

## ГОДИШНО ТЕМАТИЧНО РАЗПРЕДЕЛЕНИЕ

по учебния предмет *информатика* за **8. клас**

**ПЪРВИ УЧЕБЕН СРОК – 18 седмици x 2 часа в блок седмично = 36 часа**

№ по ред	Учебна седмица по ред	Тема на урочната единица	Очаквани резултати от обучението	Методи за работа	Бележки/Коментари
(1)	(2)	(3)	(4)	(5)	(6)
1	1	Начален инструктаж и <b>установяване на входното ниво</b>	Ученикът знае и спазва правилата за безопасна работа в компютърен кабинет.	Ученикът: – се запознава с правилата за работа в компютърните кабинети и интернет и удостоверява това с подписа си; – попълва тест за входно ниво или участва в дискусия за определяне на познанията и компетентностите му.  Тест за проверка на началните знания/участие в дискусия.	Учителят преценява дали учениците да попълнят входно ниво, или да участват в дискусия.
<b>ТЕМА 1. ОСНОВИ НА ИНФОРМАТИКАТА</b>					
<b>1.1. Числата и техните представяния</b>					
2	1	1.1.1. Бройни системи	Ученикът: – 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; – дава примери за използване на двоична и шестнадесетична бройна система.	Учениците: – отговарят на въпроси; – участват в дискусия; решават задачи, свързани с представянето на числата. Устно: обсъждане на задачи и аргументация; писмено: решаване на задачи. Практическа работа.	

(1)	(2)	(3)	(4)	(5)	(6)
3	2	1.1.2. Превръщане на числа в различни бройни системи	Ученикът превръща числа от десетична в двоична и шестнадесетична бройна система и обратно.	Ученикът: – превръща цели числа от десетична в двоична и шестнадесетична бройна система; – превръща цели числа от двоична или шестнадесетична в десетична бройна система.  Устно: обсъждане на задачи и аргументация; писмено: решаване на задачи. Практическа работа.	
4	2	1.1.3. Аритметични операции в двоична бройна система	Ученикът извършва събиране, изваждане и умножение на две числа в двоична бройна система.	Ученикът отговаря на въпроси.  Писмено: решаване на задачи. Практическа работа.	
5	3	1.2. Информационни дейности и процеси	Ученикът: – обяснява предмета на информатиката и ролята ѝ в съвременното общество; – 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 решаването 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.	Ученикът: – отговаря на въпроси; – работи в екип за изпълнение на практически задачи.  Устно: обсъждане и предлагане на примери за информация, данни, информационен процес. Практическа работа.	Учениците работят по двойки.
<b>1.3. Алгоритми и езици за програмиране</b>					
6	3	1.3.1. Алгоритъм	Ученикът: – дефинира понятието „алгоритъм“ и описва основните му характеристики (резултатност, крайност, детерминираност, масовост);	Ученикът: – дава примери за алгоритми от живота; – описва алгоритми чрез текст и блок схема.  Практическа работа.	

(1)	(2)	(3)	(4)	(5)	(6)
			– описва и проследява несложни, линейни и разклонени алгоритми с различни средства.		
7	4	1.3.2. Видове алгоритми	Ученикът описва същността, структурата и разновидностите на цикличните алгоритмични конструкции.	Ученикът: – описва различни видове алгоритми; работи в екип от двама.  Практическа работа. по двойки	
8	4	1.3.3. Езици за програмиране	Ученикът: – обяснява същността и функционалното предназначение на език за програмиране; – обяснява същността и предназначението на транслатор (интерпретатор, компилатор); – представя исторически факти, свързани със създаването и развитието на съвременните езици и среди за програмиране.	Ученикът: – участва в беседа; – проучва информация в интернет; сравнява езици за програмиране.  Участие в обсъждане, практическа изследователска работа	
<b>ТЕМА 2. СРЕДА ЗА ВИЗУАЛНО ПРОГРАМИРАНЕ</b>					
9	5	<b>Тема 1. Тест</b>  2.1. Интегрирана среда за визуално програмиране	Ученикът проверява в каква степен е овладял изучените понятия в темата.  Ученикът: – 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.	Ученикът решава тест за самопроверка. Ученикът: – се запознава на практика с ИСП Eclipse; – разглежда инструмент за визуално програмиране Windows Builder в Eclipse; – зарежда предварително създаден проект в ИСП; – редактира ГПИ на приложение в ИСП; запазва проект.  Практическа работа.	В първата половина на часа се прави тест за самопроверка върху тема 1.
10	5	2.1. Интегрирана среда за визуално програмиране	Ученикът: – стартира приложение с графичен потребителски интерфейс чрез средствата на интегрирана среда за програмиране; – разпознава основни компоненти на интегрирана среда за програмиране –	Ученикът работи с предварително предоставен проект в ИСП.  Практическа работа.	

(1)	(2)	(3)	(4)	(5)	(6)
			графичен и текстов редактор, панел с контроли, панел за свойства на обект, панел за съобщения, панел за преглед на структурата на приложението.		
<b>2.2. Основни етапи на създаване и изпълнение на компютърна програма</b>					
11	6	2.2.1. Създаване и тестване на компютърна програма	Ученикът: – анализира задача с несложен математически модел; – създава математически модел за решаване на несложна задача; – съпоставя математически модел с програмно решение на даден проблем; – открива основните компоненти на математически модел в демонстрирано програмно решение на даден проблем; – стартира чрез средствата на интегрирана среда предварително подготвена компютърна програма с графичен потребителски интерфейс; – тества предварително подготвен несложен проект.	Ученикът: – участва в беседа, свързана с етапите от разработка на проект; – проигрива на практика основните етапи с помощта на предварително подготвен проект; описва етапите от разработка на проект, решаващ конкретна практическа задача от ежедневието.  Практическа работа.	
12	6	2.2.2. Видове грешки при програмиране	Ученикът: – 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.	Ученикът: – разглежда и анализира в ИСП предварително подготвени проекти, съдържащи различни типове грешки; – разчита съобщения за синтактични грешки; – прави предположения за причината за грешка; – открива логически грешки в предварително предоставени приложения.  Участие в беседа. Дискусия. Практическа работа.	
<b>2.3. Проектиране на графичен потребителски интерфейс</b>					
13	7	2.3.1. Компоненти на графичен потребителски интерфейс	Ученикът: – знае предназначението на основни контейнери и контроли – форма, етикет,	Ученикът: – създава приложение с ГПИ (Hello, Java!); – проектира чрез средства за визуално програмиране	

(1)	(2)	(3)	(4)	(5)	(6)
			<p>текстово поле, бутон, диалогова кутия;</p> <ul style="list-style-type: none"> <li>– разпознава основни свойства на графични обекти-контроли – име, състояние, етикет, фон, настройка на шрифт и др.;</li> <li>– проектира несложна форма, съдържаща етикет, текстово поле, бутон;</li> <li>– настройва основни свойства на форма, етикет, текстово поле и бутон;</li> <li>– именува обекти-контроли съгласно общоприета конвенция;</li> <li>– задава функционалност на бутон, свързана с извеждането на статично съобщение в диалогова кутия.</li> </ul>	<p>собствена форма, отговаряща на изискванията на графичен дизайн;</p> <ul style="list-style-type: none"> <li>– управлява свойства на основни контроли;</li> <li>– описва метод, асоцииран със събитие на бутон.</li> </ul> <p>Практическа работа.</p>	
14	7	2.3.2. Управление на разположението на компонентите		<p>Ученикът:</p> <ul style="list-style-type: none"> <li>– усъвършенства проекта Hello, Java! чрез добавяне на функционалност на останалите бутони;</li> <li>– тества изгледи на приложение при различни разположения на контролите върху контейнер (Layouts) – променя стойности на свойства на контроли чрез програмен код;</li> <li>– създава самостоятелно прости приложения с ГПИ.</li> </ul> <p>Практическа работа.</p>	
<b>ТЕМА 3. ПРОГРАМИРАНЕ</b>					
<b>3.1. Основни типове данни</b>					
15	8	3.1.1. Тип низ	<p>Ученикът:</p> <ul style="list-style-type: none"> <li>– познава правила за именуване на константи и променливи;</li> <li>– декларира, описва и инициализира променливи и константи от тип низ;</li> <li>– присвоява стойност на променлива от</li> </ul>	<p>Ученикът създава с помощ приложение с ГПИ, което извежда персонализиран поздрав в текстово поле и стандартна диалогова кутия.</p> <p>Участие в беседа. Практическа работа.</p>	
16	8	Тип низ – упражнение	<p>тип низ;</p> <ul style="list-style-type: none"> <li>– въвежда и извежда данни от тип низ в/от текстово поле;</li> <li>– извежда данни от тип низ в/от етикет;</li> </ul>	<p>Ученикът самостоятелно създава:</p> <ul style="list-style-type: none"> <li>– приложение с ГПИ, което симулира работата на телевизор в хотел;</li> <li>– приложение, което извежда рекламен афиш за</li> </ul>	

(1)	(2)	(3)	(4)	(5)	(6)
			<ul style="list-style-type: none"> <li>– извършва конкатенация на низове;</li> <li>– използва стандартни методи на интерфейсни компоненти за форматиране на текст.</li> </ul>	<p>постановка на ученически театър.</p> <p>Практическа работа.</p>	
17	9	3.1.2. Целочислени типове данни	<p>Ученикът:</p> <ul style="list-style-type: none"> <li>– декларира, описва и инициализира променливи и константи от целочислен тип данни;</li> <li>– използва вградени функции за преобразуване на низ в цяло число и обратното;</li> <li>– въвежда и извежда данни от целочислен тип;</li> <li>– използва различни целочислени типове данни;</li> <li>– познава целочислените аритметични операции и техния приоритет;</li> <li>– конструира аритметични изрази, съдържащи само целочислени данни, спазвайки синтаксиса и семантиката на конкретния език за програмиране;</li> <li>– прилага и анализира резултатите от операциите събиране, изваждане, умножение, деление, намиране на цяла част и остатък от целочислено деление;</li> <li>– реализира модел за решаване на задачи, базиран на целочислени типове данни.</li> </ul>	<p>Ученикът:</p> <ul style="list-style-type: none"> <li>– създава с помощ на приложение с ГПИ за намиране на периметъра на правоъгълник;</li> <li>– самостоятелно преобразува приложението за намиране на периметъра на друга геометрична фигура.</li> </ul> <p>Участие в беседа.</p> <p>Коментиране на примери и очаквани резултати от изпълнението на команди.</p> <p>Практическа работа.</p>	
18	9	Целочислени типове данни – упражнение		<p>Ученикът:</p> <ul style="list-style-type: none"> <li>– създава с помощта на упътване в учебника приложение за намиране на сумата на монетите в касичка;</li> <li>самостоятелно създава приложение с ГПИ за пресмятане на необходимия брой опаковки с продукти за изпълнение на готварска рецепта.</li> </ul> <p>Практическа работа.</p>	
19	10	3.1.3. Реални типове данни	<p>Ученикът:</p> <ul style="list-style-type: none"> <li>– декларира, описва и инициализира</li> </ul>	<p>Ученикът създава приложение с ГПИ за пресмятане на началната скорост при изпреварване при зададени</p>	

(1)	(2)	(3)	(4)	(5)	(6)
			<p>променливи и константи от реален тип;</p> <ul style="list-style-type: none"> <li>– използва вградени функции за преобразуване на низ в реално число и обратното;</li> <li>– въвежда и извежда данни от реален тип данни;</li> <li>– използва различни реални типове данни;</li> <li>– познава приоритетите на аритметичните операции при реални типове данни;</li> <li>– конструира аритметични изрази, съдържащ реални типове данни, спазвайки синтаксиса и семантиката на конкретния език за програмиране;</li> <li>– прилага и анализира резултатите от операциите събиране, изваждане, умножение и деление.</li> <li>– реализира модел за решаване на задачи, базиран на реални типове данни.</li> </ul>	<p>ограничения.</p> <p>Практическа работа.</p>	
20	10	Реални типове данни – упражнение		<p>Ученикът:</p> <ul style="list-style-type: none"> <li>– създава приложение, което пресмята скоростта и ускорението на асансьор при зададени от потребителя разстояние и време за изкачване;</li> <li>проучва информация за асансьорите в известни сгради.</li> </ul> <p>Практическа работа.</p>	
<b>3.1.4. Аритметични изрази и вградени математически функции. Приоритет на операциите</b>					
21	11	3.1.4.1. Аритметични изрази в език за програмиране	<p>Ученикът:</p> <ul style="list-style-type: none"> <li>– оценява числената стойност на аритметичен израз, записан на език за програмиране;</li> </ul>	<p>Ученикът:</p> <ul style="list-style-type: none"> <li>– създава приложение за конвертиране на валута.</li> </ul> <p>Участие в беседа.</p> <p>Практическа работа.</p>	
22	11	Аритметични изрази в език за програмиране – упражнение	<ul style="list-style-type: none"> <li>– записва аритметичен израз със средствата на език за програмиране;</li> <li>– прилага и използва вградени в езика за програмиране математически функции – абсолютна стойност, повдигане на степен, закръгляване, извличане на цялата част на</li> </ul>	<p>Ученикът:</p> <ul style="list-style-type: none"> <li>– създава приложение за намиране на броя стандартни опаковки на хранителни продукти, необходими за изпълнението на готварска рецепта;</li> <li>– създава приложение за пресмятане на дължимите данъци и осигуровки по дадена заплата.</li> </ul>	

(1)	(2)	(3)	(4)	(5)	(6)
			реално число; – използва приоритет на операциите в аритметични изрази, съдържащи вградени функции; – описва синтаксис и семантика на оператор за присвояване.	Устно: анализ и разработка на математически модел. Практическа работа.	
23	12	3.1.4.2. Аритметични изрази, съдържащи различни типове данни	Ученикът: – създава аритметични изрази, съдържащи различни типове данни, като се съобразява със съвместимостта им; – форматира изхода на реално число.	Ученикът създава приложение за намиране на най-къс маршрут в 2D и 3D пространство при определени условия.  Практическа работа.	
24	12	Аритметични изрази, съдържащи различни типове данни – обобщение	Ученикът: – си припомня и затвърждава изучените в темата понятия; – си припомня и отработва изучените конструкции; – работи уверено по практически задачи от темата.	Ученикът създава приложение с ГПИ за изчисляване на индекса „усеща се“ на температура при зададени реална температура и скорост на вятъра.  Практическа работа.	
25	13	Аритметични изрази, съдържащи различни типове данни – обобщение  <b>3.1. Тест</b>	Проследява се текущото развитие на ученика.	Ученикът: – създава приложение с ГПИ за изчисляване на индекса „усеща се“ на температура при зададени реална температура и влажност на въздуха; – решава тест за самопроверка.  Практическа работа.  Тест за самопроверка.	Тестът за самопроверка се прави в клас, а този за оценяване – в началото на следващото занятие.
<b>3.2. Създаване на компютърна програма за решаване на конкретна задача</b>					
26	13	3.2.1. Оформяне на програмния код. Коментари	Ученикът спазва изисквания за оформяне на програмния код, включващи подравняване, коментари, именуване на програмните единици.	Ученикът добавя коментари и оформя код в готово приложение.  Практическа работа.	Започва се с тест върху тема 3.1.
27	14	3.2.2. Създаване на компютърна програма за решаване на конкретна задача	Ученикът: – знае основните етапи при създаване на компютърна програма; – анализира и проектира решението на	Ученикът: – проектира и създава приложение с ГПИ – симулатор на касов апарат; – тества и отстранява грешки в приложението.	



(1)	(2)	(3)	(4)	(5)	(6)
			<p>конкретна задача;</p> <ul style="list-style-type: none"> <li>– създава математическия модел за решаване на задачата;</li> <li>– разработва алгоритъм за решаване на задачата;</li> <li>– определя входно-изходни данни и техните типове;</li> <li>– структурира и разработва графичен интерфейс, като използва обекти и декларира променливи;</li> <li>– създава и описва програмния код;</li> <li>– стартира, тества и валидира готовия проект;</li> <li>– открива синтактични и логически грешки в програмата;</li> <li>– отстранява синтактични и логически грешки при програмиране;</li> <li>– спазва изисквания за оформяне на програмния код, включващи подравняване, коментари, именуване на програмните единици.</li> </ul>	<p>Практическа работа.</p>	
28	14	3.2.3. Създаване на компютърна програма – практически задачи		<p>Ученикът:</p> <ul style="list-style-type: none"> <li>– създава приложение с ГПИ за пресмятане на броя ролки тапети, необходими за облепване на стая;</li> <li>– създава приложение с ГПИ, симулиращо работата на автомат за напитки.</li> </ul> <p>Практическа работа.</p>	
<b>3.3. Програмни конструкции за реализация на разклонен алгоритъм</b>					
<b>3.3.1. Булев тип данни</b>					
29	15	3.3.1.1. Булеви изрази	<p>Ученикът:</p> <ul style="list-style-type: none"> <li>– обосновава необходимостта от разклоняване на алгоритмичния процес;</li> <li>– използва константите от булев тип данни;</li> <li>– дава примери, в които се използва булев тип данни;</li> </ul>	<p>Ученикът описва булеви изрази, свързани с условия за пране и гладене на дрехи.</p> <p>Устно: участие в беседа;</p> <p>писмено: описание на булеви изрази.</p> <p>Практическа работа.</p>	

(1)	(2)	(3)	(4)	(5)	(6)
			<ul style="list-style-type: none"> <li>– декларира булева променлива;</li> <li>– присвоява стойност на булева променлива.</li> </ul>		
30	15	Булеви изрази. Операции за сравнение – упражнение	Ученикът записва на език за програмиране булев израз, съдържащ операция за сравнение.	<p>Ученикът дефинира булеви изрази за проверка дали:</p> <ul style="list-style-type: none"> <li>– дадено число е „квадратно“;</li> <li>– дадено число удовлетворява условията на математически трик.</li> </ul> <p>Практическа работа.</p>	
31	16	3.3.1.2. Логически операции	<p>Ученикът:</p> <ul style="list-style-type: none"> <li>– изписва синтактично правилно на езика за програмиране на основните логически операции – логическо отрицание, дизюнкция, конюнкция;</li> <li>– познава приоритета на логическите операции;</li> <li>– пресмята без използване на компютър стойността на булев израз.</li> </ul>	<p>Ученикът описва съставен булев израз за проверка дали определени климатични условия удовлетворяват зададени изисквания.</p> <p>Устно: участие в беседа или дискусия; писмено прилагане на логически операции при формиране на съставен булев израз. Практическа работа.</p>	
32	16	Съставяне със средствата на език за програмиране сложен булев израз – упражнение	Ученикът съставя със средствата на език за програмиране сложен булев израз, отговарящ на дадена логическа ситуация.	<p>Ученикът:</p> <ul style="list-style-type: none"> <li>– описва съставни логически изрази, проверяващи дали:</li> <li>– шахматна фигура „заплашва“ поле от шахматната дъска;</li> <li>– дадено поле от шахматната дъска е черно.</li> </ul> <p>Писмено: описание на съставен булев израз. Практическа работа.</p>	
33	17	3.3.2. Условен оператор	<p>Ученикът:</p> <ul style="list-style-type: none"> <li>– описва синтаксиса и семантиката на кратка и пълна форма на условен оператор;</li> <li>– описва разклонен алгоритъм с помощта на условен оператор;</li> <li>– използва условен оператор за проверка на коректността на входните данни за програма;</li> <li>– използва условен оператор за обработка на свойства на радиобутон и поле за отметка.</li> </ul>	<p>Ученикът:</p> <ul style="list-style-type: none"> <li>– създава приложение с ГПИ за определяне на вида на триъгълник;</li> <li>– проверява коректност на входните данни за приложение.</li> </ul> <p>Устно: анализ и създаване на математически модел. Практическа работа.</p>	

(1)	(2)	(3)	(4)	(5)	(6)
34	17	Условен оператор – упражнение		Ученикът: – създава приложение с ГПИ за изчисляване на цената на услуга във фризьорски салон; – използва полета за отметка за избор на набор от услуги.  Устно: анализ и създаване на математически модел. Практическа работа.	
35	18	3.3.3. Вложени условни оператори	Ученикът: – 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.	Ученикът създава приложение с ГПИ за пресмятане на оценката от тест с избираеми отговори.  Устно: анализ и създаване на математически модел. Практическа работа.	
36	18	Вложени условни оператори – упражнение		Ученикът създава приложение с ГПИ за: – определяне на вида на електромагнитна вълна по зададена дължина на вълната; – определяне на вида на шум по зададени децибелни.  Практическа работа.	

**ВТОРИ УЧЕБЕН СРОК – 18 седмици x 2 часа в блок седмично = 36 часа**

№ по ред	Учебна седмица по ред	Тема на урочната единица	Очаквани резултати от обучението	Методи и за работа	Бележки/Коментари
(1)	(2)	(3)	(4)	(5)	(6)
37	19	Програмна реализация на разклонени алгоритми – обобщение	Ученикът създава и реализира алгоритъм за решаване на задача чрез използване на условни конструкции.	Ученикът създава приложение с ГПИ за: – намиране на корените на квадратно уравнение; – пресмятане на цената за отпечатване на документи; географски тест игра – познаване на столиците на държави.  Практическа работа.	Една от задачите се използва за контрол и оценка в началото на следващото занятие.
38	19	<b>3.3. Тест</b>	Ученикът проверява в каква степен е овладял изучените в темата понятия.	Практическа работа.. Тест за самопроверка	През първата половина на часа се прави кратко обобщение със задача, а през втората – тест.
<b>3.4. Програмни конструкции за реализация на циклични алгоритми</b>					
<b>3.4.1. Циклични алгоритми</b>					
39	20	3.4.1.1. Цикъл, управляван от условие	Ученикът: – 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 постусловие; – записва синтактично и логически правилно оператори за цикъл с предусловие и постусловие.	Ученикът създава приложение с ГПИ, което проверява валидността на код за сейф. Участие в беседа. Практическа работа.	
40	20	Цикъл, управляван от условие – упражнение		Ученикът решава практически задачи, които се моделират чрез НОД и НОК: – намиране на максималния брой еднакви букети;	

(1)	(2)	(3)	(4)	(5)	(6)
				– намиране на минималното време за обиколка на писта от двама състезатели, така че те отново да се засекат на изходната позиция.  Практическа работа.	
41	21	3.4.1.2. Цикъл, управляван от брояч	Ученикът: – посочва елементите на циклична конструкция – инициализация, тяло на цикъла и условие на цикъла; – записва синтактично и логически правилно оператори за цикъл, управлявани от брояч.	Ученикът създава приложение с ГПИ за намиране на сложна лихва.  Участие в беседа/ дискуссия. Практическа работа.	
42	21	Цикъл, управляван от брояч – упражнение		Ученикът създава приложение с ГПИ за: – намиране на поредното число в редицата на Фибоначи; намиране на приближение на Златното сечение.  Практическа работа.	
43	22	3.4.1.3. Списъчно поле	Ученикът прилага алгоритми с циклични конструкции за проверка на входни данни.	Ученикът създава приложение с ГПИ за: – разделяне на списък от ученици по групи; – пресмятане на стойността на комунални услуги по зададен списък от разходи, като се използва списъчно поле за въвеждане на данните.  Практическа работа.	
44	22	3.4.1.4. Сравнение между циклични алгоритмични конструкции	Ученикът: – оценява необходимостта от използване на алгоритми с циклични конструкции с условие или управлявани от брояч; – преобразува програмен код, съдържащ циклична конструкция, управлявана от брояч, в циклична конструкция, управлявана от условие.	Ученикът проектира приложение с ГПИ за вземане на решение по кои предмети да се оформи група за факултативно обучение.  Участие и аргументация при обсъждане/дискусия за избор на оператор за цикъл. Практическа работа.	Проектът е обемен и обобщителен за темата. Учениците работят по него в продължение на три учебни часа.
45	23	3.4.1.4. Сравнение между циклични алгоритмични конструкции	Ученикът открива синтактични и логически грешки в програмния код на алгоритми с циклична конструкция.	Ученикът създава приложение с ГПИ за: – вземане на решение по кои предмети да се оформи група за факултативно обучение; – изчислява общия брой сформирани групи,	

(1)	(2)	(3)	(4)	(5)	(6)
				като прави обоснован избор относно оператора за цикъл, който да използва; – заменя един вид оператор за цикъл с друг и наблюдава разликите при изпълнение на програмата. Участие и аргументация при обсъждане/дискусия за избор на оператор за цикъл. Практическа работа.	
46	23	3.4.1.4. Сравнение между циклични алгоритмични конструкции	Ученикът прилага циклични алгоритми за управление на графичен потребителски интерфейс.		
<b>3.4.2. Приложение на условни и циклични конструкции</b>					
47	24	3.4.2.1. Компютърна графика	Ученикът прилага циклични алгоритми за изчертаване на графични примитиви.	Ученикът създава приложения с ГПИ за изчертаване на модели, базирани на повтарящи се геометрични фигури.  Участие с аргументи при създаването на математически модел. Практическа работа.	
48	24	3.4.2.1. Компютърна графика			
49	25	3.4.2.2. Вход и изход от файл	Ученикът използва циклични алгоритми за въвеждане и извеждане на данни от файл.	Ученикът създава приложение с ГПИ, което въвежда координати на точки от файл, трансформира го спрямо КС на Java, записва новите координати в друг файл и изчертава геометрични фигури чрез свързване на точките.  Участие с аргументи при създаването на математически модел. Практическа работа.	
50	25	3.4.2.3. Работа с изображения		Ученикът създава приложение с ГПИ, което представлява електронна рамка за снимки, и използва изображения, записани във файлове.  Участие с аргументи при създаването на математически модел. Практическа работа.	

(1)	(2)	(3)	(4)	(5)	(6)
51	26	Условни и циклични конструкции – обобщение	Ученикът прилага програмни конструкции за реализация на алгоритми за намиране на сума, минимален/максимален елемент, средно аритметично и др. в редици от числа, въведени от потребителския интерфейс клавиатурата.	Ученикът създава приложение с ГПИ, което: – въвежда данни от файл за климата на дадено населено място; – намира минимална и максимална годишна температура; – намира средна температурна стойност; – намира годишна сума на валежите.  Устно: участие в обсъждане на постановката, анализ, извеждане на алгоритмите. Практическа работа.	Задачата се използва за проверка и оценка на знанията и уменията през следващия учебен час.
52	26	3.4.2.4. Условни и циклични конструкции – обобщение			
53	27	3.4.2.4. Условни и циклични конструкции – обобщение			
54	27	<b>3.4. Тест</b>	Ученикът проверява в каква степен е овладял изучените в темата понятия.	Ученикът: – представя завършен проект; – решава тест за самопроверка.  Представяне на завършен проект.  Тест за самопроверка.	Представя се проектът от миналите занятия.
55	28	<b>3.5. Тестване и верификация на програма</b>	Ученикът: – 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.	Ученикът: – описва спецификация за ASCII Art приложение; – създава ASCII Art приложение; – генерира тестови данни; – тества приложението; – отстранява грешки.  Устно: участва с аргументи в обсъждането на възможните причини за наличие на логическа грешка. Практическа работа.	
56	28	Тестване и верификация на програма – упражнение		Ученикът разработва, тества и верифицира самостоятелно приложения за: – изчисляване на обем на геометрично тяло;	

(1)	(2)	(3)	(4)	(5)	(6)
				създаване на ASCII Art изображение по избор. Практическа работа.	
<b>3.6. Съставни типове данни. Едномерен масив</b>					
<b>3.6.1. Едномерен масив</b>					
57	29	3.6.1.1 Дефиниране и инициализиране на едномерен масив	Ученикът: – 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.	Ученикът разработва приложение с ГПИ, което следи броя на писмата в пощенска кутия.  Устно: – участие в обсъждане/дискусия; – отговори на въпроси. Практическа работа.	
58	29	3.6.1.2 Въвеждане, извеждане и обхождане на едномерен масив			
<b>3.6.2. Основни алгоритми за работа с едномерен масив</b>					
59	30	3.6.2.1. Сума и произведение на елементите на едномерен масив	Ученикът пресмята сбор и произведение на стойностите на елементите на едномерен масив.	Ученикът разработва приложения с ГПИ за: – пресмятане на среден успех в училище; – намиране на вероятността за съществуване на извънземен разум по уравнението на Дрейк; – пресмятане на стойността на многочлен; – пресмятане на общия и средния разход на домакинство.  Устно: – участие в обсъждане/ дискусия; – отговори на въпроси. Практическа работа.	



(1)	(2)	(3)	(4)	(5)	(6)
60	30	Сума и произведение на елементите на едномерен масив – упражнение			
61	31	3.6.2.2. Търсене на минимална и максимална стойност в едномерен масив	Ученикът търси елемент от масива с максимална и минимална стойност.	Ученикът разработва приложение с ГПИ за намиране на резултатите от спортно състезание.  Устно: – участие в обсъждане/ дискусия; – отговори на въпроси.  Практическа работа.	
62	31	Търсене на минимална и максимална стойност в едномерен масив – упражнение			
63	32	3.6.2.3. Последователно търсене на елемент от масив, отговарящ на дадено условие	Ученикът търси елементи от масива, отговарящи на дадено условие.	Ученикът разработва приложение с ГПИ за игра „Бесеница“.  Устно: – участие в обсъждане/дискусия; – отговори на въпроси.  Практическа работа.	
64	32	Основни алгоритми за работа с едномерен масив – обобщение	Ученикът: – си припомня и затвърждава изучените в темата понятия; – си припомня и отработва изучените конструкции; – работи уверено по практически задачи от темата.	Ученикът разработва приложения с ГПИ за: – класифициране на климат по зададени месечни температури; – определяне на годишен индекс на засушаване на регион по зададени месечни температури и суми на валежите; – представя проект.  Устно: – участие в обсъждане/дискусия; – отговори на въпроси. Практическа работа.	

(1)	(2)	(3)	(4)	(5)	(6)
<b>ТЕМА 4. Създаване на софтуерно приложение</b>					
65	33	<b>3.6. Тест</b>  4.1. Създаване на софтуерен проект	Ученикът: – проверява в каква степен е овладял изучените в темата понятия; – 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.	Ученикът: – решава тест за самопроверка; – отговаря на въпроси; – работи в екип за изпълнение на практическо задание за пресмятане на цената на дограма.  Тест за самопроверка. Практическа работа и степен на участие на всеки ученик за реализацията на проекта на екипа.	Тестът се прави в началото на часа.
66	33	4.1. Създаване на софтуерен проект			
67	34	4.2. Работа в екип при създаване на софтуерен проект		Ученикът: – отговаря на въпроси; – работи в екип за изпълнение на практическо задание за управление на складовата наличност в склад за плодове и зеленчуци; за проверка на валидността на номер на кредитна карта; за представяне на код на цвят в различни системи; за пресмятане на популацията на биологични видове; – работи в екип по подготовката на документацията и презентацията на проекта.  Практическа работа и степен на участие на всеки ученик за реализацията на проекта на екипа. Спазване на срокове. Разпределение на ролите и отговорностите в екипа.	Работи се в екипи от двама ученици.
68	34	4.2. Работа в екип при създаване на софтуерен проект			

(1)	(2)	(3)	(4)	(5)	(6)
69	35	4.2. Работа в екип при създаване на софтуерен проект			
70	35	4.3. Презентация и защита на готов софтуер проект			
71	36	4.3. Презентация и защита на готов софтуер проект	Ученикът презентира и защитава готовия софтуерен проект.	Ученикът представя заедно с екипа си разработения проект.  Степен на участие на ученика, умения за представяне и аргументиране	Учениците представят екипно разработените от тях проекти.
72	36	4.3. Презентация и защита на готов софтуер проект  <b>Исходно ниво</b>		Ученикът: – представя заедно с екипа си разработения проект; – решава тест.  Презентация със защита.  Тест.	

Разработил: .....

(Име, фамилия, подпис)

## ПОЯСНИТЕЛНИ БЕЛЕЖКИ:

1. Годишното тематично разпределение се разработва от преподаващия учител за всяка учебна година и за всеки клас (а при необходимост – и по паралелки), като се отчитат интересите на учениците и спецификата на образователната среда.
2. Годишното тематично разпределение на учителя по т. 1 се утвърждава от директора на училището преди началото на учебната година.
3. В колона 1 се записва поредният номер на учебния час. Броят на учебните часове в тематичното разпределение трябва да отговаря на броя на часовете по училищен учебен план за съответния клас.
4. В колона 2 се посочва учебната седмица по ред, като следва да се отчита броят на учебните седмици по заповед на министъра за графика на учебното време.
5. В колона 3 се посочва темата на урочната единица, като тя трябва да отговаря на темата, записана в дневника. Темата на урочната единица се определя от учителя и може да не е същата като темата на урока в учебника или темата в учебната програма.
6. В колона 4 се описват накратко компетентностите като очаквани резултати от обучението в рамките на конкретната урочна единица.
7. В колона 5 се посочват методите и формите за оценяване (те може да са свързани с конкретната тема на урочната единица, но може да са и ориентирани върху цял раздел) при спазване на ДОС за оценяване на резултатите от обучението на учениците, както и за оценяване на другите дейности (домашни работи, лабораторни упражнения, семинари, работа по проекти и др.) и при отчитане на съотношението при формиране на срочна и годишна оценка в раздел „Специфични методи и форми за оценяване на постиженията на учениците“ на съответната учебна програма.
8. При възникнали обстоятелства от обективен характер годишното тематично разпределение подлежи на изменение, допълнение и реструктуриране, което се отразява в колона 6 или в допълнителна таблица и се утвърждава допълнително от директора на училището при спазване на препоръчителното процентно разпределение на задължителните учебни часове за годината.