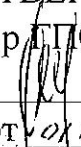


Государственное профессиональное образовательное учреждение Тульской области «Болоховский машиностроительный техникум»

Принято
педагогическим
советом Протокол № 12
от 30.08.2022

УТВЕРЖДАЮ
Директор ГПОУ ТО «БМТ»
 Косинова Е.А.
Приказ от 01.09.2022 № 64/02



**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА
технической направленности
«ПРОГРАММИРОВАНИЕ НА PYTHON»**

Возраст обучающихся: 15-17 лет
Срок реализации: 1 год

Автор: Петраков А.А.
педагог дополнительного
образования ГПОУ ТО «БМТ»

Болохово
2022

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Python – язык программирования общего назначения, оптимизированный для создания качественного программного обеспечения. Язык Python один из самых используемых языков программирования в мире в таких областях, как создание веб-сценариев, системное программирование, создание пользовательских интерфейсов, настройка программных продуктов под пользователя, численное программирование и др.

Дополнительная общеобразовательная общеразвивающая программа «Мастерская программирования на языке Python» (далее – Программа) технической направленности ознакомительного уровня, специально адаптирована для детей с ОВЗ и детей-инвалидов и нацелена на то, чтобы каждый обучающийся мог эффективно использовать современные компьютерные технологии в учебной, творческой, самостоятельной и досуговой деятельности. Программа способствует развитию познавательных интересов и творческих способностей обучающихся, удовлетворению их потребностей в интеллектуальном, нравственном, физическом совершенствовании, она имеет практическую направленность по развитию ИТ-компетентности.

Актуальность программы обусловлена быстрым внедрением компьютерной техники в повседневную жизнь, переходом к новым технологиям обработки информации. Изучая программирование, обучающиеся лучше понимают возможности и границы применения компьютеров. К ним приходит осознание того, что компьютер является инструментом, управляемым людьми. Не все обучающиеся станут профессиональными программистами, но все выиграют от того, что постигли природу программирования и научились создавать собственные программы.

Новизна программы заключается в построении индивидуальной образовательной траектории обучающегося, в приобретении им знаний, востребованных на рынке труда, в повышении самооценки и осознании перспектив будущей жизни, дальнейшей социализации.

Педагогическая целесообразность программы заключается в том, что занятие программированием, даёт необычайно сильный толчок для развития интеллекта обучающихся. формирует их логическое мышление, вырабатывает привычку аккуратной и систематической работы.

Отличительная особенность программы состоит в том, что она является мощным образовательным инструментом, позволяющим дать обучающимся навыки по программированию на языке Python.

Реализация Программы, основана на деятельностном подходе, более 60% времени отводится практической деятельности, способствующей развитию активной познавательной деятельности, творчества и достижению высоких результатов в области информационно-коммуникационных технологий.

Цель программы ознакомить обучающихся с возможностями, синтаксисом, технологией языка Python и обучить методами программирования для решения прикладных математических и информационных задач.

Реализация поставленной цели предусматривает решение ряда задач.

Задачи Программы

Обучающие:

- сформировать представление об основах программирования в среде Python;
- сформировать навыки грамотной работы в системе программирования Python;
- ознакомить с базовыми понятиями теории алгоритмов при решении математических задач;
- обучить методам решения задач, реализуемым на языке Python;
- сформировать практические навыки решения прикладных задач;
- сформировать навыки поиска информации, работы с технической литературой.

Развивающие:

- развить навыки самостоятельного и творческого подхода к решению задач с помощью средств современной вычислительной техники;
- развить алгоритмическое и логическое мышление учащихся;
- развить творческие способности обучающихся, их потребность в самореализации;
- развить интеллектуальные и практические умения,
- самостоятельно приобретать и применять на практике полученные знания;

Воспитательные:

- содействовать воспитанию устойчивого интереса к изучению программирования;
- содействовать воспитанию информационной культуры;
- формировать потребность в творческой деятельности, стремление к самовыражению через техническое творчество;
- содействовать воспитанию интереса профессиям, связанным с программированием.

Категория обучающихся

Обучение в разновозрастных группах, которые комплектуются из обучающихся 13-18 лет, имеющих начальные умения и навыки работы в текстовых редакторах, в Skype и Google, с почтовой программой, с браузером, использовании встроенной панели рисования для отображения блок-схем.

Рекомендуемое количество обучающихся в группе – 12 человек.

Сроки реализации

Программа рассчитана на один года обучения. Общее количество часов в год составляет 72 часов.

Формы и режим занятий

Программа реализуется через занятия 1 раз в неделю по 2 часа. Программа включает в себя теоретические и практические занятия.

Формы организации деятельности – групповая.

Планируемые результаты освоения Программы

По итогам обучения обучающиеся будут знать:

- технику безопасности и требования, предъявляемые к организации рабочего места
- термины: «данные», «кодирование», «алгоритм», «программа», понимание различий между употреблением этих терминов в обыденной речи и в информатике;
- методы представления и алгоритмы обработки данных, программную реализацию алгоритмов;
- основы алгоритмической культуры, нормы информационной этики;
- математические и компьютерные модели, их использование;
- виды программного обеспечения, решаемые с его помощью задачи;
- вредоносное программное обеспечение и средства защиты от него;

По итогам обучения обучающиеся будут

уметь:

- составлять несложные программы;
- работать с основными видами программных систем и

определять понятия, создавать обобщения, устанавливать аналогии, классифицировать, самостоятельно выбирая основания и критерии для классификации;

- устанавливать причинно-следственные связи, делать логическое умозаключение (индуктивное, дедуктивное и по аналогии) и делать выводы;
- создавать, применять и преобразовывать знаки и символы, модели и схемы для решения учебных и познавательных задач;
- составлять алгоритмы управления исполнителями и записывать их на выбранном алгоритмическом языке;
- использовать логические значения, операции и выражения с ними;
- выполнять алгоритмы, описанные с использованием конструкций ветвления (условные операторы) и повторения (циклы), вспомогательных алгоритмов, простых и табличных величин;
- записывать алгоритмические структуры на языке программирования
- Python;
- решать простые, сложные и нестандартные задачи.

СОДЕРЖАНИЕ ПРОГРАММЫ

Учебно-тематический план

№	Названия раздела/темы	Количество часов			Формы аттестации и контроля
		Всего	Теория	Практика	
1.	Введение в язык программирования Python	5	2	4	
1.1.	Вводное занятие. Понятия «алгоритм» и «программа». Начальное знакомство с языком. Инструктаж	2	1	1	Тест.
1.2.	Элементы языка. Структура программы. Операции и переменные. Типы данных	2	1	1	
1.3.	Ввод и вывод данных на языке Python. Ввод и отладка программ в среде	1	-	1	Текущий контроль. Практикум
2.	Линейные алгоритмы и их реализация на языке Python	9	2	7	
2.1.	Линейные алгоритмы целочисленных данных и их реализация на Python	2	1	1	
2.2.	Решение задач по теме «Обработка целочисленных данных»	1	-	1	Текущий контроль. Практикум
2.3.	Линейные алгоритмы вещественных чисел и их реализация на Python	2	1	1	
2.4.	Решение задач по теме «Обработка вещественных чисел»	2	-	2	Текущий контроль. Практикум
2.5.	Самостоятельная работа по темам раздела «Линейные алгоритмы и их реализация в среде Python»	2	-	2	Текущий контроль. Самостоятельная работа
3.	Разветвляющиеся алгоритмы и их реализация на языке Python	14	4	10	

3.1.	Алгоритм «выбор», графическое изображение, полное и неполное ветвление	2	1	1	
3.2.	Реализация алгоритма «выбор» на Python. Примеры решения задач	2	1	1	
3.3.	Решение задач по теме «Полное и неполное ветвление»	2	-	2	Текущий контроль. Практикум
3.4.	Разветвляющиеся алгоритмы. Сложные условия. Каскадные ветвления	2	1	1	
3.5.	Множественный выбор и его реализация с помощью вложенных ветвлений	2	1	1	
3.6.	Решение задач по теме «Сложные условия. Каскадные ветвления»	2	-	2	Текущий контроль. Практикум
3.7.	Контрольная работа по темам раздела: «Разветвляющиеся алгоритмы и их реализация на языке Python»	2	-	2	Промежуточный контроль. Контрольная работа
4.	Циклические алгоритмы и их реализация на Python	26	4	22	
4.1.	Циклические алгоритмы. Цикл с предусловием. Инструкции управления циклом	2	1	1	
4.2.	Решение задач по теме «Циклические алгоритмы. Цикл с предусловием»	6	-	6	Текущий контроль. Практикум
4.3.	Циклические алгоритмы. Цикл с постусловием. Инструкции управления	2	1	1	
4.4.	Решение задач по теме «Циклические алгоритмы. Цикл с постусловием»	2	-	2	Текущий контроль. Практикум
4.5.	Самостоятельная работа по теме «Циклические алгоритмы. Цикл с предусловием»	2	-	2	Текущий контроль. Самостоятельная работа

4.6.	Циклические алгоритмы. Цикл с параметром. Инструкции управления циклом	2	1	1	
4.7.	Решение задач по теме «Циклические алгоритмы. Цикл с параметром»	4	-	4	Текущий контроль. Практикум
4.8.	Самостоятельная работа по теме «Циклические алгоритмы. Циклы со счетчиком»	2	-	2	Текущий контроль. Самостоятельная работа
4.9.	Сложные циклические алгоритмы. Вложенные циклы	2	1	1	
4.10.	Самостоятельная работа по разделу: «Циклические алгоритмы и их реализация на Python». Анализ самостоятельной работы	2	-	2	Текущий контроль. Самостоятельная работа
5.	Этапы решения задач на языке Python	6	1	5	
5.1.	Последовательное конструирование алгоритма	2	1	1	
5.2.	Этапы решения задачи на компьютере. Последовательное конструирование алгоритма	4	-	4	Текущий контроль. Практикум
6.	Вспомогательные алгоритмы и их реализация на языке Python	10	2	8	
6.1.	Вспомогательные алгоритмы и подпрограммы. Функции	2	1	1	
6.2.	Вспомогательные алгоритмы и подпрограммы. Решение задач с применением функций	4	-	4	Текущий контроль. Практикум
6.3.	Вспомогательные алгоритмы и подпрограммы. Рекурсия	2	1	1	
6.4.	Вспомогательные алгоритмы и подпрограммы. Решение задач с применением рекурсии	2	-	2	Текущий контроль Практикум
7.	Итоги обучения	4	1	3	

7.1	Итоговая самостоятельная работа по теме «Реализация основных типов алгоритмов»	2	-	2	
7.2	Подведение итогов обучающихся	2	1	1	
	Итого	72	16	56	

СОДЕРЖАНИЕ УЧЕБНО-ТЕМАТИЧЕСКОГО ПЛАНА

Раздел 1. Введение в язык программирования Python

Тема 1.1. Вводное занятие. Понятия «алгоритм» и «программа». Начальное знакомство с языком. Инструктаж

Теория. Структура образовательной программы, её цель и задачи, содержание обучения. Основные правила и требования техники безопасности при работе за компьютером. Понятие алгоритма и программы.

Практика. Первичная диагностика. Тестирование. Установка, работа в среде программирования Python.

Тема 1.2. Элементы языка. Структура программы. Операции и переменные. Типы данных

Теория. Знакомство с основными типами переменных, синтаксисом языка программирования. основными процедурами ввода исходных данных и вывода результатов

Практика. Отработка процедур ввода исходных данных и вывода результатов. Тестирование.

Тема 1.3. Ввод и вывод данных на языке Python. Ввод и отладка программ в среде

Практика. Практикум по созданию элементарных программ ввода-вывода данных, работа со средой, отладка программ. Анализ возможных синтаксических ошибок.

Раздел 2. Линейные алгоритмы и их реализация на языке Python

Тема 2.1. Линейные алгоритмы целочисленных данных и их реализация на языке Python

Теория. Знакомство со структурой линейного алгоритма, правилами записи арифметических выражений. Различные типы данных, допустимые операции над ними и ресурсы оперативной памяти. Выполнение операции присваивания в оперативной памяти (далее – ОП) компьютера.

Практика. Создание блок-схемы линейного алгоритма. Запись арифметических выражений на языке программирования.

Тема 2.2. Решение задач по теме «Обработка целочисленных данных»

Практика. Практикум по разработке линейных алгоритмов. Ввод и отладка программ, реализующих линейный алгоритм обработки целых чисел. Анализ готовых линейных программ.

Тема 2.3. Линейные алгоритмы вещественных чисел и их реализация на Python

Теория. Различные типы данных, допустимые операции над вещественными числами и ресурсы оперативной памяти.

Практика. Разбор типичных задач с линейной структурой алгоритма.

Тема 2.4. Решение задач по теме «Обработка вещественных чисел»

Практика. Практикум по реализации линейных алгоритмов вещественных чисел. Ввод и отладка программ, реализующих линейный алгоритм обработки вещественных чисел.

Тема 2.5. Самостоятельная работа по темам раздела: «Линейные алгоритмы и их реализация в среде Python»

Практика. Самостоятельная работа по составлению линейного алгоритма, написанию программы, вводу и отладке программного кода, анализу результатов.

Раздел 3. Разветвляющиеся алгоритмы и их реализация на языке Python

Тема 3.1. Алгоритм «выбор», графическое изображение, полное и неполное ветвление

Теория. Понятие алгоритма «выбор», графическое изображение.

Практика. Составление алгоритма «полное ветвление».

Тема 3.2. Реализация алгоритма «выбор» на Python. Примеры решения задач

Теория. Типовые задачи, использующие алгоритм «выбор», изображение алгоритма в виде блок-схемы.

Практика. Ввод и отладка программ в среде Python.

Тема 3.3. Решение задач по теме «Полное и неполное ветвление»

Практика. Практикум по реализации алгоритмов «полное ветвление» и «неполное ветвление». Ввод и отладка программ в среде Python.

Тема 3.4. Разветвляющиеся алгоритмы. Сложные условия. Каскадные ветвления

Теория. Составление и анализ алгоритмов.

Практика. Ввод и отладка программ в среде Python.

Тема 3.5. Множественный выбор и его реализация с помощью вложенных ветвлений

Теория. Понятие множественного выбора, изображение на блок-схеме.

Практика. Ввод и отладка программ в Python.

Тема 3.6. Решение задач по теме «Сложные условия. Каскадные ветвления»

Практика. Практикум по составлению алгоритмов для решения задач со сложными условиями и каскадными ветвлениями. Ввод и отладка программ в среде Python.

Тема 3.7. Контрольная работа по темам раздела «Разветвляющиеся алгоритмы и их реализация на языке Python»

Практика. Контрольная работа по составлению алгоритмов, написанию программного кода, вводу и отладке программ в среде Python. Анализ работы

Раздел 4. Циклические алгоритмы и их реализация на языке Python

Тема 4.1. Циклические алгоритмы. Цикл с предусловием.

Инструкции управления циклом

Теория. Понятие цикла с предусловием, графическое изображение.

Практика. Составление алгоритмов, написание программного кода, ввод и отладка программ в среде Python.

Тема 4.2. Решение задач по теме «Циклические алгоритмы. Цикл с предусловием»

Практика. Практикум по составлению блок-схем, написанию программного кода, вводу и отладке программ, реализующих цикл с предусловием в среде Python.

Тема 4.3. Циклические алгоритмы. Цикл с постусловием.

Инструкции управления циклом

Теория. Понятие цикла с постусловием, графическое изображение.

Практика. Ввод и отладка простейших программ, реализующих цикл с постусловием в среде Python.

Тема 4.4. Решение задач по теме «Циклические алгоритмы. Цикл с постусловием»

Практика. Практикум по решению типовых задач с помощью цикла с постусловием. Составление блок-схем, написание программного кода, ввод и отладка программ, реализующих цикл с постусловием в Python.

Тема 4.5. Самостоятельная работа по теме «Циклические алгоритмы. Цикл с предусловием»

Практика. Самостоятельная работа по решению типовых задач с помощью цикла с предусловием. Ввод и отладка программ, реализующих цикл с предусловием в среде Python. Анализ составленного алгоритма.

Тема 4.6. Циклические алгоритмы. Цикл с параметром.
Инструкции управления циклом

Теория. Понятие цикла с параметром, графическое изображение.

Практика. Составление алгоритмов, написание программного кода, ввод и отладка программ в среде Python.

Тема 4.7. Решение задач по теме: «Циклические алгоритмы. Цикл с параметром»

Практика. Практикум по разработке алгоритмов цикла с параметром, анализу логических ошибок. Ввод и отладка программ, реализующих цикл с параметром в среде Python.

Тема 4.8. Самостоятельная работа по теме «Циклические алгоритмы. Циклы со счетчиком».

Практика. Самостоятельная работа по разработке блок-схем и отладке программ, реализующих циклы с параметром в среде Python. Анализ самостоятельной работы.

Тема 4.9. Сложные циклические алгоритмы. Вложенные циклы.
Решение задач

Теория. Понятие вложенного цикла, графическое изображение.

Практика. Решение задач по разработке и отладке программ, реализующих вложенные циклы в среде Python. Анализ разработанного алгоритма. Оценка эффективности разработанного алгоритма.

Тема 4.10. Самостоятельная работа по темам раздела «Циклические алгоритмы и их реализация на языке Python». Анализ самостоятельной работы

Практика. Самостоятельная работа по разработке блок-схем и отладке программ, реализующих циклические алгоритмы в среде Python.

Раздел 5. Этапы решения задач на языке Python

Тема 5.1. Последовательное конструирование алгоритма

Теория. Этапы и особенности решения задачи на компьютере. Этап создания алгоритма. Использование принципа последовательного конструирования алгоритма. Будет также рассмотрен принцип его работы.

Практика. Разработка алгоритмов. Написание программного кода, ввод и отладка программ в среде Python.

Тема 5.2. Этапы решения задачи на компьютере. Последовательное конструирование алгоритма

Практика. Практикум по последовательному конструированию алгоритмов. Написание программного кода, ввод и отладка программ в среде Python.

Раздел 6. Вспомогательные алгоритмы и их реализация на языке Python

Тема 6.1. Вспомогательные алгоритмы и подпрограммы. Функции

Теория. Понятие вспомогательного алгоритма. Формат записи вспомогательного алгоритма в виде функции. Типовые задачи.

Практика. Ввод и отладка программ с использованием функции в среде Python.

Тема 6.2. Вспомогательные алгоритмы и подпрограммы. Решение задач с применением функций

Практика. Практикум по программированию решения отдельных подзадач с помощью отдельных функций, которые потом при необходимости вызываются в различных местах программы. Польза функций при решении задач. Важные принципы в программировании – модульность и повторное использование кода. Их польза при разработке сложных программ.

Тема 6.3. Вспомогательные алгоритмы и подпрограммы. Рекурсия

Теория. Понятие рекурсии, в чём её сложность. Рекурсивные алгоритмы

и их особенности: когда нужно использовать рекурсию в программировании, а когда лучше обойтись без этого.

Практика. Ввод и отладка программ с использованием рекурсии в среде Python.

Тема 6.4. вспомогательные алгоритмы и подпрограммы. Решение задач с применением рекурсии

Практика. Практикум по решению типовых задач с вводом и отладкой программ с использованием рекурсии в среде Python.

Раздел 7. Итоги обучения

Тема 7.1. Итоговая самостоятельная работа по теме «Реализация основных типов алгоритмов»

Практика. Зачётная работа по составлению алгоритма, написанию программы, вводу и отладке программного кода.

Тема 7.2. Подведение итогов обучения

Теория. Обзор пройденного материала.

Практика. Анализ результатов работы.

ФОРМЫ КОНТРОЛЯ И ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

Результативность обучения обеспечивается применением различных форм, методов и приемов, которые тесно связаны между собой и дополняют друг друга. Большая часть занятий отводится практической работе, в ходе практической деятельности педагог тактично контролирует, советует, направляет обучающихся. Содержание и объем материала, подлежащего проверке, определяется Программой. При проверке усвоения материала необходимо выявлять полноту, прочность усвоения учащимися теории и умение применять ее на практике в знакомых и незнакомых ситуациях.

Текущий контроль уровня усвоения материала осуществляется по окончании изучения каждой темы – выполнением практических заданий. Итоговый контроль проходит в конце учебного года – в форме зачета, на котором обучающиеся представляют свои работы и обсуждают их.

Формы проведения аттестации:

- тестирование;
- практикум;
- контрольная работа;
- зачетная работа.

УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

Теоретические занятия предполагают:

- лекционные формы (материал выкладывается в сети), • уроки-беседы,
- демонстрационные формы и др.

Практические занятия предполагают:

- самостоятельную работу обучающихся (основа практических занятий);
- совместную работу группы (группы объединяет общая цель работы, ее тема, содержание и сроки сдачи материала преподавателю);
- работу с материалами лекций, дополнительными источниками информации:
 - анализ собственных работ;
 - групповое и индивидуальное консультирование;
 - подготовка и защита индивидуальных и групповых проектов.

Подобная организация обучения способствует развитию познавательной активности и творческих способностей обучающихся.

Материально-технические условия реализации программы

Для успешной реализации Программы необходимо, чтобы рабочее место обучающегося и преподавателя включали в себя:

- компьютеры, обеспечивающие возможность работы с мультимедийным контентом: воспроизведение видеоизображений, качественный стереозвук в наушниках, речевой ввод с микрофона и др.;
- периферийное оборудование:
 - принтер (черно/белой печати, формата А4);
 - устройства для ввода визуальной информации (сканер, цифровой фотоаппарат, web-камера и др.);
 - устройства создания графической информации (графический планшет), использующиеся для создания и редактирования графических объектов, ввода рукописного текста;
 - акустические колонки;
 - оборудование, обеспечивающее подключение к сети Интернет (комплект оборудования для подключения к сети Интернет, сервер).
- программное обеспечение компьютера:
 - операционная система семейства MacOS или Windows;

- программа Python;
- файловый менеджер (в составе операционной системы или др.);
- почтовый клиент (в составе операционных систем или др.);
- браузер (в составе операционных систем или др.);
- мультимедиа проигрыватель (в составе операционной системы или др.);
- антивирусная программа;
- программа-архиватор;
- программа-переводчик;
- программа интерактивного общения;
- текстовый редактор;
- растровый графический редактор;
- звуковой редактор;
- редактор Web-страниц.

СПИСОК ЛИТЕРАТУРЫ

1. Банкрашков А. Программирование для детей на языке Python. – Москва: АСТ, 2017.
2. Бригге Дж. Python для детей. Самоучитель по программированию. – Москва: Манн, 2018.
3. Бэрри П. Изучаем программирование на Python. – Москва: Эксмо, 2017.
4. ВандерПлас Дж. Python для сложных задач. Наука о данных и машинное обучение. – Санкт-Петербург: Питер, 2018.
5. Васильев А. Python на примерах. Практический курс. Москва: Наука и техника, 2018.
6. Вордэрман К. Программирование на Python. Иллюстрированное руководство для детей. – Москва: Манн, Иванов и Фербер, 2018.
7. Гэддис Т. Начинаем программировать на Python. – Санкт-Петербург: БВХ-Петербург, 2019.
8. Доусон М. Программируем на Python. – Санкт-Петербург: Прогресс книга, 2019.
9. Жуков Р. Язык программирования Python: практикум. Учебное пособие. – Москва: Инфра-М, 2020.
10. Златопольский Д.М. Основы программирования на языке Python. – Москва: ДМК Пресс, 2018.
11. Лутц М. Изучаем Python. В 2-х т. – Москва: Вильямс, 2019.
12. Любанович В. Простой Python. Современный стиль программирования. – Санкт-Петербург: Питер, 2019.
13. МакГрат М. Программирование на Python для начинающих. Москва: Эксмо, 2015.
14. Маккини У. Python и анализ данных. – Москва: ДМК Пресс, 2018.
15. Мартелли А., Рейвенскрофт А., Холден С. Python. Справочник. Полное описание языка. – Москва: Диалектика, 2018.
16. Мэтис Э. Изучаем Python. Программирование игр, визуализация данных, веб-приложения. – Санкт-Петербург: Питер, 2020.
17. Мюллер Дж.П. Python для чайников. – Москва: Вильямс, 2019.
18. Персиваль Г. Python. Разработка на основе тестирования. – Москва: ДМК Пресс, 2018.
19. Пратик Дж. Искусственный интеллект с примерами на Python. Создание приложений искусственного интеллекта с помощью Python для взаимодействия с окружающим миром. – Москва: Вильямс, 2019.