> <p>Формиране на умения за създаване на модел за решаване на проблема поставяне заданието на проекта.</p> <p>Формиране на умения за проектиране на графичен потребителски интерфейс.</p> <p>Актуализиране на знания и умения за създаване на програмен код за реализация на модела.</p> <p>Актуализиране на знания и умения за създаване на тестови примери с входни данни и очаквани резултати.</p> <p>Формиране на умения за изготвяне на документация за собствен проект.</p> <p>Актуализиране на знания за презентиране и защита на готов софтуерен проект.</p>		

Учебна седмица по ред	Тема на урочна единица	Урочна единица	Компетентности като очаквани резултати от обучението	Нови понятия	Контекст и дейности за всяка урочна единица	Методи и форми на оценяване по теми и/или раздели	Забележка
32	37. Създаване на софтуерен проект. Практика	П – 2	<ul style="list-style-type: none"> – Описва етапите при реализиране на софтуерен проект. – Извършва проучване и анализ на решения за даден групов проект. – Създава модел за решаване на проблема, поставен в заданието на проекта. – Проектира графичен потребителски интерфейс. – Създава програмен код за реализация на модела. – Създава тестови примери с входни данни и очаквани резултати. 		<p>Актуализиране на знания за етапите при реализиране на софтуерен проект.</p> <p>Усъвършенстване на умения за създаване на модел за решаване на проблема.</p> <p>Усъвършенстване на умения за създаване на програмен код за реализация на модела.</p> <p>Усъвършенстване на умения за създаване на тестови примери с входни данни и очаквани резултати.</p>		
33	38. Създаване на софтуерен проект. Практика	П – 2	<ul style="list-style-type: none"> – Извършва проучване и анализ на решения за даден групов проект. – Създава модел за решаване на проблема, поставен в заданието на проекта. – Проектира графичен потребителски интерфейс. – Създава програмен код за реализация на модела. – Създава тестови примери с входни данни и очаквани резултати. 		<p>Запознаване с начини за проучване и анализ на решения за даден групов проект.</p> <p>Формиране на умения за създаване на модел за решаване на проблема за поставяне на заданието на проекта.</p> <p>Формиране на умения за проектиране на графичен потребителски интерфейс.</p> <p>Актуализиране на знания и умения за създаване на програмен код за реализация на модела.</p> <p>Актуализиране на знания и умения за създаване на тестови примери с входни данни и очаквани резултати.</p>		
34	39. Създаване на софтуерен проект. Практика	П – 2	<ul style="list-style-type: none"> – Описва етапите при реализиране на софтуерен проект. – Извършва проучване и анализ на решения за даден групов проект. – Създава модел за решаване на проблема, поставен в заданието на проекта. – Създава програмен код за реализация на модела. – Създава тестови примери с входни данни и очаквани резултати. – Изготвя документация за софтуерния проект. 		<p>Усъвършенстване на умения за познаване и прилагане на етапите при реализиране на софтуерен проект.</p> <p>Актуализиране на умения за провеждане на проучвания на необходимост от решения за даден проблем.</p> <p>Усъвършенстване на умения за създаване на модел за решаване на проблема.</p> <p>Усъвършенстване на умения за създаване на програмен код за реализация на модела.</p> <p>Актуализиране на знания и умения за създаване на тестови примери с входни данни и очаквани резултати.</p>		

Учебна седмица по ред	Тема на урочна единица	Урочна единица	Компетентности като очаквани резултати от обучението	Нови понятия	Контекст и дейности за всяка урочна единица	Методи и форми на оценяване по теми и/или раздели	Забележка
34					Формиране на умения за изготвяне на документация за собствен проект.		
35	40. Създаване на софтуерен проект. Практика	П – 2	<ul style="list-style-type: none"> – Описва етапите при реализиране на софтуерен проект. – Извършва проучване и анализ на решения за даден групов проект. – Създава модел за решаване на проблема, поставен в заданието на проекта. – Създава програмен код за реализация на модела. – Създава тестови примери с входни данни и очаквани резултати. – Изготвя документация за софтуерния проект. – Презентира и защитава готовия софтуерен проект. 		<p>Усъвършенстване на умения за познаване и прилагане на етапите при реализиране на софтуерен проект.</p> <p>Актуализиране на умения за провеждане на проучвания на необходимост от решения за даден проблем.</p> <p>Усъвършенстване на умения за създаване на модел за решаване на проблема.</p> <p>Усъвършенстване на умения за създаване на програмен код за реализация на модела.</p> <p>Актуализиране на знания и умения за създаване на тестови примери с входни данни и очаквани резултати.</p> <p>Формиране на умения за изготвяне на документация за собствен проект.</p>		
36	41. Създаване на софтуерен проект. Контрол и оценка	К0 – 1	Контрол и оценка на знанията и уменията на учениците по темата „Създаване на софтуерен проект“.		Контрол и оценка на знанията и уменията за софтуерен проект.		
36	42. Проверка на знанията и уменията по информатика. Изходно равнище	К0 – 1	Контрол и оценка на знанията и уменията на учениците по информатика.		Контрол и оценка на знанията и уменията по информатика.		

Приложение № 16 към т. 16

**УЧЕБНА ПРОГРАМА ПО ИНФОРМАЦИОННИ ТЕХНОЛОГИИ
ЗА VIII КЛАС
(ОБЩООБРАЗОВАТЕЛНА ПОДГОТОВКА)**

**КРАТКО ПРЕДСТАВЯНЕ НА УЧЕБНАТА
ПРОГРАМА**

Информатиката е наука, която се занимава с методите за структуриране, събиране, обработка и разпространение на данни. Постиженията на тази научна област са пряко и динамично свързани със създаването и развитието на високотехнологични средства (компютри, операционни и комуникационни системи, потребителски софтуер, софтуер за разработка на приложения и др.), които са важен инструмент и/или инфраструктура, върху която функционират почти всички сфери на съвременното общество. В тази връзка учебната програма (а и обучението като цяло) по информатиката в VIII клас е съобразена със система от взаимосвързани фактори, по-важните от които са:

1. Мястото на предмета в учебния план:
 - брой на часовете;
 - връзките и значението му (в хоризонтален и вертикален план) с другите учебни предмети, особено с математика и информационни технологии (ИТ).
2. Изискванията за резултатите от обучението по учебния предмет информатика в Държавния образователен стандарт за общообразователна подготовка.

3. Съвременното състояние, понятийният апарат, логическата структура, методите и средствата на научната област.

4. Технологичното оборудване в системата на средното образование – компютри, системен и приложен софтуер, мрежова и комуникационна инфраструктура, мултимедийни и други съвременни дидактически средства.

5. Възрастовите (познавателни и психологични) възможности и интереси на учениците, които имат интереси в областта на софтуерните и хардуерните науки, природо-математическите и икономическите направления.

6. Процесът на обучението по информатика притежава потенциални възможности за личностно изграждане и развитие на ученика – формиране на абстрактно и логическо мислене, възпитание и формиране на адекватно отношение към заобикалящата действителност.

7. Учебният предмет е и възлов инструмент, с помощта на който могат да се мотивират, ориентират, привличат и развият младите хора, така че впоследствие (по-горните класове и университетите) да бъдат създадени специалисти, способни да осигурят кадрово информационното обслужване на всички обществени сфери.

ОЧАКВАНИ РЕЗУЛТАТИ ОТ ОБУЧЕНИЕТО В КРАЯ НА КЛАСА

Област на компетентност	Знания, умения и отношения <i>В резултат на обучението ученикът:</i>
Информатика	Описва предмета и ролята на информатиката за моделиране. Познава представянето на информация във вид на данни. Посочва примери на обекти и явления, при които е практически приложимо използването на средствата на обектно-ориентираното моделиране
Обектно-ориентирано програмиране	Обяснява основните етапи при създаване и изпълнение на компютърна програма. Прилага обектно-ориентиран подход при създаване на несложна компютърна програма. Описва основни начини за създаване, изпълнение и тестване на програмен проект в интегрирана среда за разработка с използване на визуални графични средства. Използва библиотеки от готови компоненти. Спазва добър стил на програмиране
Графичен потребителски интерфейс	Разбира и използва основни компоненти на среда за визуално програмиране при разработка на софтуер. Проектира графичен потребителски интерфейс с визуални средства Избира подходяща графична компонента в съответствие с необходимата функционалност на графичния интерфейс. Умее да настройва свойствата на графичните компоненти. Програмира подразбиращи се събития за основни компоненти от графичния интерфейс
Алгоритми и структури	Разбира същността на „тип данни“. Разграничава различни типове данни. Определя подходящ тип данни за определена задача. Разбира същността на алгоритмите и начини за описанието им Прилага основни управляващи конструкции. Структурира данни в едномерен масив. Прилага основни алгоритми за намиране на сума, минимален/максимален елемент и средно аритметично. Чете и записва данни в текстов файл
Софтуерни приложения	Използва визуално програмиране за решаване на несложни задачи. Създава програми за графично изобразяване на геометрични обекти със стандартни средства в езика. Създава програмни приложения с мултимедийни компоненти. Представя аргументирано разработено софтуерно приложение пред публика

УЧЕБНО СЪДЪРЖАНИЕ

Теми	Компетентности като очаквани резултати от обучението	Нови понятия
1. Основи на информатиката 1.1. Числата и техните представяния	<p>Разширява и обобщава знанията, свързани с числата и техните представяния: непозиционни бройни системи; същност на позиционните бройни системи; формат на числата в десетична, двоична и шестнадесетична бройна система.</p> <p>Превръща числа от десетична в двоична бройна система и обратно. Извършва събиране, изваждане и умножение на две числа в двоична бройна система.</p> <p>Дава примери за използване на двоична и шестнадесетична бройна система.</p>	<p>непозиционни бройни системи</p> <p>позиционни бройни системи</p> <p>експоненциален формат и неговото предназначение</p> <p>двоична бройна система</p> <p>шестнадесетична бройна система</p>
1.2. Информационни дейности и процеси	<p>Обяснява предмета на информатиката и ролята ѝ в съвременното общество.</p>	<p>основни информационни дейности</p>

Теми	Компетентности като очаквани резултати от обучението	Нови понятия
1.3. Алгоритми и езици за програмиране	<p>Изброява и описва основните информационни дейности събиране, съхраняване, преработка и разпространение и общата схема на информационните потоци.</p> <p>Описва понятието информационен процес и дава примери на информационни процеси, свързани с решаване на житейски задачи.</p> <p>Различава понятията информация и данни.</p> <p>Обяснява и илюстрира с примери връзката между информация и данни.</p> <p>Обяснява и илюстрира с примери същността на дискретното представяне на информацията за трансформирането ѝ в данни.</p>	<p>информация, данни, дискретно представяне на информацията</p>
1.3. Алгоритми и езици за програмиране	<p>Дефинира понятието алгоритъм и описва основните му характеристики (резултатност, крайност, детерминираност, масовост).</p> <p>Описва и проследява несложни, линейни и разклонени алгоритми с различни средства.</p> <p>Описва същността, структурата и разновидностите на цикличните алгоритмични конструкции.</p> <p>Обяснява същността и функционалното предназначение на език за програмиране.</p> <p>Обяснява същността и предназначението на транслятор (интерпретатор, компилатор).</p> <p>Представя исторически факти, свързани със създаването и развитието на съвременните езици и среди за програмиране.</p>	<p>алгоритъм</p> <p>език за програмиране</p> <p>транслятор</p>

2. Среда за визуално програмиране		
2.1. Интегрирана среда за визуално програмиране	<p>Отваря проект в интегрирана среда за визуално програмиране. Редактира дизайн на графичния потребителски интерфейс на приложението в интегрирана среда за програмиране. Запазва проект на приложението чрез средствата на интегрирана среда за програмиране.</p> <p>Стартира приложението с графичен потребителски интерфейс чрез средствата на интегрирана среда за програмиране.</p> <p>Разпознава основни компоненти на интегрирана среда за програмиране – графичен и текстов редактор, панел с контроли, панел за свойства на обект, панел за съобщения, панел за преглед на структурата на приложението.</p>	<p>интегрирана среда за програмиране свързващ редактор (linker) програма за откриване и отстраняване на грешки (debugger) редактор за проектиране на дизайн на графичен потребителски интерфейс</p>

Теми	Компетентности като очаквани резултати от обучението	Нови понятия
2.2. Основни етапи на създаване и изпълнение на компютърна програма	<p>Анализира задача с несложен математически модел.</p> <p>Създава математически модел за решаване на несложна задача.</p> <p>Съпоставя математически модел с програмно решение на даден проблем.</p> <p>Открива основните компоненти на математически модел в демонстрирано програмно решение на даден проблем.</p> <p>Стартира чрез средствата на интегрирана среда предварително подготвена компютърна програма с графичен потребителски интерфейс.</p> <p>Тества предварително подготвен несложен проект.</p> <p>Разпознава видовете грешки при програмиране.</p> <p>Разчита и прави предположение за естеството на синтактична грешка в даден проект.</p> <p>Открива и прави предположение за причината за логическа грешка в дадено приложение.</p> <p>Открива и прави предположение за причината за грешка по време на изпълнение на приложението.</p>	<p>синтактични грешки в компютърна програма логически грешки в компютърна програма грешки по време на изпълнение на програмата</p>
2.3. Проектиране на графичен потребителски интерфейс	<p>Знае предназначението на основни контейнери и контроли – форма, етикет, текстово поле, бутон, диалогова кутия.</p> <p>Разпознава основни свойства на графични обекти-контроли – име, състояние, етикет, фон, настройка на шрифт и др.</p> <p>Проектира несложна форма, съдържаща етикет, текстово поле, бутон.</p> <p>Настройва основни свойства на форма, етикет, текстово поле и бутон.</p> <p>Именува обекти-контроли съгласно общоприетата конвенция.</p> <p>Задава функционалност на бутон, свързана с извеждането на статично съобщение в диалогова кутия.</p>	<p>интерфейсен компонент (контрола) контейнер на контроли свойство на обект метод на обект</p>

3. Програмиране		
Теми	Компетентности като очаквани резултати от обучението	Нови понятия
3.1. Основни типове данни	<p>Познава правила за именуване на константи и променливи. Декларира, описва и инициализира променливи и константи от тип низ. Присвоява стойност на променлива от тип низ. Въвежда и извежда данни от тип низ в/от текстово поле. Извежда данни от тип низ в/от етикет. Извършва конкатенация на низове. Използва стандартни методи на интерфейсни компоненти за форматиране на текст.</p>	<p>символ низ множество на допустимите данни множество на допустимите операции име, тип стойност на променлива име, тип и стойност на константа присвояване на стойност конкатенация</p>
3.1.1. Тип низ		
3.1.2. Целочислени типове данни	<p>Декларира, описва и инициализира променливи и константи от целочислен тип данни. Използва вградени функции за преобразуване на низ в цяло число и обратното. Въвежда и извежда данни от целочислен тип. Използва различни целочислени типове данни. Познава целочислените аритметични операции и техния приоритет. Конструира аритметични изрази, съдържащи само целочислени данни, спазвайки синтаксиса и семантиката на конкретния език за програмиране. Прилага и анализира резултатите от операциите – събиране, извеждане, умножение, деление, цяла част и остатък от целочислено деление. Реализира модел за решаване на задачи, базиран на целочислени типове данни.</p>	<p>целочислен тип данни конвенция за именуване на константи и променливи</p>
3.1.3. Реални типове данни	<p>Декларира, описва и инициализира променливи и константи от реален тип. Използва вградени функции за преобразуване на низ в реално число и обратното. Въвежда и извежда данни от реален тип данни.</p>	<p>реален тип данни</p>
Теми	Компетентности като очаквани резултати от обучението	Нови понятия
	<p>Използва различни реални типове данни. Познава приоритетите на аритметичните операции при реални типове данни. Конструира аритметични изрази, съдържащи реални типове данни, спазвайки синтаксиса и семантиката на конкретния език за програмиране. Прилага и анализира резултатите от операциите – събиране, извеждане, умножение, деление. Реализира модел за решаване на задачи, базиран на реални типове данни.</p>	

<p>3.1.4. Аритметични изрази и вградени математически функции. Приоритет на операциите</p>	<p>Оценява числената стойност на аритметичен израз, записан на език за програмиране. Записва аритметичен израз със средствата на език за програмиране. Прилага и използва вградени в езика за програмиране математически функции – абсолютна стойност, побдигане на степен, закръгляване, извличане на цялата част на реално число. Използва приоритет на операциите в аритметични изрази, съдържащи вградени функции. Създава аритметични изрази, съдържащ различни типове данни, като се съобразява със съвместимостта им. Описва синтаксис и семантика на оператор за присвояване. Форматира изхода на реално число.</p>	<p>аритметичен израз в език за програмиране вградени математически функции в език за програмиране съвместимост на типове данни форматиран изход</p>
<p>3.2 Създаване на компютърна програма за решаване на конкретна задача</p>	<p>Знае основните етапи при създаване на компютърна програма. Анализира и проектира решението на конкретна задача. Създава математическия модел за решаване на задачата. Разработва алгоритъм за решаване на задачата. Определя входно-изходни данни и техните типове. Структурира и разработва графичен интерфейс, като използва обекти и декларира променливи. Създава и описва програмния код. Стартира, тества и валидира готовия проект. Открива синтактични и логически грешки в програмата. Отстранява синтактични и логически грешки при програмиране. Спазва изисквания за оформяне на програмния код, включващи подравняване, коментари, именуване на програмните единици.</p>	<p>коментари оформяне на програмния код</p>

<p>Теми</p>		
<p>Компетентности като очаквани резултати от обучението</p>		
<p>3.3. Програмни конструкции за реализация на разклонен алгоритъм</p>	<p>Обосновава необходимостта от разклоняване на алгоритмичния процес. Използва константите от булев тип данни. Дава примери, в които се използва булев тип данни. Декларира булева променлива. Присвоява стойност на булева променлива. Записва на език за програмиране булев израз, съдържащ операция за сравнение. Изписва синтактично правилно на езика за програмиране основните логически операции – логическо отрицание, дизюнкция, конюнкция. Познава приоритета на логическите операции. Пресмята без използване на компютър стойността на булев израз. Съставя със средствата на език за програмиране сложен булев израз, отговарящ на дадена логическа ситуация.</p>	<p>Нови понятия булев тип данни булеви константи – false, true логически операции приоритет на логическите операции булев израз</p>
<p>3.3.1. Булев тип данни</p>		

	<p>Описва синтаксиса и семантиката на кратка и пълна форма на условен оператор. Описва разклонен алгоритъм с помощта на условен оператор.</p> <p>Използва условен оператор за проверка на коректността на входните данни за програма.</p> <p>Използва условен оператор за обработка на свойства на радио-бутон и поле за отметка.</p>	условен оператор съставен оператор
<p>3.3.2. Условен оператор</p>		вложен условен оператор
<p>3.3.3. Вложени условни оператори</p>		
<p>Теми</p>	<p>Компетентности като очаквани резултати от обучението</p>	Нови понятия
<p>3.4. Програмни конструкции за реализация на циклични алгоритми</p>	<p>вложени условни оператори.</p> <p>Записва синтактично и логически правилно вложени условни оператори.</p> <p>Създава модел и алгоритъм за решаване на задача чрез използване на вложени условни конструкции.</p> <p>Реализира модел за решаване на задача чрез използване на вложени условни оператори.</p> <p>Заменя вложен условен оператор с единичен и обратно.</p>	
<p>3.4.1. Циклични алгоритми</p>	<p>Посочва елементите на циклична конструкция – инициализация, тяло на цикъла и условие на цикъла.</p> <p>Оценява необходимостта от използването на алгоритми с циклични конструкции с предусловие и постусловие.</p> <p>Записва синтактично и логически правилно оператори за цикъл с предусловие, постусловие и управлявани от брояч.</p> <p>Прилага алгоритми с циклични конструкции за проверка на входни данни.</p> <p>Оценява необходимостта от използване на алгоритми с циклични конструкции с условие или управлявани от брояч.</p> <p>Преобразува програмен код, съдържащ циклична конструкция, управлявана от брояч в циклична конструкция или управлявана от условие.</p> <p>Открива синтактични и логически грешки в програмния код на алгоритми с циклична конструкция.</p> <p>Прилага циклични алгоритми за управление на графичен потребителски интерфейс.</p>	<p>структура на циклични алгоритмични конструкции</p> <p>оператор за цикъл с предусловие</p> <p>оператор за цикъл с постусловие</p> <p>оператор за цикъл, управляван от брояч</p> <p>списъчно поле</p>
<p>3.4.2. Приложение на условни и циклични конструкции</p>	<p>Прилага циклични алгоритми за изчертаване на графични примитиви.</p> <p>Използва циклични алгоритми за въвеждане и извеждане на данни от файл.</p> <p>Прилага програмни конструкции за реализация на алгоритми за намиране на сума, минимумен/максимален елемент, средно</p>	<p>графичен примитив</p> <p>кутия за изобразяване</p> <p>текстов файл</p>

Темѝ	Компетентности како очаквани резултати од обучението	Нови понятия
	аритметично и гр. в редици од числа, ввеждани од потребителскиот интерфејс/клавиатурата.	
3.5. Тестване и верификација на програмата	Објаснува и разграничува понятијата тестване и верификација. Дефинира тесстови данни Дефинира очаквани резултати од тестването при определени входни данни. Използва инструмент за откривање и отстранявање на грешки (debugger). Прилага процедури за тестване и верификација на вече создадени програми.	тестване верификација тесстови данни
3.6. Съставни типове данни. Едномерен масив		
3.6.1. Едномерен масив	Разбира необходимоста од използване на масиви. Идентифицира елементите на масив. Разпознава индекс и стойност на елемент на масив. Дефинира масив със средствата на език за програмиране. Създава и инициализира масив със средствата на език за програмиране. Осъществява достъп до елемент на масив. Обхожда, ввежда и извежда стойностите на елементите на масив. Използва списъчно поле за извеждане на стойностите на масив.	логическо описание на масив базов тип на масив индекс и стойност на елемент от масив
3.6.2. Основни алгоритми за работа с едномерен масив	Пресмята сбор и произведение на стойностите на елементите на едномерен масив. Търси елемент от масива с максимална и минимална стойност. Търси елементи от масива, отговарящи на дадено условие.	последователно търсене
4. Създаване проект	Описва етапите при реализиране на софтуерен проект. Извършва проучване и анализ на решения за даден групов проект. Създава модел за решаване на проблема, поставен в заданието на проекта. Проектира графичен потребителски интерфејс. Създава програмен код за реализация на модела. Създава тесстови примери с входни данни и очаквани резултати. Изготвя документация за софтуерния проект. Презентира и защитава готовия софтуерен проект.	

ПРЕПОРЪЧИТЕЛНО ПРОЦЕНТНО РАЗПРЕДЕЛЕНИЕ НА ЗАДЪЛЖИТЕЛНИТЕ УЧЕБНИ ЧАСОВЕ ЗА ГОДИНАТА

Изисквания за провеждане на обучението:

1. Програмата предвижда обучението да се извършва на базата на език за визуално програмиране – Visual Basic, C# или Java по избор на преподавателя. Разработена е така, че усвояването на необходимите понятия и умения да става чрез активно участие на ученика в учебния процес.

2. Препоръчва се обучението по първите три теми да се провежда предимно под формата на комбинирани уроци, с продължителност 2

последователни учебни часа. Уроците от този тип е подходящо да са със следната най-обща структура:

2.1. Упражнения за затвърждаване на учебното съдържание, съчетани с проверка и оценка на знанията и уменията и ориентирани (при възможност) към мотивация на новото учебно съдържание.

2.2. Мотивиране и въвеждане на новото учебно съдържание (теоретична и манипулативна компонента). Препоръчва се новото учебно съдържание да се дозира и въвежда от учителя като последователност от смислово обособени и логически подредени части (неголеми модули). След всеки такъв модул се отделя учебно време за самостоятелна практическа работа на учениците, подпомагана от учителя.

2.3. Самостоятелна работа на учениците, свързана с усвояването на теоретичната и манипулативната компонента на учебното съдържание.

2.4. Анализ на хода и резултатите от урока и задаване на домашна работа.

3. Желателно е учебното време на комбинирани уроци по Тема 2 и Тема 3 да съчетава въвеждането на нови знания с упражнения на компютър в приблизително отношение 30% : 70%. Чисто теоретичните занятия в лекционен стил трябва да бъдат сведени до минимум. Желателно е като цяло обемът на практическата работа да има превес над теоретичните части.

4. Обучението по Тема 4 предвижда учениците да разработват софтуерен проект, съдържащ проверка за коректност на входни данни, разклонени и циклични алгоритми, графичен интерфейс, формиране на изход. С цел развиване на компетентност за работа в екип учениците работят в групи от по двама души, като се прилага софтуерната технология „Работа по двойки“ (pair programming).

5. За осъществяване на ефективно обучение е необходимо да се осигури свободен достъп на учениците до компютърните зали в извънучебно време. Така всички ученици ще разполагат с компютърна техника независимо от социалния им статус и ще се създадат реални условия за работа на учениците в екип при създаване на софтуерни продукти.

Препоръчително разпределение на часовете:

За нови знания и умения	30%
За обобщение, контрол и оценка	20%
Практически дейности	50%

СПЕЦИФИЧНИ МЕТОДИ И ФОРМИ ЗА ОЦЕНЯВАНЕ НА ПОСТИЖЕНИЯТА НА УЧЕНИЦИТЕ

Форми на оценяване:

• Устна

Устната форма на оценка е удачна при групови обсъждания, дискусии, генериране на идеи. Оценяват се идеите и аргументите на ученика. Не се препоръчва устно изпитване, отнасящо се до запаметяване на теория.

• Писмена

Писмената форма е подходяща при проверка и оценка на теоретични знания и разбиране на синтаксиса и семантиката на даден оператор. Би могла да включва въпроси с избираем отговор, задачи за изчисление на израз, задачи за проследяване на работата на алгоритъм или програмен фрагмент, задачи за откриване на логически грешки в алгоритъм или програмен фрагмент. Писмената форма на изпитване може да се провежда самостоятелно или в съчетание с практическа форма.

• Практическа

Тъй като предметът е с предимно практическа насоченост, препоръчва се това да е преобладаващата форма в оценяването на постиженията на учениците. Практическо изпитване се прави на базата на портфолио, съдържащо резултатите от практическа работа в клас, защитена домашна работа и разработка на проект в качеството му на изходно ниво от обучението. Желателно е оценката да отразява всички етапи от практическата разработка (анализ, модел, програмиране, защита и др. според заданието), като за тази цел учителите трябва да подготви съответните критериални матрици.

Съотношение при формиране на срочна и годишна оценка:	
Текущи оценки от работа в клас, участие в групови обсъждания и дискусии, писмено изпитване	20%
Текущи оценки от домашни работи	25%
Текущи оценки от практически задания в клас	25%
Оценка на изходно ниво (проект)	30%

**ДЕЙНОСТИ ЗА ПРИДОБИВАНЕ НА КЛЮЧОВИТЕ КОМПЕТЕНТНОСТИ, КАКТО И
МЕЖДУПРЕДМЕТНИ ВРЪЗКИ**

Дейност 1.	Проучване и представяне на допълнителна информация, свързана с изучаваните теми. Учениците развиват умения за учене, социални и граждански компетентности, създават се междупредметни връзки с български език и литература.
Дейност 2.	Структуриране на статистическа информация във вид на данни в контекста на конкретна информационна задача. На учениците се дава възможност да демонстрират социални и граждански компетентности, както и компетентности в областта на българския език.
Дейност 3.	Идентифициране и дефиниране на стандартите в информатиката и използване на конвенция за именуване. Учениците демонстрират дигитални компетентности и умения за общуване на чужди езици.
Дейност 4.	Редактиране, компилиране и изпълнение на готова програма, създадена за различни сфери на обществения живот. Учениците развиват дигитални, математически компетентности и основни компетентности в областта на природните науки и на технологиите, социални и граждански компетентности. Осъществяват се междупредметни връзки с природни науки, информационни технологии и математика.
Дейност 5.	Редактиране на дизайна на приложение с графичен потребителски интерфейс във визуална среда за програмиране. Реализират се междупредметни връзки на информационните технологии и изобразителното изкуство - цвят, цветови схеми, елементи на интерфейса и гр.
Дейност 6.	Създаване на математически модели на ситуации, изучавани в часовете по математика и природни науки, като се използват средствата на програмирането. Учениците демонстрират математически и дигитални компетентности в областта на природните науки и на технологиите.
Дейност 7.	Настройки на основните свойства на графичните компоненти на приложение. Създаване на програма за изчисляване на печалбата на дадено предприятие, общия приход (разход) на домакинство. Демонстрира се компетентността инициативност и предприемчивост.
Дейност 8.	Описване на алгоритми за решаване на линейни уравнения, линейни неравенства и сравняване на отсечки и ъгли. Учениците развиват дигитални, математически компетентности и основни компетентности в областта на природните науки.
Дейност 9.	Въвеждане и извеждане на данни в приложения, свързани в природо-математическите дисциплини, бита и бизнеса. Класифициране на типовете данни. Учениците развиват дигитални, математически компетентности и основни компетентности в областта на природните науки и на технологиите, социални и граждански компетентности. Осъществяват се междупредметни връзки с природните науки, информационните технологии и математиката.
Дейност 10.	Въвеждане и извеждане на данни от тип низ в текстово поле. Прилагат граматически правила в българския език за конструиране на изречение, работят с измерителни единици. Осъществяват се междупредметни връзки с български език и литература, математика, природни науки.
Дейност 11.	Създаване на аритметични изрази, съдържащи различни типове данни, отчитайки съвместимостта им. Осъществяване междупредметни връзки с математиката (пресмятане на числена стойност на израз, разлагане на множители), физиката (определяне на пътя и скоростта при равноускорително движение и свободно падане).
Дейност 12.	Реализиране на несложни алгоритми със средствата на език за програмиране. Учениците развиват дигитални, математически

СЪДЪРЖАНИЕ

Обръщение към учителите.....	3
Методически бележки	4
I. ОСНОВИ НА ИНФОРМАТИКАТА	
1. Основи на информатика	9
2. Основи на информатиката. Практика.....	11
3. Основи на информатиката. Обобщение.....	14
4. Основи на информатиката. Контрол и оценка	16
II. СРЕДА ЗА ВИЗУАЛНО ПРОГРАМИРАНЕ	
5. Среда за визуално програмиране.....	18
6. Интегрирана среда за визуално програмиране. Практика	18
7. Основни етапи на създаване и изпълнение на компютърна програма. Практика.....	19
8. Проектиране на графичен потребителски интерфейс	20
9. Проектиране на графичен потребителски интерфейс. Практика	20
10. Проектиране и оценка на потребителски интерфейси. Практика.....	21
11. Проектиране на графичен потребителски интерфейс. Контрол и оценка	21
III. ПРОГРАМИРАНЕ	
12. Основни типове данни.....	23
13. Основни типове данни. Практика	24
14. Основни типове данни. Обобщение	25
15. Основни типове данни. Контрол и оценка	26
16. Създаване на компютърна програма за решаване на конкретна задача	27
17. Създаване на компютърна програма за решаване на конкретна задача. Практика	27
18. Програшни конструкции за реализация на разклонен алгоритъм	28
19. Програшни конструкции за реализация на разклонен алгоритъм. Практика	29
20. Програшни конструкции за реализация на разклонен алгоритъм. Практика.....	32
21. Условни конструкции и разклонени алгоритми. Обобщение	33
22. Условни конструкции и разклонени алгоритми. Практика.....	34
23. Условни конструкции и разклонени алгоритми. Контрол и оценка.....	36
24. Програшни конструкции за реализация на циклични алгоритми	38
25. Програшни конструкции за реализация на циклични алгоритми. Практика.....	39

26. Програмни конструкции за реализация на циклични алгоритми в компютърна графика. Практика	41
27. Циклични конструкции и циклични алгоритми. Обобщение	41
28. Циклични конструкции и циклични алгоритми. Контрол и оценка.....	42
29. Тестване и верификация на програма.....	43
30. Съставни типове данни. Едномерен масив.....	44
31. Съставни типове данни. Едномерен масив. Практика	45
32. Обработка на редици от елементи. Обобщение.....	45
33. Обработка на редици от елементи. Практика.....	46
34. Общи задачи по програмиране. Практика	46
35. Обработка на редици от елементи. Контрол и оценка	47
IV. СОФТУЕРЕН ПРОЕКТ	
36. Създаване на софтуерен проект.....	49
37. Създаване на софтуерен проект. Практика	49
38. Създаване на софтуерен проект. Практика	50
39. Създаване на софтуерен проект. Практика.....	50
40. Създаване на софтуерен проект. Практика	51
41. Създаване на софтуерен проект. Контрол и оценка.....	51
42. Проверка на знанията и уменията по информатика. Изходно равнище.....	52
V. ДЕЙНОСТИ ЗА ПРИДОБИВАНЕ НА КЛЮЧОВИ КОМПЕТЕНТНОСТИ. МЕЖДУПРЕДМЕТНИ ВРЪЗКИ	54
VI. ИДЕИ ЗА СОФТУЕРНИ ПРОЕКТИ.....	58
ПРИМЕРНО ГОДИШНО РАЗПРЕДЕЛЕНИЕ НА ТЕМИТЕ ПО ИНФОРМАТИКА ЗА 8. КЛАС	59
УЧЕБНА ПРОГРАМА ПО ИНФОРМАЦИОННИ ТЕХНОЛОГИИ ЗА VIII КЛАС	
(ОБЩООБРАЗОВАТЕЛНА ПОДГОТОВКА)	77

**Книга за учителя
по информатика
за 8. клас**

Автори

Галина-Момчева
Тодорка Глушкова
Росица Кузманова-Маринова

Редактор

Снежина Машева

Графичен дизайн

Николай Пекарев

Коректор

Мила Томанова

Българска. Първо издание, 2017
Формат 60x90/8. Печатни коли 11
ISBN 978-954-18-1057-6
ISBN 978-619-215-100-3
№ 17-08250040-17

Издателства

БУЛВЕСТ 2000

1505, София, ул. „Васил Друмев“ 36
Тел.: (02) 806 13 00, факс: (02) 806 13 13

ИК Анобис

1124, София, ул. „Младен Павлов“ 1, ет. 3
Тел.: (02) 944 35 03, факс: (02) 944 16 43

Печат

„БУЛВЕСТ ПРИНТ“ АД



БЕЗПЛАТЕН ЕКЗЕМПЛЯР