

Комитет по образованию администрации
муниципального округа город Славгород Алтайского края
Муниципальное бюджетное общеобразовательное учреждение
«Славгородская средняя общеобразовательная школа»

ПРИНЯТА
на заседании
педагогического совета
Протокол №1
от «29» августа 2024 г



**Дополнительная общеобразовательная общеразвивающая программа
технической направленности
«Программирование на языке Python»**

Возраст учащихся: 13–16 лет
Срок реализации программы: 1 год

Автор-составитель:
Череш Александр Михайлович,
учитель информатики

1. Комплекс основных характеристик дополнительной общеобразовательной общеразвивающей программы

1.1. Пояснительная записка

Нормативные правовые основы разработки ДООП:

- Федеральный закон «Об образовании в Российской Федерации» N 273-ФЗ от 29.12.2012;
- Концепция развития дополнительного образования детей до 2030г. (Распоряжение Правительства РФ от 31.03.2022 №678-р);
- Приказ Минпросвещения России от 27.07.2022 № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам;
- Методические рекомендации по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы) (Приложение к письму Департамента государственной политики в сфере воспитания детей и молодежи Министерства образования и науки РФ от 18.11.2015 № 09-3242);
- Письмо Министерства образования и науки Российской Федерации 09-3242 от 18.11.2015 г. О направлении информации «Методические рекомендации по проектированию дополнительных общеразвивающих программ (включая разноуровневые);
- Постановление Главного государственного санитарного врача Российской Федерации от 28.09.2020 № 28. Санитарные правила СП 2.4.3648-20 «Санитарноэпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи»;
- Устав МБОУ "Славгородская СОШ;
- Положение о порядке разработки, оформления и утверждения дополнительной общеобразовательной общеразвивающей программы МБОУ «Славгородская СОШ».

Актуальность

В настоящее время прослеживается острый кризис квалифицированных кадров в различных сферах рынка труда. Работодатели, хоть и обращают внимание на область предметных знаний, но все же отдают предпочтения универсальным навыкам. Подобная позиция связана с постоянно изменяющимися условиями труда, технологическим прогрессом. Подобные универсальные навыки (способность работать в команде, многозадачность, творческий подход, критическое мышление, продуктивное мышление, нацеленность на результат) вырабатываются в процессе профессионального становления.

Развитие навыков программирования является уникальным способом формирования интереса учащихся к техническим направлениям деятельности, а также совершенствования алгоритмического, креативного мышления и навыков проектной деятельности.

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.

Обучение включает в себя следующие основные предметы:

Обучение включает в себя следующие основные предметы:

- информатика

Вид ДООП: Программа «Программирование на языке Python» является модифицированной, составлена на основе учебного пособия: Златопольского Д. М. «Основы программирования на языке Python». Оно способствует обретению первичных навыков программирования. Осознаю того, что процесс программирования не является недостижимым и абсолютно сложным для большинства детей.

Направленность ДООП:

Естественнонаучная.

Адресат ДООП:

Программа ориентирована на дополнительное образование обучающихся 13-16 лет, интересующихся программированием.

Программа рассчитана на 1 год обучения - 34 часа.

Основной формой обучения являются групповые занятия. В основе образовательного процесса лежит проектный подход. Набор детей свободный, без предъявления требований к уровню подготовленности. Количество детей в группе – 12-15 человек.

Срок и объем освоения ДООП:

1 год, 34 часа, из них:

- «Стартовый уровень» - 1 год, 34 часа;

Форма обучения:

Очная.

Особенности организации образовательной деятельности:

Разновозрастная группа.

Режим занятий:

Таблица 1.1.1

Режим занятий

Предмет	Стартовый уровень
Информатика	1 часа в неделю; 34 часа в год.

1.2. Цель, задачи, ожидаемые результаты**Цель программы**

Целью курса – способствовать формированию алгоритмического и критического мышления, навыков проектной деятельности через использование языка программирования Python как инструмента для реализации идей в области своих интересов.

Задачи программы

Достижение поставленной цели складывается из выполнения следующих задач:

Обучающие:

1. Познакомить с базовыми понятиями основ программирования.
2. Познакомить с особенностями синтаксиса языка программирования Python.
3. Познакомить с принципами объектно-ориентированного программирования.
4. Сформировать навык правильного оформления кода.

5. Сформировать навык использования итеративного подхода при решении различных задач.
6. Формировать навык решения задач и разработки проектов с помощью Python.
7. Формировать навык разработки игр с помощью языка программирования Python.
8. Формировать навык тестирования и оптимизации созданных проектов.

Развивающие:

1. Формировать и развивать логическое, алгоритмическое, критическое, креативное и пространственное мышление.
2. Формировать навык публичного выступления и презентации.
3. Формировать навык планирования деятельности и основ тайм-менеджмента.
4. Формировать и расширять словарный запас на английском языке.
5. Формировать функциональную грамотность.
6. Расширять кругозор, развивать память, внимание, творческое воображение, абстрактно-логические и наглядно-образные виды мышления и типы памяти, основные мыслительные операции и свойства внимания.
7. Совершенствовать диалогическую речь учащихся: уметь слушать собеседника, понимать вопросы, смысл знаний, уметь задавать вопросы, отвечать на них.

Воспитательные:

1. Воспитывать у учащихся потребность в сотрудничестве, взаимодействии со сверстниками, умение подчинять свои интересы определенным правилам, развивать эмоциональный интеллект.
2. Формировать умение давать качественную обратную связь и реагировать на нее.
3. Формировать информационную культуру.

В результате освоения дисциплины обучающийся должен:

уметь:

- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- настраивать рабочую среду Python;

знать:

- этапы решения задачи на компьютере;
- типы данных;
- базовые конструкции изучаемых языков программирования;
- принципы структурного и модульного программирования;
- принципы объектно-ориентированного программирования;
- основы создания приложений в Python;
- инструментальные средства Python.

Ожидаемые результаты:

Таблица 1.2.1

Ожидаемые результаты

	Стартовый уровень
Личностные результаты	<ul style="list-style-type: none"> • умение работать в команде: работа в общем ритме, эффективное распределение задач и др. • формирование высокого познавательного интереса учащихся

	<ul style="list-style-type: none"> • формирование критического мышления • проявление технического мышления, познавательной деятельности, творческой инициативы, самостоятельности
Метапредметные результаты	<ul style="list-style-type: none"> • умение ориентироваться в информационном пространстве, продуктивно использовать техническую литературу для поиска сложных решений • способность творчески решать технические задачи • готовность и способность применения теоретических знаний по физике, информатике для решения задач в реальном мире • способность правильно организовывать рабочее место и время для достижения поставленных целей • Знание основ ТРИЗ, навыки публичного выступления и презентации результатов, навык генерации идей
Предметные результаты	<ul style="list-style-type: none"> • знание основ и принципов программирования • знание и понимание основных алгоритмических конструкций • знание основ и овладение практическими базисными знаниями программирования в Python • знание основ и овладение практическими базисными навыками разработки игр в PyGame

1.3. Содержание программы
«Программирование на языке Python»
Стартовый уровень (1 год обучения)
Учебный план

Таблица 1.3.1

Учебный план

№ п/п	Наименование раздела, темы	Количество часов			Формы аттестации/контроля
		Всего	Теория	Практика	
1.	Раздел 1. Введение в Python	2	2	0	Тест по модулю
	1.1. Инструкции и структура программы.	1	1	0	Устный опрос
	1.2. Установка Python. Ввод и вывод данных.	1	1	0	Решение задач
2.	Раздел 2. Типы данных и операции	8	4	4	Тест по модулю
	2.1. Переменные. Операторы.	2	1	1	Тест по теме
	2.2. Типы данных. Преобразование типов. Различия типов данных. Базовые функции (abs, round, int, math)	6	3	3	Решение задач
3.	Раздел 3. Инструкции и синтаксис	8	4	4	Тест по модулю
	3.1. Условный оператор if, Альтернативный условный оператор elif, else.	2	1	1	Решение задач
	3.2. Циклы while. Цикл for. Обработка включений. Функция random. Случайные числа.	6	3	3	Решение задач с использованием циклов
4.	Раздел 4. Функции и модули в программировании	9	5	4	Тест по модулю
	4.1. Встроенные и пользовательские функции. Создание функций. Функции, возвращающие результат.	3	2	1	Тест по теме
	4.2. Строки, последовательность символов. Доступ по индексу. Длина строки и отрицательные индексы. Преобразование типов. Применение цикла для обхода строки. Срезы строк. Сравнение строк.	6	3	3	Решение задач с использованием строк
	Раздел 5. Написание и отладка программ. Самостоятельная работа	7	1	6	Защита проекта

	Итого	34	16	18	
--	--------------	-----------	-----------	-----------	--

Содержание учебного плана

Раздел 1. Введение в Python

1.1. Инструкции и структура программы.

Техника безопасности. История создания. Области применения и перспективы.
Инструкции и структура программы. Основы Алгоритмизации. Достоинства языка.

1.2. Установка Python. Ввод и вывод данных.

Установка Python. Доступ к документации. Основы ввода и вывода данных. Первая программа на языке Python. Основы Алгоритмизации.

Практическая работа: Решение задач для развития логики и понимания работы алгоритма. Установка программы Python. Написание первой программы вывод на экран.

Раздел 2. Типы данных и операции

а. Переменные. Операторы.

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.

Практическая работа: Решение задач на элементарные действия с числами. Создание программы простейший калькулятор.

б. Типы данных. Базовые функции для работы с различными типами данных.

Типы данных. Преобразование типов. Различия типов данных. Базовые функции (abs, round, int, math)

Практическая работа: Решение математических задач с использованием функции import math.

Раздел 3. Инструкции и синтаксис

3.1. Условные операторы if, elif, else.

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: Практическое закрепление знаний по условным операторам. Создание программ.

3.2. Циклы while, for. Обработка исключений. Случайные числа.

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: Числа Фибоначчи. Решение задачи с циклом for. Создание игры угадай число. Повторение пройденного.

Раздел 4. Функции и модули в программировании

4.1. Встроенные и пользовательские 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.

Практическая работа: Создание игры русская рулетка. Создание всех ранее созданных программ с использованием функций.

4.2. 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: Написание программ. Работа со строками.

Раздел 5. Написание и отладка программ. Самостоятельная работа

Практическая работа: Стиль программирования. Отладка программ. Проектная работа по курсу «Программирование на языке Python».

Форма организации деятельности: групповая, индивидуальная.

2. Комплекс организационно - педагогических условий

2.1. Календарный учебный график

Таблица 2.1.1

Календарный учебный график

Период	Сроки
Начало учебного года	16.09
Окончание учебного года	25.05
Продолжительность каникул	26.05.2024-31.08.2025
Количество учебных недель	34
Количество учебных дней	34

**Календарно-тематическое планирование
1 год обучения (Стартовый уровень)**

№ п/п	Название раздела, темы	Дата проведения
1.	Техника безопасности. Инструкции и структура программы. Основы Алгоритмизации. Достоинства языка.	
2.	Установка Python. Доступ к документации. Основы ввода и вывода данных.	
3.	Переменные. Оператор присваивания. Имена переменных и ключевые слова. Выражения. Операции.	
4.	Решение задач на элементарные действия с числами. Создание программы простейший калькулятор.	
5.	Типы данных. Преобразование типов.	
6.	Различия типов данных.	
7.	Базовые функции (abs,round, int, math)	
8.	Решение математических задач с использованием функции import math.	
9.	Решение математических задач с использованием функции import math.	
10.	Решение математических задач с использованием функции import math.	
11.	Условный оператор. Альтернативное выполнение. Примеры решения задач с условным оператором.	
12.	Практическое закрепление знаний по условным операторам. Создание программ.	
13.	Понятие цикла. Тело цикла. Условия выполнения тела цикла.	
14.	Оператор цикла с условием. Оператор цикла while.	
15.	Бесконечные циклы. Альтернативная ветка цикла while. Обновление переменной. Краткая форма записи обновления.	
16.	Вложенные циклы. Циклы в циклах. Случайные числа. Функция randrange. Функция random.	
17.	Числа Фибоначчи. Решение задачи с циклом for.	
18.	Создание игры угадай число.	
19.	Создание функций. Параметры и аргументы. Локальные и глобальные переменные. Поток выполнения. Функции, возвращающие результат.	
20.	Анонимные функции, инструкция lambda. Примеры решения задач с использованием функций. Рекурсивные функции.	
21.	Создание игры русская рулетка. Создание всех ранее созданных программ с использованием функций.	
22.	Составной тип данных - строка. Доступ по индексу. Длина строки и отрицательные индексы. Преобразование типов.	
23.	Применение цикла для обхода строки. Срезы строк. Строки нельзя изменить. Сравнение строк.	
24.	Оператор in. Модуль string. Операторы для всех типов последовательностей (строки, списки, кортежи).	
25.	Написание программ. Работа со строками.	
26.	Написание программ. Работа со строками.	
27.	Написание программ. Работа со строками.	
28.	Стиль программирования. Отладка программ.	
29.	Проектная работа по курсу «Программирование на языке	

	Python».	
30.	Проектная работа по курсу «Программирование на языке Python».	
31.	Проектная работа по курсу «Программирование на языке Python».	
32.	Проектная работа по курсу «Программирование на языке Python».	
33.	Проектная работа по курсу «Программирование на языке Python».	
34.	Проектная работа по курсу «Программирование на языке Python».	

2.2. Условия реализации программы

Таблица 2.2.1

Условия реализации программы

Аспекты	Характеристика (заполнить)
Материально-техническое обеспечение	<ul style="list-style-type: none"> • моноблок – 15 шт. • маршрутизатор – 1 шт. • проектор–1 шт.
Информационное обеспечение	<ul style="list-style-type: none"> • ОС AstraLinux • Python • Pyzo • Pygame • Текстовый редактор • KompoZer • Редактор презентаций
Кадровое обеспечение	<ul style="list-style-type: none"> • учитель информатики, педагог дополнительного образования, не ниже среднего профессионального образования, с опытом работы 1 год.

2.3. Формы аттестации

Формами аттестации являются:

- творческая работа
- практическая работа
- проектная работа

2.4 Методические материалы

Методическое обеспечение программы включает приёмы и методы организации образовательного процесса, дидактические материалы, техническое оснащение занятий.

Виды дидактических материалов, используемые при реализации программы:

Для обеспечения наглядности и доступности изучаемого материала педагог использует наглядные пособия следующих видов:

- схематические или символические (оформленные стенды и планшеты, таблицы, схемы, рисунки, графики, плакаты, диаграммы, чертежи, шаблоны и т.п.);

- картинные (иллюстрации, слайды, фотоматериалы и др.);
- звуковые (аудиозаписи);
- смешанные (видеозаписи, учебные кинофильмы и т.д.);
- дидактические пособия (карточки, рабочие тетради, раздаточный материал, вопросы и задания для опроса, тесты, практические задания, упражнения и др.).
- компьютерные программы в электронном виде (компьютеры с программами, CD, флеш-носители);
 - учебные пособия, журналы, книги, Интернет-ресурсы.

При реализации программы с целью повышения качества и эффективности процесса обучения применяются современные эффективные технологии обучения, ориентированные не на накопление знаний, а на организацию активной деятельности обучающихся:

- технологии проектной деятельности;
- компьютерные (информационные) технологии;
- технологии учебно-игровой деятельности (моделирование);
- технологии коммуникативно-диалоговой деятельности;
- модульные технологии;
- квест-технологии;
- технологии личностно-ориентированного обучения;
- кейс-технологии.

Информационные технологии используются в различных видах деятельности:

- при подготовке и проведении занятий;
- для создания авторских мультимедийных презентаций;
- в рамках индивидуальной и групповой проектной деятельности;
- для самостоятельной работы;
- для накопления демонстрационных материалов к занятиям (видеоматериалы, таблицы, презентации, карты);

Одним из основных методов является метод проектного обучения, так как он является неотъемлемой частью учебного процесса. Исходный лозунг основателей системы проектного обучения – «Все из жизни, все для жизни». Обучение строится на активной основе, через практическую деятельность ученика, ориентируясь на его личный интерес и практическую востребованность полученных знаний в дальнейшей жизни, обучающийся имеет возможность через проектную деятельность освоить получаемые знания. Проекты представляются в виде готовых программ, презентаций проектов, научных докладов, моделей, демонстрации видеофильма. Достоинствами проектной деятельности являются:

- Уметь работать в коллективе;
- Брать ответственность за выбор решения на себя;
- Разделять ответственность с другими;
- Предоставлять ребенку свободу выбора темы, методов работы;
 - Понимание каждым обучающимся важности работы и др.

2.5 Список литературы

Список литературы, используемой педагогом

Основная:

1. Клейнберг Дж. Алгоритмы: разработка и применение. СПб: Питер, 2016. - 800 с.
2. Златопольский Д. М. Основы программирования на языке Python. – М.: ДМК Пресс, 2017. – 284 с.: ил.
3. Бхаргава А. Грокаем алгоритмы. Иллюстрированное пособие для программистов и любопытствующих. - СПб.: Питер, 2017. - 288 с.

Дополнительная:

1. Луц М. Изучаем Python. СПб: Симво-плюс, 2011. - 1280 с.
2. Паронджанов В.Д. Учись писать, читать и понимать алгоритмы. М: Ямб, 2012. – 520 с.
4. Златопольский Д.М. Сборник задач по программированию. СПб: БХВ-Петербург, 2011. - 295 с.
5. ООП на Python: концепции, принципы и примеры реализации [Электронный ресурс]. – Режим доступа: <https://proglib.io/p/python-oop/>

Список рекомендуемой литературы для обучающихся

Основная:

1. Свейгар Эл. Учим python, делая крутые игры М: Эксмо, 2018. – 416 с.
2. Семакин И.Г. Основы алгоритмизации и программирования. М.: Академия, 2016. - 304 с.

Дополнительная:

6. Луц М. Изучаем Python. СПб: Симво-плюс, 2011. - 1280 с.
7. Уроки по Python для начинающих [Электронный ресурс]. – Режим доступа: <https://pythonru.com/uroki/vvedenie-uroki-po-python-dlja-nachinajushhih>
Алгоритмизация. Программирования Python 3 [Электронный ресурс]. – Режим доступа: <https://stepik.org/course/7215/promo>