**УЧЕБНА ПРОГРАМА ПО “ПРОГРАМИРАНЕ” ЗА XII КЛАС**

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

**КРАТКО ПРЕДСТАВЯНЕ НА УЧЕБНАТА ПРОГРАМА**

Обучението по **Програмиране** в гимназиален 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 систематизирани знания и умения, които надграждат вече придобити умения в областта на програмирането. Формират се нови знания и умения за писане на код в кодов редактор. Акцентът в обучението в XII клас е върху използването на адаптивни учебни материали, целящи да формират знания и умения за използване на езиците Ruby, Python.

Учебното съдържание е представено в следните основни теми:

* Python: Въведение
* Python: Условни конструкции
* Python: Цикли
* Python: Списъци
* Ruby: Въведение
* Ruby: Условни конструкции
* Ruby: Цикли
* Разработване на проект с Ruby
* Разработване на проект с Python

В програмата са включени въвеждащи теми за запознаване с особеностите в синтаксиса на конкретните програмни езици. Основната цел на тези теми е да представят набора от софтуерни средства, които ще бъдат изучавани и използвани за разработването на самостоятелн проект на Ruby, Python.

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

В края на обучението в XII клас ученикът:

* разпознава основните записвания в кода на програмните езици Python и Ruby, и обяснява тяхното предназначение
* демонстрира отношение на отговорен потребител при работа в Интернет среда
* реагира на съобщенията, извеждани от използваното приложение, и коригира своя код съобразно забележките
* прилага съответстващата българска терминология при описание на дейности, свързани със средата за програмиране
* описва и спазва правилата за безопасна работа с компютърна система
* търси и открива причините за проблемен код: безкраен цикъл, неработещи функции, съобщения в конзолата и други.
* използва основните команди за писане на работещ и постигащ предварително зададена цел код
* Разработва самостоятелно, и участва в разработването на групови проекти с изучаваните програмни езици и технологии.
* Изготвя документация за изработените софтуерни проекти
* Презентира своите проекти

**УЧЕБНО СЪДЪРЖАНИЕ**

|  |  |  |
| --- | --- | --- |
| **Теми** | **Компетентности като очаквани резултати от обучението** | **Нови понятия** |
| **Тема 1: Python: Въведение** | | |
|  | | |
| **1.1.Променливи и типове данни** | * Познава екрана на програмата, и може да работи свободно с инструментите на работното поле * Може да извежда информация в конзолата, чрез използване на функцията print * Шпзнава типовете на данните и променливите в конкретния език * Може да конкатенира информация * Подрежда правилно своя код, за да бъде лесен за разчитане. Работи с коментари * Изпозлва оператори за сравнение | Конзола, конзолни съобщения  Печатане в конзола, функцията print  Типове променливи - int, с плаваща запетая, string и boolean. |
|  | | |
| **1.2.Системни функции** | * Може да използва по предназначение Функцията'.len' * Познава употребата на Функциите '.lower()' & '.upper()', и ги използва в своя код * Познава употребата на Функциите min() и max(), и ги използва в своя код | Функцията'.len'  Функциите '.lower()' & '.upper()  Функциите min() и max() |
| **Тема 2: Python: Условни конструкции** | | |
| **2.1. Структура на конструкцията if-else** | * Работи с условна конструкция if-else * Може да обясни разликата между стандартна и вложена if конструкция * Използва коректно вложената конструкцията if - elif - else * Използва логическите оператори **or, and и !** за сравняване на булеви стойност или изрази. * Използва логическите оператори свободно в съчетание в условна конструкция | Конструкцията if - elif - else  **Логически оператори: or, and и !** |
|  | | |
| **2.2. Подредба на кода и Вложен if** | * Знае какво означава Ternary Conditional * Знае от колко части се състои Ternary Conditional * Преработва стандартен код в Ternary, и обратно |  |
| **Тема 3: Python: Цикли** | | |
| **3.1. Структура на while** | * Може да използва съвместно цикъл while с функцията print * Използва и обяснява начина на работа на цикъл while * Използва цикъл while в своя код |  |
| **3.2. while True цикъл** | * Може да използва съвместно цикъл while True с функцията print * Използва и обяснява начина на работа на цикъл while True * Използва цикъл while True в своя код |  |
| **3.3 for цикъл** | * Може да използва съвместно цикъл for с функцията print * Използва и обяснява начина на работа на цикъл for * Използва цикъл for в своя код * Знае кога е подходящо да използва цикъл while; цикъл while True и цикъл for |  |
| **Тема 4: Python: Списъци** | | |
| **4.1 Едномерни списъци** | * Изброява основните операции със списъци * Редактира предварително зададен код с използване на списъци * Взема броя на елементите в едномерен списък, с цел обработка на елементите |  |
| **4.1.3. Обхождане на едномерен и двумерен списък с цикъл for** | * Знае как да *обходи* елементите на едномерен списък чрез използване на цикъл for * Знае как да *обходи* списък със стандартна и съкратена конструкция * Знае как да *търси* елемент в двумерни списъци |  |
| **Тема 5: Ruby: Въведение** | | |
| **5.1. Променливи и типове данни** | * Може да извежда информация в конзолата, чрез използване на функциите print и puts * Конкатенира променливи, за да се свържат в смислени изречения. * Използва свободно типовете променливи, съгласно поставена задача | функции print и puts  Типове променливи - int, double, string и boolean |
| **5.2. Коментари и Математически операции** | * Подрежда правилно своя код, за да бъде лесен за разчитане. |  |
| **5.3 Системни методи** | * Може да използва по предназначение Методът '.length' * Познава употребата на методите '.toLowerCase' & '.toUpperCase', и ги използва в своя код * Познава употребата на методите .min и .max, и ги използва в своя код | Методът '.length'  Методите '.toLowerCase' & '.toUpperCase'  Методите .min и .max |
| **Тема 6: Ruby: Условни конструкции** | | |
| **6.1. Условни конструкции** | * Работи с условна конструкция if-else * Може да обясни разликата между стандартна и вложена if конструкция * Използва коректно вложената конструкцията if -elsif- else | Конструкцията if -elsif- else |
| **6.1.2. Логически оператори: or, and и !** | * Използва логическите оператори за сравняване на булеви стойност или изрази. * Използва логическите оператори свободно в съчетание в условна конструкция * Работи с Case, съкратено записване с Ternary Conditional, оператор when | Операторите and, or , !  Условната конструкция case  Оператор when  условен оператор (?:) |
| **Тема 7: Ruby: Цикли** | | |
| **7.1 while цикъл** | * Може да използва съвместно цикъл while с функциите print и puts * Използва и обяснява начина на работа на цикъл while * Използва цикъл while в своя код |  |
| **7.2 Цикъл begin-while** | * Може да използва съвместно цикъл begin-while с функциите print и puts * Използва и обяснява начина на работа на цикъл begin-while * Използва цикъл begin-while в своя код |  |
| **7.3 Цикъл for** | * Може да използва съвместно цикъл for с функциите print и puts * Използва и обяснява начина на работа на цикъл for * Използва цикъл for в своя код |  |
| **Тема 8: Работа по проекти с Python: Изработка на Quiz** | | |
| **8. 1 Подготовка за започване на проекта** | * Описва етапите при създаването на проекта * Извършва проучване, и посочва технологиите, които са необходими за използване, за изграждане на проект по предварително зададени критерии * Създава модел за решаване на заданието, поставено в проекта |  |
| **8.2 Работа по проект, 1** | * Използва двумерен масив за съхраняване на въпросите и отговорите на теста * Използва цикъл for за обхождане на масива |  |
| **8.3 Работа по проект, 2** | * Използва условни конструкции за проверка на въведени данни * В резултат от проверката, увеличава или намалява точките на играча * Принтира съобщения в конзолата |  |
| **8.4 Изготвяне на документация, защита на проект** | * Изготвя документация за софтуерния проект * Презентира и защитава изготвения софтуерен проект |  |
| **Тема 9: Работа по проекти с Ruby: Изработка на Quiz** | | |
| **9. 1 Подготовка за започване на проекта** | * Описва етапите при създаването на проекта * Извършва проучване, и посочва технологиите, които са необходими за използване, за изграждане на проект по предварително зададени критерии * Създава модел за решаване на заданието, поставено в проекта |  |
| **9.2 Работа по проект, 1** | * Използва двумерен масив за съхраняване на въпросите и отговорите на теста * Използва цикъл for за обхождане на масива |  |
| **9.3 Работа по проект, 2** | * Използва условни конструкции за проверка на въведени данни * В резултат от проверката, увеличава или намалява точките на играча * Принтира съобщения в конзолата |  |
| **9.4 Изготвяне на документация, защита на проект** | * Изготвя документация за софтуерния проект * Презентира и защитава изготвения софтуерен проект |  |

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

**Допълнителни уточнения за конкретния учебен предмет**

Обучението се осъществява в компютърна зала, като за всеки ученик има самостоятелно работно място.

Над 50% от часовете се организират под формата на комбиниран урок, по време на който учениците изпълняват практически задачи.

**Препоръчително разпределение на часовете:**

|  |  |
| --- | --- |
| **За нови знания и умения** | **30** |
| **За упражнения в лабораторна среда** | **56** |
| **За обобщение** | **6** |
| **За контролни работи** | **8** |

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

Проверката и оценката на знанията и уменията в обучението по информационни технологии трябва да бъдат насочени към измерване постигането на заложените в учебната програма очаквани резултати.

Очакваните резултати от обучението са свързани с усвояването на специфична за предмета терминология, практически умения за съставяне на код в среда за визуално програмиране, и в кодов редактор, умения за аргументиране при избора на технологично средство.

Поради спецификата и разнообразния характер на очакваните резултати при оценяването на знанията и уменията на учениците могат да се използват различни методи и средства за проверка и оценка: 

* Тестове, съдържащи въпроси и задачи със структуриран отговор или с ограничена свобода на отговора. Подборът на тестовите задачи трябва да се съобрази с формулираните в учебната програма очаквани резултати. Тестовете дават възможност да се обхване по-голям обем от учебното съдържание за по-кратко време. Могат да се използват за установяване на входно и изходно равнище или контролно, проведено в рамките на 20-25 минути. 
* Решаване на практически задачи, разработване на самостоятелни или групови софтуерни проекти, решението на които се реализира на компютър в час, или под формата на домашна работа. Този тип задачи може да съдържа отделни компоненти, които измерват усвояването на конкретни умения за работа с изучавания софтуер, умения за извличане на информация, умения за създаване на модели, умения за творческо трансформиране и представяне на различни видове информация в дигитален формат и др. 
* Оценяване уменията при работа по проект въз основа на зададената роля на отделния ученик при изпълнение на проекта. 
* Портфолио, което може да съдържа решаваните от ученика практически задачи в часовете, домашни работи, проучвания по дадена тема, тестове. За оформянето на портфолиото учителят може да посочи кои от решаваните практически задачи ще бъдат задължително включени в него и да представи критерии за оценяване на отделните задачи и на портфолиото като цяло. Задачите, включени като задължителни компоненти, трябва да измерват постигането на формулираните в учебната програма очаквани резултати. Портфолиото може да включва и допълнителни задачи.

*Забележка:* Индивидуалното портфолио може да се използва за оценяване на отделен ученик, при условие че всеки ученик работи самостоятелно на компютър, или включва само компоненти, които ученикът разработва самостоятелно – домашни работи, проучвания, тестове.

*Забележка:* Предложените проекти са примерни. Преподавателят може да предложи други, които обхващат разглеждания материал.

**Съотношение при формиране на срочна и годишна оценка:**

|  |  |
| --- | --- |
| Текущи оценки от устни, от писмени и от практически изпитвания върху конкретна задача | 40% |
| Оценки от контролни (теоретични или практически) или изходно ниво | 30% |
| Оценки от работа по проекти и индивидуално портфолио по предварително зададени критерии, домашни работи | 30% |

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

***Дейности за цялата програма, които могат да се включват във всяка тема***

Дейности, свързани с развитие на умения за учене: 

* Поставяне на задачи за работа с фрагменти от учебните помагала или помощната информация с цел самостоятелно запознаване с елементи на изучавания материал и пограмен език. 
* Използване на демонстрации и експериментиране в средата на изучаваното софтуерно приложение.

***Дейности, свързани с развитие на уменията за общуване на чужд език: ***

* Използване на английско-български и българо-английски речник за елементи от интерфейса на изучаваните софтуерни приложения.

***Примерни дейности за отделни раздели и теми***

|  |  |
| --- | --- |
| Ключови компетентности | Примерни дейности и междупредметни връзки |
| Компетентности в областта на българския език | * Въвеждане на текст в определените от дигиталната среда места.  * Анализиране на потенциалните възможности, за решаването на конкретен проблем или проект * Създаване и записване на собствен текст - коментари - за поясняване на създадения код. |
| Умения за общуване на чужди езици | * Използване на команди и код, означени както на български, така и на английски език.  * Въвеждане на английските, съвместно с българските наименования, на основните елементи на изучавания приложен софтуер и интерфейс * Използване на последователност от латински букви и/или знаци за означаване на наименования на методи, означения, коментари |
| Математическа компетентност и основни компетентности в областта на природните науки и на технологиите | * Използване на математически оператори за съставяне на условни конструкции, цикли и означаване на индекси; * Използване на знаци за сравнение при съставяне на тялото на условна конструкция. |
| Дигитална компетентност | * Обработване на информация.  * Разглеждане на допълнителна информация в Интернет, свързана с възможностите на програмните езици - видео материали, печатни, аудио материали. * Използване дигитална идентичност. * Прилагане правила за безопасна работа в дигитална среда, и защита на личния профил в Интернет пространството.  * Създаване на дигитално съдържание.  * Решаване на проблеми с използване на дигитални технологии * Изучаване на логиката на дисциплината Програмиране, чиито правила са в сила за повечето програмни езици. |
| Умения за учене | * Търсене и обработване на информация от различни източници. * Откриване на грешки в собствен и чужд код * Предлагане на повече от едно вярно решение |
| Социални и граждански компетентности | * Разглеждане на информация от сайтове, свързани с безопасно използване на интернет. * Разглеждане на информация от сайтове, свързани с възможностите за приложение на програмните езици |
| Инициативност и предприемчивост | * Планиране, изготвяне и представяне на софтуерен проект по зададена тема. |
| Културна компетентност и умения за изразяване чрез творчество | * Предлага идеи за проекти, чието решение включва изучаваните IT технологии. |
| Умения за подкрепа на устойчивото развитие и за здравословен начин на живот и спорт | * Изработване на проекти - създаване на персонално портфолио, което включва изучените до момента понятия и технологии в програмирането. * Предлагане на идеи за обучителни програми с елементи от познати спортове и демонстриращи здравословен начин на живот и хранене. |