

Юго-Восточное управление министерства образования и науки Самарской области

Структурное подразделение государственного бюджетного общеобразовательного учреждения
Самарской области средней общеобразовательной школы "Образовательный центр" имени
Героя Советского Союза Ваничкина Ивана Дмитриевича
с. Алексеевка муниципального района
Алексеевский Самарской области - центр
дополнительного образования детей "Развитие"

Утверждаю
Директор

«01» августа 2023 г.



И.А. Чередникова

Согласовано:

Председатель

методического совета

Лизункова /Т.Н. Лизункова/
«01» августа 2023 г.

Программа рассмотрена на заседании
кафедры «Дополнительное образование»
Протокол № 1 от «01» августа 2023 г.

Руководитель кафедры

Лопатина /Г.В. Лопатина/

Дополнительная общеобразовательная общеразвивающая
программа технической направленности

Мастерская программирования

Возраст обучающихся – 13 -17 лет
Срок реализации программы: 1 год

Разработчик:

Телегина Екатерина Викторовна
педагог дополнительного образования

с. Алексеевка, 2023

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» (далее – Программа) технической направленности 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 IT- 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.

Программа разработана в соответствии с нормативно-правовыми документами:

- Федеральный закон от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации»;
- Указ Президента Российской Федерации «О национальных целях развития Российской Федерации на период до 2030 года»;
- Концепция развития дополнительного образования до 2030 года (утверждена распоряжением Правительства РФ от 31.03.2022 № 678-р)
- ИЗМЕНЕНИЯ, которые вносятся в распоряжение Правительства РФ от 31.03.2022 №678-р (утверждены распоряжением Правительства РФ от 15.05.2023 №1230-р);
- Стратегия развития воспитания в Российской Федерации на период до 2025 года (утверждена распоряжением Правительства Российской Федерации от 29.05.2015 № 996-р);
- План мероприятий по реализации в 2021 - 2025 годах Стратегии развития воспитания в Российской Федерации на период до 2025 года (утвержден распоряжением Правительства Российской Федерации от 12.11.2020 № 2945-р);
- Приказ Министерства образования и науки Российской Федерации от 23.08.2017 № 816 «Об утверждении Порядка применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ»;

- Приказ Министерства просвещения Российской Федерации от 27 июля 2022 г. № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;
- Приказ Министерства просвещения РФ от 03.09.2019 № 467 «Об утверждении Целевой модели развития региональных систем дополнительного образования детей»;
- Приказ Министерства просвещения РФ от 21.04.2023 № 302 «О внесении изменений в Целевую модель развития региональных систем дополнительного образования детей, утвержденную приказом Министерства просвещения РФ от 3.09.2019 г. № 467»;
- Постановление Главного государственного санитарного врача РФ от 28.09.2020 № 28 «Об утверждении СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи»;
- Стратегия социально-экономического развития Самарской области на период до 2030 года (утверждена распоряжением Правительства Самарской области от 12.07.2017 № 441);
- Письмо Министерства образования и науки РФ от 18.11.2015 № 09-3242 «О направлении информации» (с «Методическими рекомендациями по проектированию дополнительных общеразвивающих программ (включая разноуровневые 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 по Программе ведется в разновозрастных группах, которые комплектуются из обучающихся 13-17 лет, имеющих начальные умения и навыки работы в текстовых редакторах, с почтовой программой, с браузером.

Рекомендуемое количество обучающихся в группе – 8-12 человек.

Сроки реализации:

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

Формы и режим занятий

Обучение очное с элементами дистанционного обучения. Формы занятий: лекции, индивидуальные практические занятия, индивидуальные консультации, групповые консультации, групповые практические занятия.

Занятия проводятся в очной форме еженедельно. Занятия проводятся один раз в неделю по 2 часа. Продолжительность одного академического часа – 40 минут.

Объем программы

Распределение часов по темам дано из расчета максимум 72 часа в год.

СОДЕРЖАНИЕ ПРОГРАММЫ

УЧЕБНЫЙ ПЛАН

№ п/п	Тематический блок	Количество часов			Форма контроля итогов
		теория	пр-ка	всего	
1.	Классификация языков программирования. Компиляция, интерпретация. Среда разработки.	1	1	2	практическое задание
2.	Знакомство с языком Python. Синтаксис. Документация. Создание и запуск скриптов.	2	2	4	программный продукт
3.	Встроенные типы данных. Переменные. Основные операции с данными.	3	5	8	программный продукт
4.	Ввод/вывод данных в программу.	2	2	4	программный продукт
5.	Понятие алгоритм. Типы алгоритмов. Линейный алгоритм и с условием. Условный оператор.	2	2	4	программный продукт
6.	Типы ошибок в программировании. Поиск ошибок. Отладка программы. Обработка исключений в Python.	2	2	4	программный продукт
7.	Управляющие инструкции – циклы.	3	7	10	программный продукт
8.	Функции. Встроенные функции. Модули. Использование дополнительных модулей. Создание собственных функций. Создание собственных модулей.	4	6	10	программный продукт
9.	Строки. Списки. Словари. Кортежи.	6	6	12	программный продукт
10.	Файловый ввод/вывод.	2	2	4	программный продукт
11.	Основы объектно-ориентированного Программирования.	3	3	6	программный продукт
12.	Итоговый контроль.		4	4	программный продукт
	ИТОГО:	30	42	72	

Учебно-тематический план

1. Классификация языков программирования. Компиляция, интерпретация. Среда разработки		1	1	2
1	Классификация языков программирования. Компилируемые и интерпретируемые языки. Области применения. Язык программирования Python. Области применения.	1		1
2	<i>Дистрибутивы интерпретатора языка Python. Установка официального Дистрибутива интерпретатора языка Python 3.x. Запуск интерактивного режима Python. Работа в официальном IDLE Python. Работа в среде PyScripter, запуск/остановка скриптов на языке Python.</i>		1	1
2. Знакомство с языком Python. Синтаксис. Документация. Создание и запуск скриптов		2	2	4
3	Документация на язык программирования - PEP8. Стиль написания кода: отступы. Имена переменных, длина строки. Рекомендации по созданию имен переменных. Комментарии.	1		1
4	Ключевые (служебные/зарезервированные слова). Вызов справки, поиск информации на ресурсах.	1		1
5	<i>Первая программа. Подсветка синтаксиса в средах IDLE. Сообщения интерпретатора. Знакомство с функциями print() и input().</i>		1	1
6	<i>Выполнение команд в интерактивном режиме. Создание скриптов. Сохранение скриптов. Программа приветствие.</i>		1	1
3. Встроенные типы данных. Переменные. Основные операции с данными		3	5	8
7	Основные типы данных в программировании. Особенность оперирования данными в языке Python. Переменные, соотношение имени переменной со значением в памяти компьютера. Числовые типы данных. Преобразования числовых типов. <i>Основные операции с числовыми Данными.</i>	1	1	2
8	<i>Базовые функции языка Python Для работы с числовыми Данными. Ввод/вывод числовых Данных. Ограничение точности вычислений при работе с вещественными (float) типом Данных - IEEE754.</i>		1	1
9	Основы строкового типа данных. <i>Склеивание строк.</i>	1	1	2
10	Логический тип данных. Принцип высказываний. Базовые операции И, ИЛИ, НЕ. Примеры использования логических операций.	1		1

11	<i>Использование логических операций. Взаимодействие с числовыми и строковыми типами Данных. Битовые операции.</i>		2	2
4. Ввод/вывод данных в программу.		2	2	4
12	Область видимости переменных. Присваивание значения переменных в коде программы. Ввод значения переменных пользователем с помощью функции input(). <i>Использование функции input(). Преобразование типов вводимых Данных.</i>	1	1	2
13	Функция print() для вывода данных на экран и в файл. <i>Использование функции print(). Формирование строки. Вывод псевдографики.</i>	1	1	2
5. Понятие алгоритм. Типы алгоритмов. Линейный алгоритм и с условием. Условный оператор.		2	2	4
14	Понятие алгоритма их типы. Описание задачи в виде алгоритма.	1		1
15	Алгоритмы с ветвлением. Условный оператор. Синтаксис условного оператора. <i>Применение условного оператора if... else Для решения прикладных задач.</i>	1	1	2
16	<i>Вложенные условия. Множественные условия. Ленивая оценка условий. Сложные условия. Формулировка условий. Создание текстовой игры «Волк, коза и капуста» с использованием только условного оператора.</i>		1	1
6. Типы ошибок в программировании. Поиск ошибок. Отладка программы. Обработка исключений в Python.		2	2	4
17	Отладка программ. Типы ошибок: синтаксические, ошибки выполнения, семантические. Использование исключений для обработки ошибок выполнения без аварийного завершения программы.	2		2
18	<i>Разработка алгоритма программ с учетом возможных ошибок выполнения. Сообщения интерпретатора об ошибках. Поиск Документации по ошибкам.</i>		1	1
19	<i>Отладка программы с семантическими ошибками: экспериментальная отладка, создание контрольных меток.</i>		1	1
7. Управляющие инструкции - циклы.		3	7	10
20	Циклические алгоритмы. Циклы с предусловием и постусловием. Счетчик итераций.	1		1
21	Цикл for, синтаксис. Функция range(). Команды break и continue. Вложенные циклы for. <i>Решение задач с помощью цикла for.</i>	1	2	3

22	Цикл while, синтаксис. Счетчик итераций. Команды break и continue. Вложенные циклы. <i>Бесконечный цикл. Формулировка условия. Решение задач с использованием циклов while.</i>	1	2	3
23	<i>Решение задач различного типа с помощью циклических алгоритмов.</i>		3	3
8. Функции. Встроенные функции. Модули. Использование дополнительных модулей. Создание собственных функций. Создание собственных модулей.		4	6	10
24	Понятие функция. Назначение. Встроенные функции. Вызов справки по функциям. Встроенные модули, назначение. Вызов справки по модулям.	2		2
25	Математические функции. Подключение модулей. Модуль Math. <i>Использование функций модуля Math Для решения математических задач.</i>	1	1	2
26	Создание собственных функций: определение и использование. Параметры и аргументы функций. Локальные параметры и переменные. Возвращаемое значение. Область видимости переменных. <i>Использование функций в программах. Создание собственных модулей.</i>	1	1	2
27	<i>Создание игры «Волк, коза и капуста» с использованием циклического алгоритма.</i>		2	2
28	<i>Знакомство с модулем NumPy и matplotlib.</i>		1	1
29	<i>Знакомство с модулем tkinter.</i>		1	1
9. Строки. Списки. Словари. Кортежи.		6	6	12
30	Строка как последовательность. Функция len. Перебор элементов строки. Срезы строк. Поиск. Подсчет. <i>Методы строк. Сравнение строк. Работа с текстовыми файлами.</i>	1	1	2
31	Списки. Перебор элементов списка. Операции со списками. Срезы списков. <i>Методы списков. Отображение, фильтрация и сокращение. Удаление элементов списка. Создание синонимов в списке.</i>	1	1	2
32	Словари. Словари как счетчики. Циклы и словари. Обратный поиск. <i>Словари и списки. Глобальные переменные.</i>	2	2	4
33	Кортежи. Неизменяемый тип. Кортежи и операции присваивания. Кортежи в качестве возвращаемого значения. <i>Кортежи с переменным числом аргументов. Списки и кортежи. Словари и кортежи. Сравнение кортежей. Последовательности последовательностей.</i>	2	2	4

10. Файловый ввод/вывод.		2	2	4
34	Понятие файл. Имена файлов. Место положение файлов. <i>Открытие файла. Чтение файла. Текстовые и бинарные файлы. Запись в файл.</i>	1	1	2
35	Базы данных. <i>Сериализация. Десериализация. Конвейеры.</i>	1	1	2
11. Основы объектно-ориентированного программирования.		3	3	6
36	Основы объектно-ориентированного подхода. Объекты в реальном мире, объекты в Python. Объект = атрибуты + методы. <i>Создание объектов. Примеры класса.</i>	1	1	2
37	Скрываем данные. Полиморфизм и наследование. Экземпляры класса в качестве возвращаемого значения. <i>Использование классов в программах.</i>	2	2	4
12. Итоговая аттестация.			4	4
<i>Индивидуальное задание. Разработка концепции программы по заданию. Написание и отладка программы. Презентация программного продукта.</i>			4	4
ИТОГО:		30	42	72

СОДЕРЖАНИЕ УЧЕБНО-ТЕМАТИЧЕСКОГО ПЛАНА

1. Классификация языков программирования. Компиляция, интерпретация. Среда разработки.

Теория:

Классификация языков программирования. Компилируемые и интерпретируемые языки. Области применения. Язык программирования Python. Достоинства и недостатки. Области применения. Интерактивный режим. Официальный сайт языка Python. Дистрибутивы языка Python.

Практические занятия:

1. Дистрибутивы интерпретатора языка Python.
2. Установка официального дистрибутива интерпретатора языка Python 3.x.
3. Запуск интерактивного режима Python.
4. Работа в официальном IDLE Python.
5. Работа в среде PyScripter, настройка, запуск/остановка скриптов на языке Python.

2. Знакомство с языком Python. Синтаксис. Документация. Создание и запуск скриптов.

Теория:

Документация на язык программирования – PEP8. Стиль написания кода: отступы,

базовый синтаксис. Имена переменных, длина строки. Рекомендации по созданию имен переменных. Ключевые (служебные/зарезервированные слова). Первая программа. Сообщения интерпретатора.

Практические занятия:

1. Первая программа.
2. Подсветка синтаксиса в средах IDLE.
3. Сообщения интерпретатора.
4. Знакомство с функциями print() и input().
5. Выполнение команд в интерактивном режиме.
6. Создание скриптов. Сохранение скриптов.
7. Программа приветствие.

3. Встроенные типы данных. Переменные. Основные операции с данными.

Теория:

Основные типы данных в программировании. Особенность оперирования данными в языке Python. Переменные, соотношение имени переменной со значением в памяти компьютера. Числовые типы данных. Преобразования числовых типов. Ограничение точности вычислений при работе с вещественными (float) типом данных – IEEE754. Основы строкового типа данных. Склеивание строк. Логический тип данных. Принцип высказываний. Базовые операции И, ИЛИ, НЕ.

Практические занятия:

1. Основные операции с числовыми данными.
2. Базовые функции языка Python для работы с числовыми данными.
3. Ввод/вывод числовых данных.
4. Ограничение точности вычислений при работе с вещественными (float) типом данных – IEEE754.
5. Склеивание строк.
6. Использование логических операций.
7. Взаимодействие с числовыми и строковыми типами данных.
8. Битовые операции.

4. Ввод/вывод данных в программу.

Теория:

Область видимости переменных. Присваивание значения переменных в коде программы. Ввод значения переменных пользователем с помощью функции input(). Функция print() для вывода данных на экран и в файл. Формирование строки. Вывод псевдографики.

Практические занятия:

1. Использование функции input().
2. Преобразование типов вводимых данных.
3. Использование функции print().
4. Формирование строки.
5. Вывод псевдографики.

5. Понятие алгоритм. Типы алгоритмов. Линейный алгоритм и с условием. Условный оператор.

Теория:

Понятие алгоритма их типы. Описание задачи в виде алгоритма. Алгоритмы с ветвлением. Условный оператор. Синтаксис условного оператора. Вложенные условия. Множественные условия. Ленивая оценка условий. Сложные условия. Формулировка условий.

Практические занятия:

1. Применение условного оператора if...else для решения прикладных задач.
2. Вложенные условия.
3. Множественные условия.
4. Ленивая оценка условий.
5. Сложные условия.
6. Формулировка условий.
7. Создание текстовой игры «Волк, коза и капуста» с использованием только условного оператора.

6. Типы ошибок в программировании. Поиск ошибок. Отладка программы. Обработка исключений в Python.

Теория:

Отладка программ. Типы ошибок: синтаксические, ошибки выполнения, семантические. Сообщения интерпретатора об ошибках. Поиск документации по ошибкам. Использование исключений для обработки ошибок выполнения без аварийного завершения программы. Разработка алгоритма программ с учетом возможных ошибок выполнения. Отладка программы с семантическими ошибками.

Практические занятия:

1. Сообщения интерпретатора об ошибках.
2. Поиск документации по ошибкам.
3. Разработка алгоритма программ с учетом возможных ошибок выполнения.
4. Отладка программы с семантическими ошибками: экспериментальная отладка, создание контрольных меток.

7. Управляющие инструкции – циклы.

Теория:

Циклические алгоритмы. Циклы с предусловием и постусловием. Счетчик итераций. Цикл `for`, синтаксис. Функция `range()`. Команды `break` и `continue`. Вложенные циклы `for`. Цикл `while`, синтаксис. Счетчик итераций. Команды `break` и `continue`. Вложенные циклы. Бесконечный цикл. Формулировка условия. Составление оптимального алгоритма программы.

Практические занятия:

1. Решение задач с помощью цикла `for`.
2. Бесконечный цикл.
3. Формулировка условия.
4. Решение задач с использованием циклов `while`.
5. Решение задач различного типа с помощью циклических алгоритмов.

8. Функции. Встроенные функции. Модули. Использование дополнительных модулей. Создание собственных функций. Создания собственных модулей.

Теория:

Понятие функция. Назначение. Встроенные функции. Вызов справки по функциям. Встроенные модули, назначение. Вызов справки по модулям. Математические функции. Подключение модулей. Модуль `Math`. Создание собственных функций: определение и использование. Параметры и аргументы функций. Локальные параметры и переменные. Возвращаемое значение. Область видимости переменных. Создание собственных модулей.

Практические занятия:

1. Использование функций модуля `Math` для решения математических задач.
2. Использование функций в программах.
3. Создание собственных модулей.
4. Создание игры «Волк, коза и капуста» с использованием циклического алгоритма.
5. Знакомство с модулем `NumPy` и `matplotlib`.
6. Знакомство с модулем `tkinter`.

9. Строки. Списки. Словари. Кортежи.

Теория:

Строка как последовательность. Функция `len`. Срезы строк. Методы строк. Работа с текстовыми файлами. Списки. Перебор элементов списка. Операции со списками. Срезы списков. Отображение, фильтрация и сокращение. Удаление элементов. Создание синонимов. Словари. Словари как счетчики. Циклы и словари. Обратный поиск. Словари и списки. Глобальные переменные. Кортежи. Неизменяемый тип.

Кортежи и операции присваивания. Кортежи в качестве возвращаемого значения.

Кортежи с переменным числом аргументов.

Практические занятия:

1. Методы строк.
2. Сравнение строк.
3. Работа с текстовыми файлами.
4. Методы списков.
5. Отображение, фильтрация и сокращение.
6. Удаление элементов списка.
7. Создание синонимов в списке.
8. Словари и списки.
9. Глобальные переменные.
10. Кортежи с переменным числом аргументов.
11. Списки и кортежи.
12. Словари и кортежи.
13. Сравнение кортежей.
14. Последовательности последовательностей.

10. Файловый ввод/вывод.

Теория:

Понятие файл. Имена файлов. Место положение файлов. Открытие файла. Чтение файла. Текстовые и бинарные файлы. Запись в файл. Базы данных. Сериализация. Десериализация. Конвейеры.

Практические занятия:

1. Открытие файла.
2. Чтение файла.
3. Текстовые и бинарные файлы.
4. Запись в файл.
5. Сериализация.
6. Десериализация.
7. Конвейеры.

11. Основы объектно-ориентированного программирования.

Теория:

Основы объектно-ориентированного подхода. Объекты в реальном мире, объекты в Python. Объект = атрибуты + методы. Создание объектов. Скрываем данные. Полиморфизм и наследование. Экземпляры класса в качестве возвращаемого значения.

Практические занятия:

1. Создание объектов.
2. Примеры класса.
3. Использование классов в программах.

12. Итоговый контроль

Практические занятия:

1. Индивидуальное задание.
2. Разработка концепции программы по заданию.
3. Написание и отладка программы.
4. Презентация программного продукта.

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ

1. Обучающие

- знание синтаксиса языка и базовых алгоритмических конструкций, умение применять теоретические знания для решения конкретных проблем;
- умение разрабатывать, тестировать и отлаживать несложные программы;
- понимание, что такое проект, структура и дизайн проекта;
- устойчивая мотивация к познавательной и творческой деятельности.

2. Воспитательные

- сформированность коммуникативной культуры обучающихся, навыков работы в команде;
- самостоятельность в изучении дополнительного материала по пройденным темам;
- профориентация.

3. Развивающие

- навыки применения логики при решении поставленных задач; – проявление творческой активности обучающихся.
- создание продуктов оригинальной, творческой деятельности;
- проявление интереса обучающихся к программированию;
- проявление трудолюбия и целеустремленности.

Основными критериями оценки эффективности образовательного процесса являются:

- степень сформированности основных знаний, умений и навыков, предусмотренных программой;
- способность практически применять знания при создании программного продукта, при решении задач;
- писать программы без ошибок или устранять ошибки при отладке программы.

Обучающиеся, освоившие программу в полном объеме, должны:

- уметь качественно формулировать и излагать свои мысли;
- уметь работать как самостоятельно, так и в команде;
- знать средства языка программирования Python в рамках курса;
- организовывать самостоятельное обучение, стремиться к самосовершенствованию;
- анализировать текст чужих программ, находить в них неточности, оптимизировать алгоритм, создавать собственные варианты решения.

КОМПЛЕКС ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИХ УСЛОВИЙ

УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

Материально-техническое обеспечение:

Для реализации программы имеется компьютерный класс, мультимедийный проектор, необходимое программное обеспечение (среда программирования языка Python, операционная система Windows, пакет Microsoft Office), компьютеры, принтер и копировальный аппарат, интернет.

Кадровое обеспечение:

Дополнительную общеразвивающую программу реализуют педагоги дополнительного образования с образованием в области программирования.

Информационное обеспечение:

Официальный сайт языка Python - <https://www.python.org/>

Интерактивный учебник языка Питон - <http://pythontutor.ru/Python3>

для начинающих - <https://pythonworld.ru/>

База знаний, примеры по программированию на языке Python -

<https://pythonru.com> Сайт «Школа программиста» — <https://acmp.ru/> Подборка

материалов по языку Python - <https://proglib.io/p/learning-python/>Выполнение

программы онлайн - <https://rextester.com/1/python3>

ФОРМЫ КОНТРОЛЯ

Программа предполагает обязательную промежуточную и итоговую аттестацию по всем темам программы обучения, для осуществления мониторинга освоения образовательной программы. Помимо итоговой аттестации проводятся периодические проверки знаний в форме индивидуальных заданий после изучения одной или нескольких тем, с целью выявить возможные проблемы с усвоением материала курса.

Итоговый контроль предполагает дифференциальный зачет в виде

презентации программы. Обучающимся предлагается небольшое индивидуальное задание для создания программы, предполагается игра с графическим интерфейсом. По результатам обучающийся может получить одну из трёх оценок: *незачёт*, *зачёт* и *зачёт с повышенным освоением программы*. Критерием проверки индивидуальных работ: программа записана без ошибок, программа написана полностью самостоятельно, или взята готовая, понята и модифицирована. Повышенным уровнем освоения программы будет считаться применение оригинальных решений в итоговой работе.

МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

Python поддерживает множество различных парадигм программирования: объектно-ориентированное, структурное, функциональное и др.

Python – язык с динамической типизацией данных, интерпретируемый язык, язык сценариев (скриптов). Это означает, что программы написанные на python требуют для своего запуска компилятор (интерпритатор) языка. Python обладает простым и лаконичным синтаксисом, а богатая стандартная библиотека является одной из причин широкой популярности языка в совершенно различных сферах деятельности.

Разработка языка Python была начата в конце 1980-х годов сотрудником голландского института CWI Гвидо ван Россумом. Разработка языка идет по двум направлениям (версии языка), не сохраняющих совместимость – это версия 2.7 и 3.x. Подробно об этом на официальном сайте Python. В настоящее время актуальной версией языка является 3.8.

- Официальный сайт: www.python.org
- Страница загрузки: www.python.org/download/

Язык программирования **Python 3.x** входит в состав допустимых языков программирования на Всероссийской олимпиаде школьников по информатике. В демонстрационной версии ЕГЭ — Python один из языков на котором представлен программный код.

Методические материалы

- Allen Downey. Думать на языке Python.
- Федоров Д.Ю. Основы программирования на примере языка Python.

Задачники по программированию

- М.Э. Абрамян 1000 задач по программированию. Часть I. Скалярные типы данных, управляющие операторы, процедуры и функции.
- М.Э. Абрамян 1000 задач по программированию. Часть II. Минимумы и максимумы, одномерные и двумерные массивы, символы и строки, двоичные файлы.

- М. Э. Абрамян 1000 задач по программированию. Часть III. Текстовые файлы, составные типы данных в процедурах и функциях, рекурсия, указатели и динамические структуры.
- Златопольский Д.М. Сборник задач по программированию.

Методическая основа программы «Мастерская программирования» предназначена для начального знакомства с основными алгоритмами и с программированием на языке Python в интегрированных средах разработки (IDE) Python и PyScripter. В программе изучаются типовые задачи в рамках базового курса информатики, алгоритмы их решения и реализация этих алгоритмов на языке Python. Дополнительно в курсе описываются основы программирования графики в Python с использованием кросс-платформенной библиотеки Tkinter.

ЛИТЕРАТУРА ДЛЯ ОБУЧАЮЩИХСЯ

1. Allen Downey. Думать на языке Python. Green Tea Press. 2012. Перевод на русский язык Николай Орехов 2017. https://bitbucket.org/thinkpython_ru/book/src
2. Федоров Д. Ю. Основы программирования на примере языка Python. //Учебное пособие. – Санкт-Петербург: 2016.
3. Васильев А.Н. Python на примерах. Практический курс по программированию. - СПб.: Наука и Техника, 2016. - 432 с.: ил.
4. Доусен М. Програмуємо на Python. - СПб.: Питер, 2014. - 416 с.: ил.
5. Пейн, Брайсон. Python для детей и родителей / Брайсон Пейн. – Москва: Издательство «Э». 2017. – 352 с.: ил.
6. Хайнеман, Джордж, Пояяис, Гэри, Сеяков, Стэнли. Алгоритмы. Справочник с примерами на C, C++, Java и Python, 2-е изд.: Пер. с англ. — СПб.: ООО —Альфа-книга, 2017. — 432 с.: ил. — Парал. тит. англ.
7. Седжвик, Роберт, Уэйн, Кевин, Дондеро, Роберт. Программирование на языке Python: учебный курс.: Пер. с англ. - СПб. : ООО "Альфа-книга": 2017. - 736 с.: ил. - Парал. тит. англ.
8. У. Сэнд, К. Сэнд. Hello World! Занимательное программирование. — СПб.: Питер, 2016. — 400 с.: ил. — (Серия «Вы и ваш ребенок»).

ЛИТЕРАТУРА ДЛЯ ПЕДАГОГА

1. Бизли Д. М. Язык программирования Python : справочник : пер. с англ. / Д.М. Бизли. – Киев : ДинаСофт, 2000
2. Гифт Н. Python в системном администрировании UNIX и Linux: пер. с англ. /Н.

- Гифт, Д. Джонс. – СПб.: Символ-Плюс, 2009
3. Лейнингем И. Освой самостоятельно Python за 24 часа: пер. с англ. /И. Лейнингем. – М.: Издательский дом «Вильямс», 2001
 4. Лесса А. Python. Руководство разработчика: пер. с англ. / А. Лесса. – СПб.Диасофт ЮП, 2001
 5. Лутц М. Изучаем Python: пер. с англ. / М. Лутц. – СПб.: Символ-Плюс, 2009
 6. Лутц М. Программирование на Python : пер.с англ. / М. Лутц. – СПб.: Символ-Плюс, 2002
 7. Саммерфельд М. Программирование на Python 3 Подробное руководство : пер. с англ. / М. Саммерфельд. – СПб.: Символ-Плюс, 2009
 8. Сузи Р. А. Python / Р. А. Сузи. – СПб.: БХВ-Петербург, 2002
 9. Сузи Р. А. Язык Python и его применения : учеб. пособие / Р.А. Сузи. – М. : Интернет - Университет информационных технологий: БИНОМ. Лаборатория знаний, 2006
 10. Язык программирования Python / Г. Россум [и др.]. – СПб.: АНО «Институт логики» – Невский диалект, 2001
 11. Сэнд У., Сэнд К. Hello World! Занимательное программирование. — СПб.:Питер,2016
 12. Шоу, Зед. Легкий способ выучить Python / Зед Шоу ; [пер. с англ. М. А.Райтмана]. — М: Издательство «Э», 2017.
 13. Мэтиз Эрик. Изучаем Python. Программирование игр, визуализация данных, веб-приложения. — СПб.: Питер, 2017.
 14. Прохоренок, Н. А. Python 3 и PyQt 5. Разработка приложений / Н. А. Прохоренок, В. А. Дронов. – СПб.: БХВ-Петербург, 2016
 15. Саммерфилд М. Программирование на Python 3. Подробное руководство. СПб.:Символ-Плюс — 2009
 16. Златопольский Д. М. Основы программирования на языке Python. – М.: ДМКПресс, 2017
 17. Пэйн, Брайсон Python для детей и родителей. М.: «Э», 2017
 18. Лучано Рамальо Python. К вершинам мастерства / Пер. с англ. Слинкин А. А. —М.: ДМК Пресс, 2016
 19. Гифт Н.,Джонс Д. Python в системном администрировании UNIX и Linux. СПб.: Символ-Плюс, 2009
 20. М.Э. Абрамян. 1000 задач по программированию. Часть I. Методические указания для студентов механико-математического, физического и экономического факультетов, ГОУ ПО РФ Ростовский государственный университет, 2004.
 21. М.Э. Абрамян. 1000 задач по программированию. Часть II. Методические указания для студентов механико-математического, физического и экономического факультетов, ГОУ ПО РФ Ростовский государственный университет, 2004.

22. М.Э. Абрамян. 1000 задач по программированию. Часть III. Методические указания для студентов механико-математического, физического и экономического факультетов, ГОУ ПО РФ Ростовский государственный университет, 2004.
23. Златопольский Д.М. Сборник задач по программированию. – 3-е изд., перераб. и доп. – СПб.