

# **УЧЕБНА ПРОГРАМА ПО “ВЪВЕДЕНИЕ ВЪВ ВИЗУАЛНОТО ПРОГРАМИРАНЕ” ЗА II КЛАС**

**(ВЪВЕЖДАНЕ НА ДИСЦИПЛИНАТА ПО ПРОЕКТ “ИНОВАТИВНО УЧИЛИЩЕ”)**

## **КРАТКО ПРЕДСТАВЯНЕ НА УЧЕБНАТА ПРОГРАМА**

Обучението по **Въведение във Визуалното Програмиране** в начален етап е насочено към овладяване на базисни знания, умения и отношения, свързани с учебния предмет, с изграждането на дигитални компетентности на ученика и с приложението им в областта на дигиталните технологии.

В този клас се придобиват първоначални систематизирани знания и умения за циклите в програмирането. Формират се нови знания и умения за работа в среда за визуално програмиране чрез блокчета за визуално програмиране. Акцентът в обучението във II клас е върху използването на образователни игрови по визуално програмиране, целящи да формират знания и умения за използване на основополагащи понятия в програмирането на различните програмни езици, като се акцентира на Ruby.

Учебното съдържание е представено в следните основни теми: □

- Преговор на цикъл за повторение
- Работа с цикъл until.

- Работа с цикъл while в среда на визуално програмиране.
- Логически задачи: Отрицание в програмирането: while not, until not

Използване на променливи във визуалното програмиране

В програмата са включени въвеждащи теми за надграждане на материала от предишния клас. Учениците правят преговор на изучавания цикъл, и се запознават с нови цикли в програмирането. Решават елементарни логически задачи, като използват отрицанието в кодирането. Запознават се с променливите в програмирането, като аналог на неизвестното в математиката. Учебното съдържание се доразвива в програмите за следващите класове.

### **ОЧАКВАНИ РЕЗУЛТАТИ ОТ ОБУЧЕНИЕТО В КРАЯ НА КЛАСА**

В края на обучението в II клас ученикът: ☒

- Познава здравните норми при работа с дигитални устройства
- Познава работното поле и се ориентира в конкретна визуална среда
- Подрежда блокове в правилна последователност, за да достигне до вярно решение

- Знае, че в програмирането съществуват повече от едно вярно решение, за поставен конкретен проблем
- Подрежда блокове за визуално програмиране, като извършва действията плъзгане и пускане, изтриване на блок, дублиране на блок.
- Познава 3 вида цикли в програмирането.

Сглобява крайна последователност от блокове, за изграждане на цикли за повтаряне на код times.do, until, while

- Открива грешки в готов код - последователност от блокове, и ги коригира
- реагира на съобщенията, извеждани от използваното приложение, и коригира своя код съобразно забележките
- обяснява основните предназначения на използваните блокчета
- зарежда интернет страници в специализирана програма чрез въвеждане на адрес
- Може да обясни базовата употреба на циклите в програмирането.
- Може да направи съпоставка между променливите в програмирането, и неизвестното в математиката

## УЧЕБНО СЪДЪРЖАНИЕ

Теми	Компетентности като очаквани резултати от обучението	Нови понятия
<b>Тема 1: Преговор на цикъл за повторение</b>		
<b>1.1. Преговор на цикъл за повторение.</b>	<ul style="list-style-type: none"> <li>• Познава работното поле на конкретна визуална среда: работи с блокове за прости команди, и с блокове за цикли               <ul style="list-style-type: none"> <li>• Знае къде се намират допълнителните блокчета за визуално програмиране</li> <li>• Знае как да добавя, изтрива, дублира блокче.</li> </ul> </li> <li>Ориентира се в цялостния прозорец на средата               <ul style="list-style-type: none"> <li>• Може да обясни с пример как работи цикъл за повторение times.do                   <ul style="list-style-type: none"> <li>• стартира, използва и приключва работа със средата за визуално програмиране, ориентира се със системата за изпращане на съобщения и комуникации</li> <li>• осъществява диалог със системата и останалите ѝ потребители, като използва елементите на потребителския интерфейс</li> </ul> </li> </ul> </li> </ul>	<p>цикъл until</p> <p><i>Забележка:</i> Понятията се въвеждат в темата, но оперирането с тях и затвърдяването им продължават във всички теми, свързани с използване на софтуерни приложения.</p>

Тема 2: Работа с цикъл until.		
<b>2.1 Запознава се с начина, по който изглежда блокче until</b>	<ul style="list-style-type: none"> <li>• Познава структурата на цикъла в средата за визуално програмиране</li> <li>• Знае, че е необходимо да използва блок за условие на цикъла</li> <li>• Знае, че е необходимо да добави код, за тяло на цикъла</li> </ul>	цикъл until, условие на цикъл, тяло на цикъл
<b>2.2 Използване на блок until в среда за визуално програмиране</b>	<ul style="list-style-type: none"> <li>• Работи с предоставените в интерфейса блокове, като ги подрежда в нужната последователност</li> <li>• Подрежда познатите блокове, за решаване на нестандартни задачи/условия</li> <li>• Може да преработи код, съставен от единични блокчета - като използва блок until.</li> <li>• Може да използва цикъла многократно, като активира пвоече от един герой.</li> </ul>	скъсяване и оптимизиране на код
<b>2.2 Активиране на повече от един</b>	<ul style="list-style-type: none"> <li>• Използва правилните блокове, за активиране на второстепенен герой.</li> </ul>	Оптимално решение, варианти на решение

<p><b>герой</b></p>	<ul style="list-style-type: none"> <li>• Разбира, че задачите могат да имат пвоече от едно вярно решение.</li> <li>• Търсене на решение с най-кратък код</li> </ul>	
<p><b>Тема 3: Работа с цикъл while в среда на визуално програмиране</b></p>		
<p><b>3.1 Запознава се с начина, по който изглежда блокче while</b></p>	<ul style="list-style-type: none"> <li>• Използва блокове от визуалната среда за построяване на цикъл while в програмирането</li> <li>• Разбира необходимостта от използване на цикли, за спестяване на писането на дълъг код</li> </ul>	
<p><b>3.2 Построява цикъл while за повторно изпълнение на код</b></p>	<ul style="list-style-type: none"> <li>• Използва блок за програмиране, което указва комбинирано движение на героя</li> <li>• Съчетава комбинирано блокче за програмиране с блокче за цикъл</li> </ul>	
<p><b>3.3 Използване на цикли, в съчетание с нови команди</b></p>	<ul style="list-style-type: none"> <li>• Използва блок за програмиране, което указва комбинирано движение на героя</li> <li>• Съчетава комбинирано блокче за програмиране с блокче за цикъл</li> </ul>	

	<ul style="list-style-type: none"> <li>• Може да избере кой вид цикъл да използва за решаване на конкретно поставена задача, като преценява наличното в средата за визуално програмиране</li> </ul>	
<b>Тема 4: Логически задачи: Отрицание в програмирането: while not, until not</b>		
<b>4.1 Решава елементарни логически задачи</b>	<ul style="list-style-type: none"> <li>• Умее да съобрази как функционира блокче за отрицание not</li> <li>• Умее да пренапише код, като използва заместващ цикъл, и запази същата функционалност</li> <li>• Умее да открие и коригира грешки в код</li> </ul>	логическа задача
<b>Тема 5: Използване на променливи във визуалното програмиране</b>		
<b>5.1 Запознаване с променливите в програмирането</b>	<ul style="list-style-type: none"> <li>• Знае как да въведе име на променлива във визуалната среда</li> <li>• знае къде се намира, и как да използва блокчето на променливите</li> <li>• Може да въвежда стойност на променлива</li> <li>• може да съпостави променливите в програмирането, като</li> </ul>	променливи, стойност на променлива

	аналог на неизвестното в математиката	
<b>5.2 Предметите като променливи</b>	<ul style="list-style-type: none"> <li>• Знае, че променливите могат да приемат за стойност число, или предмет</li> <li>• Използване на променливите за писане на безопасен код</li> <li>• Използва променливите за поставяне на условие в цикъл</li> </ul>	

**ПРЕПОРЪЧИТЕЛНО ПРОЦЕНТНО РАЗПРЕДЕЛЕНИЕ НА ЗАДЪЛЖИТЕЛНИТЕ УЧЕБНИ ЧАСОВЕ ЗА ГОДИНАТА**

<b>За нови знания и умения</b>	<b>50 %</b>
<b>За упражнения в лабораторна среда</b>	<b>30 %</b>
<b>За обобщение и затвърждаване на новите знания</b>	<b>14 %</b>



За диагностика на входното и изходно ниво	6 %
---	-----

## СПЕЦИФИЧНИ МЕТОДИ И ФОРМИ ЗА ОЦЕНЯВАНЕ НА ПОСТИЖЕНИЯТА НА УЧЕНИЦИТЕ

### Допълнителни уточнения за конкретния учебен предмет

Обучението се осъществява в компютърна зала, или в класната стая. За всеки ученик има самостоятелно работно място, или мобилно дигитално устройство (например таблет, лап-топ, Chrome book или друго).

Знанията и уменията на учениците от втори клас се оценяват предимно чрез резултата от работа във визуалната среда по програмиране. Използват се устни и писмени форми за проверка. За всички форми на оценяване поставената оценка е качествен показател, който може да бъде изразен вербално или невербално.

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

Ключови компетентности	Примерни дейности и междупредметни връзки
<p><i>Компетентности в областта на българския език</i></p>	<ul style="list-style-type: none"> <li>• Въвеждане на текст в определените от дигиталната или визуалната среда места.</li> <li>• Общуване в писмен вид с останалите участници в системата за визуално програмиране - изпращане на съобщения.</li> <li>• Анализиране на потенциалните възможности за действия на героите, за решаването на конкретен проблем във визуалното програмиране</li> <li>• Коментиране на възможностите за решаване по различни начини на конкретна проблемна ситуация.</li> </ul>
<p><i>Умения за общуване на чужди езици</i></p>	<ul style="list-style-type: none"> <li>• Използване на блокове, означени както на български, така и на английски език. □</li> <li>• Въвеждане на английските, съвместно с българските наименования, на основните елементи на изучавания приложен софтуер и интерфейс</li> <li>• Използване на последователност от латински букви и/или знаци за означаване имена на променливи.</li> </ul>

<i>Математическа компетентност и основни компетентности в областта на природните науки и на технологиите</i>	<ul style="list-style-type: none"> <li>• прави аналогия между понятията променлива в кодирането, и неизвестно в математиката</li> <li>• Изчисляване на броя на стъпките на героите, чрез използване на изчисление и броене</li> </ul>
<i>Дигитална компетентност</i>	<ul style="list-style-type: none"> <li>• Обработване на информация.</li> <li>• Използване дигитална идентичност.</li> <li>• Прилагане правила за безопасна работа в дигитална среда.</li> <li>• Изучаване на логиката на циклите и променливите в програмирането, чиито правила са в сила за всички програмни езици.</li> </ul>
<i>Умения за учене</i>	<ul style="list-style-type: none"> <li>• Откриване на грешки в собствен и чужд код</li> <li>• Предлагане на повече от едно вярно решение</li> </ul>
<i>Социални и граждански компетентности</i>	<ul style="list-style-type: none"> <li>• Съставяне на програмен код, при който героите работят в екип за постигане на обща цел (ситуации, които включват толерантно общуване на героите).</li> </ul>
<i>Инициативност и предприемчивост</i>	<ul style="list-style-type: none"> <li>• Предлагане на идеи за различни сюжети за образователна игра</li> <li>• Възможност за креативно решаване на предварително поставена задача, чрез инструментите на средата за визуално програмиране</li> </ul>
<i>Културна компетентност и умения за изразяване чрез творчество</i>	<ul style="list-style-type: none"> <li>• Работа с колекция от герои, и предлагане на сюжет за образователна игра, чието решение включва изучаваните IT технологии.</li> </ul>
<i>Умения за подкрепа на устойчивото развитие и за здравословен начин на</i>	<ul style="list-style-type: none"> <li>• Запознаване с правилата за безопасна работа с компютър, чрез заемане на правилна стойка на стола, и правилен режим на обучение и почивка</li> </ul>

*живот и спорт*