

Управление образования администрации
Губкинского городского округа

Муниципальное бюджетное общеобразовательное учреждение

«Троицкая средняя общеобразовательная школа» Губкинского
района Белгородской области

УТВЕРЖДАЮ:

Директор

МБОУ «Троицкая СОШ»

_____ С.В. Гончарова приказ

№ _____

от «__» _____ 2021 г.

Дополнительная общеразвивающая программа естественнонаучной
направленности

«Азы программирования на языке Python»

Возраст учащихся: 15-17 лет

Срок реализации: 2021 – 2022 учебный год

Составитель: Малыхина А.А.,

учитель информатики

Программа дополнительной общеразвивающей деятельности: программа «Азы программирования на языке Python» является модифицированной. В её основу заложена программа Ушакова Д.М. – Python для школьников.. При этом взятые из неё разделы адаптированы к запросам учащихся и потенциалу образовательного учреждения и педагога.

Автор программы: Малыгина А.А.

Программа рассмотрена и утверждена на заседании педагогического совета от

«__» _____ 2021 г., протокол №__ _____

Председатель _____

подпись

Ф.И.О.

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Направленность программы

Настоящая общеобразовательная общеразвивающая программа дополнительного образования детей «Азы программирования на языке Python» имеет техническую направленность и предназначена для организации внеурочной деятельности по нескольким взаимосвязанным направлениям развития личности, таким как общеинтеллектуальное, общекультурное, социальное. Программа предлагает ее реализацию в факультативной или кружковой форме. Программа ориентирована на обобщение и углубление знаний и умений по курсу информатики основной школы.

Актуальность программы

Python – это язык программирования общего назначения, распространяемый с открытыми исходными текстами. Он оптимизирован для создания качественного программного обеспечения. Язык Python используется сотнями тысяч разработчиков по всему миру в таких областях, как создание веб-сценариев, системное программирование, создание пользовательских интерфейсов, настройка программных продуктов под пользователя, численное программирование и в других. Как считают многие, один из самых используемых языков программирования в мире.

Педагогическая целесообразность

Данная дополнительная общеразвивающая программа направлена на развитие логического и пространственного мышления слушателя, способствует раскрытию творческого потенциала личности, формированию усидчивости и трудолюбия, приобретению практических умений и навыков в области компьютерных технологий, способствует интеллектуальному развитию ребенка.

Цель программы

Основной целью данного учебного курса является ознакомление слушателя с объектно-ориентированным языком программирования Python, с возможностями, синтаксисом языка, технологией и методами

программирования в среде Python, обучение практическим навыкам программирования на языке Python для решения типовых задач математики и информатики, а также при разработке простейших игр.

Задачи программы

Реализация поставленной цели предусматривает решение следующих задач:

- развитие интереса учащихся к изучению программирования;
- знакомство учащихся с основами программирования в среде Python;
- овладение базовыми понятиями теории алгоритмов при решении математических задач;
- формирование навыков работы в системе программирования Python;
- формирование самостоятельности и творческого подхода к решению задач с помощью средств современной вычислительной техники;
- формирование навыков алгоритмического и логического мышления;
- формирование навыков грамотной разработки программ;
- формирование практических навыков решения прикладных задач;

Отличительные особенности данной программы

На занятиях учащиеся познакомятся с теоретическими аспектами и синтаксисом языка, а также обучатся практическим навыкам программирования в среде Python.

Занятия начинаются с практического знакомства со средой программирования Python, далее идет непосредственное изучение синтаксических конструкций языка и отработка навыков применения элементов программирования при решении задач и создании игр, закрепляются полученные знания и навыки программирования, осваиваются новые методы, способы решения задач, рассматриваются более сложные задачи, новые технологии программирования.

Каждая новая тема завершается практическими задачами, способствующими овладению методикой программирования и изучению языка Python.

Программа ориентирована на дополнительное образование учащихся старшего школьного возраста (15 – 17 лет).

Количество часов: 36 академических часа.

Режим занятий по дополнительной образовательной программе проводятся 1 раз в неделю по 1 часу (продолжительность учебного часа – 45 минут).

Формы обучения: очная форма

Формы организации деятельности: групповая, индивидуальная, индивидуально-групповая и фронтальная.

Программой предусмотрено проведение комбинированных занятий. Занятия состоят из теоретической и практической частей, причём большее количество времени занимает именно практическая часть.

2. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ИЗУЧЕНИЯ КУРСА

В результате изучения дисциплины получают дальнейшее развитие личностные регулятивные, коммуникативные и познавательные-универсальные учебные действия, учебная (общая и предметная) и общепользовательская ИКТ-компетентность обучающихся, составляющая психолого-педагогическую, инструментальную основу формирования способности и готовности к освоению систематических знаний, к их самостоятельному пополнению, переносу и интеграции, способности к сотрудничеству и коммуникации, решению личностно и социально значимых проблем и воплощению решений в практику, способности к самоорганизации, саморегуляции и рефлексии.

В ходе изучения данного курса в основном формируются и получают развитие следующие **метапредметные результаты**:

- умение самостоятельно планировать пути достижения цели, в том числе альтернативные, осознанно выбирать наиболее эффективные способы решения задач;
- умение соотносить свои действия с планируемыми результатами, осуществлять контроль всей деятельности в процессе достижения результата, корректировать свои действия в соответствии с изменяющейся ситуацией;
- умение оценивать правильность выполнения учебной задачи, собственные возможности ее решения;
- владение основами самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в учебной и познавательной деятельности;
- умение организовывать сотрудничество и совместную деятельность с учителем и сверстниками; работать индивидуально в группе: находить общее решение и разрешать конфликты на основе согласования позиций и учета интересов; формулировать, аргументировать и отстаивать свое мнение;

- формирование и развитие компетентности в области использования информационно-коммуникационных технологий.

Вместе с тем, вносится существенный вклад в развитие **личностных результатов**:

- формирование ответственного отношения к обучению, готовности и способности обучающихся к саморазвитию и самообразованию на основе мотивации к обучению и познанию;
- формирование коммуникативной компетентности в общении и сотрудничестве со сверстниками, взрослыми в процессе образовательной, общественно-полезной, учебно-исследовательской, творческой и других видах деятельности.

В части развития **предметных результатов** наибольшее влияние изучение курса оказывает на:

- формирование знаний, умений и навыков при решении задач информатики и программирования разных видов;
- приобретение опыта использования информационных технологий в индивидуальной и коллективной учебной и познавательной, в том числе проектной деятельности.
- формирование информационной и алгоритмической культуры; формирование представления о компьютере как универсальном устройстве обработки информации; развитие основных навыков и умений использования компьютерных устройств;
- формирование представления об основных изучаемых понятиях: информация, алгоритм, модель – и их свойства;
- развитие алгоритмического мышления, необходимого для профессиональной деятельности в современном обществе; развитие умений составить и записать алгоритм для конкретного исполнителя; формирование знаний об алгоритмических конструкциях, логических значениях и операциях; знакомство с одним из языков программирования

и основными алгоритмическими структурами – линейной, условной и циклической;

- владение системой базовых знаний, отражающих вклад информатики в формирование современной научной картины мира;
- овладение понятием сложности алгоритма, знание основных алгоритмов обработки числовой и текстовой информации, алгоритмов поиска и сортировки;
- владение универсальным языком программирования высокого уровня Python, представлениями о базовых типах данных и структурах данных; умением использовать основные управляющие конструкции;
- владение навыками и опытом разработки программ в среде программирования Python, включая тестирование и отладку программ; владение элементарными навыками формализации прикладной задачи и документирования программ;
- формирование умения работать с библиотеками программ; получение опыта использования компьютерных средств представления и анализа данных.

В результате изучения дисциплины учащиеся должны **знать**:

- основные типы алгоритмов;
- иметь представление о структуре программы, основы программирования на языках высокого уровня;
- базовые алгоритмические конструкции;
- содержание этапов разработки программы: алгоритмизация-кодирование-отладка-тестирование;
- дополнительные возможности языка Python для выражения различных алгоритмических ситуаций;
- алгоритмы и программы на языке Python решения простых, сложных и нестандартных задач в математической области;
- основы разработки простых игр в системе программирования Python.

Учащиеся должны уметь:

- записывать алгоритмические структуры на языке программирования Python;
- использовать Python для решения задач из области математики, физики;
- строить алгоритмы методом последовательного уточнения (сверху вниз), изображать эти алгоритмы в виде блок-схем;
- использовать основные алгоритмические приемы при решении математических задач;
- решать простые, сложные и нестандартные задачи;
- создавать простые игры;
- анализировать текст чужих программ, находить в них неточности, оптимизировать алгоритм, создавать собственные варианты решения.

3. СОДЕРЖАНИЕ КУРСА

Тема 1. Знакомство с языком Python

Общие сведения о языке Python. Установка Python на компьютер. Режимы работы Python. Что такое программа. Первая программа. Структура программы на языке Python. Комментарии.

Практическая работа 1.1. Установка программы Python

Практическая работа 1.2. Режимы работы с Python

Тест № 1. Знакомство с языком Python

Тема 2. Переменные и выражения

Типы данных. Преобразование типов. Переменные. Оператор присваивания. Имена переменных и ключевые слова.

Выражения. Операции. Порядок выполнения операций. Математические функции. Композиция.

Ввод и вывод. Ввод данных с клавиатуры. Вывод данных на экран.

Пример скрипта, использующего ввод и вывод данных. Задачи на элементарные действия с числами. Решение задач на элементарные действия с числами.

Практическая работа 2.1. Работа со справочной системой

Практическая работа 2.2. Переменные

Практическая работа 2.3. Выражения

Практическая работа 2.5. Задачи на элементарные действия с числами

Тест № 2. Выражения и операции.

Тема 3. Условные предложения

Логический тип данных. Логические выражения и операторы. Сложные условные выражения (логические операции and, or, not). Условный оператор. Альтернативное выполнение. Примеры решения задач с условным оператором. Множественное ветвление. Реализация ветвления в языке Python.

Практическая работа 3. 1. Логические выражения

Практическая работа 3.2. "Условный оператор" Практическая работа 3 3.

Множественное ветвление

Практическая работа 3.4. " Условные операторы"

Самостоятельная работа № 1. Решение задач по теме ” Условные операторы“
Зачетная работа № 1. ”Составление программ с ветвлением”.

Тест № 3. ” Условные операторы”.

Тема 4. Циклы

Понятие цикла. Тело цикла. Условия выполнения тела цикла. Оператор цикла с условием. Оператор цикла while. Бесконечные циклы. Альтернативная ветка цикла while. Обновление переменной. Краткая форма записи обновления.

Примеры использования циклов.

Оператор цикла с параметром for. Операторы управления циклом. Пример задачи с использованием цикла for. Вложенные циклы. Циклы в циклах.

Случайные числа. Функция randrange. Функция random. Примеры решения задач с циклом.

Практическая работа 4.1. ”Числа Фибоначчи“

Практическая работа 4.2. Решение задачи с циклом for.

Практическая работа 4.3. Реализация циклических алгоритмов

Практическая работа 4.4. Случайные числа

Практическая работа 4.5. Решение задач с циклом.

Самостоятельная работа № 2 ”Составление программ с циклом“

Тест № 4. Циклы

Творческая работа № 1. ”Циклы”

Тема 5. Функции

Создание функций. Параметры и аргументы. Локальные и глобальные переменные. Поток выполнения. Функции, возвращающие результат. Анонимные функции, инструкция lambda. Примеры решения задач с использованием функций. Рекурсивные функции. Вычисление факториала. Числа Фибоначчи.

Практическая работа 5.1. Создание функций

Практическая работа 5.2. Локальные переменные

Практическая работа 5.3. Решение задач с использованием функций

Практическая работа 5.4. Рекурсивные функции

Самостоятельная работа № 3 по теме ”Функции”

Тест № 5. Функции

Тема 6. Строки - последовательности символов

Составной тип данных - строка. Доступ по индексу. Длина строки и отрицательные индексы. Преобразование типов. Применение цикла для обхода строки. Срезы строк. Строки нельзя изменить. Сравнение строк. Оператор in. Модуль string. Операторы для всех типов последовательностей (строки, списки, кортежи). Примеры решения задач со строками.

Практическая работа 6.1. Строки

Практическая работа 6.2. Решение задач со строками.

Тема 7. Сложные типы данных

Списки. Тип список (list). Индексы. Обход списка. Проверка вхождения в список. Добавление в список. Суммирование или изменение списка. Операторы для списков. Срезы списков. Удаление списка. Клонирование списков. Списочные параметры. Функция range. Списки: примеры решения задач.

Матрицы. Вложенные списки. Матрицы. Строки и списки. Генераторы списков в Python.

Кортежи. Присваивание кортежей. Кортежи как возвращаемые значения. Введение в словари. Тип словарь (dict). Словарные операции. Словарные методы. Множества в языке Python. Множества. Множественный тип данных. Описание множеств. Операции, допустимые над множествами: объединение, пересечение, разность, включение. Оператор определения принадлежности элемента множеству.

Практическая работа 7.1. Списки.

Практическая работа 7.2. Решение задач со списками.

Тест № 7. Списки

Тема 8. Стиль программирования и отладка программ. Стиль программирования. Отладка программ. Зачет по курсу «Программирование на языке Python».

4. УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН

		Всего часов	Практика
Тема 1.	Знакомство с языком Python.	2	2
	Урок 1. Общие сведения о языке Практическая работа 1.1. Установка программы Python .		
	Урок 2. Режимы работы Практическая работа 1.2. Режимы работы с Python.		
Тема 2.	Переменные и выражения	4	4
	Урок 3. Переменные. Практическая работа 2.1. Работа со справочной системой Практическая работа 2.2. Переменные.		
	Урок 4. Выражения. Практическая работа 2.3. Выражения.		
	Урок 5. Ввод и вывод.		
	Урок 6. Задачи на элементарные действия с числами. Практическая работа 2.5. Задачи на элементарные действия с числами.		
Тема 3.	Условные предложения	5	5
	Урок 7. Логические выражения и операторы. Практическая работа 3.1. Логические выражения.		

	Урок 8. Условный оператор. Практическая работа 3.2. "Условный оператор" .		
	Урок 9. Множественное ветвление Практическая работа 3.3. Множественное ветвление.		
	Урок 10. Реализация ветвления в языке Python. Практическая работа 3.4. "Условные операторы" .		
	Урок 11. Зачетная работа № 1. "Составление программ с ветвлением" .		
Тема 4.	Циклы	7	7
	Урок 12. Оператор цикла с условием. Практическая работа 4.1. "Числа Фибоначчи" .		
	Урок 13. Оператор цикла for. Практическая работа 4.2. Решение задачи с циклом for.		
	Урок 14. Вложенные циклы Практическая работа 4.3. Реализация циклических алгоритмов.		
	Урок 15. Случайные числа. Практическая работа 4.4. Случайные числа.		
	Урок 16. Примеры решения задач с циклом. Практическая работа 4.5. Решение задач с циклом.		
	Урок 17. Творческая работа № 1. "Циклы".		
Тема 5.	Функции	5	5
	Урок 18. Создание функций. Практическая работа 5.1. Создание функций.		
	Урок 19. Локальные переменные Практическая работа 5.2. Локальные переменные.		

	Урок 20. Примеры решения задач с использованием функций. Практическая работа 5.3. Решение задач с использованием функций.		
	Урок 21. Рекурсивные функции. Практическая работа 5.4. Рекурсивные функции.		
Тема 6.	Строки - последовательности символов	3	2
	Урок 22. Строки. Практическая работа 6.1. Строки.		
	Урок 23. Срезы строк.		
	Урок 24. Примеры решения задач со строками. Практическая работа 6.2. Решение задач со строками.		
Тема 7.	Сложные типы данных	6	2
	Урок 25. Списки. Практическая работа 7.1. Списки.		
	Урок 26. Списки: примеры решения задач. Практическая работа 7.2. Решение задач со списками.		
	Урок 27. Матрицы.		
	Урок 28. Кортежи.		

	Урок 29. Введение в словари.		
	Урок 30. Множества в языке Python.		
Тема 8.	Стиль программирования и отладка программ	4	1
	Урок 31. Стиль программирования.		
	Урок 32. Отладка программ.		
	Урок 33. Зачет по курсу «Программирование на языке Python».		
	Урок 34. Что дальше?		
	Итого:	36	28

5. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА

1. Домашняя страница Python www.python.org. Справочные материалы, официальная документация.
2. Сайт проекта Интуит: Национальный открытый университет, курс «Введение в программирование на Python», <http://www.intuit.ru/studies/courses/12179/1172/info>.
3. Сайт проекта Интуит: Национальный открытый университет. Курс «Язык программирования Python» <http://www.intuit.ru/studies/courses/49/49/info>.
4. Сайт проекта Open Book Project openbookproject.net содержит серию практических примеров на Python Криса Мейерса.
5. Python. Подробный справочник Дэвида М. Бизли — книга со справочной информацией о языке Python и модулях стандартной библиотеки.
6. Python. Справочник Марка Лутца. Справочник по наиболее часто используемым функциям и модулям.