

**Государственное бюджетное общеобразовательное учреждение  
средняя общеобразовательная школа № 571 с углублённым изучением ан-  
глийского языка Невского района Санкт-Петербурга**

ПРИНЯТА

Педагогическим советом ГБОУ  
школы № 571 с углубленным изу-  
чением английского языка  
Невского района Санкт-  
Петербурга  
Протокол №7 от «29» августа 2025г.

УТВЕРЖДЕНА

Директором  
ГБОУ школы № 571 с углубленным  
изучением английского языка Невского  
района Санкт-Петербурга  
В.Е.Григорьев  
Приказ № 49-ДО от 25.08.2025г.



**ДОПОЛНИТЕЛЬНАЯ ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА**

**«Программирование на языке Python»**

Срок освоения программы: 1 год

Возраст обучающихся: 12-18 лет

Разработчик:

**Ложкина Ольга Владимировна,  
педагог дополнительного образования**

Санкт-Петербург  
2025

## Пояснительная записка

**Дополнительная общеразвивающая программа «Программирование на Python»** имеет **техническую направленность**

**Адресат программы:** обучающиеся от 12 до 18 лет

**Актуальность программы обусловлена** широким распространением информационно-коммуникационных технологий в обществе и необходимостью обеспечивать связанную с этим инфраструктуру специалистами. Всё большее значение приобретает умение человека грамотно обращаться с компьютером, причём на уровне начинающего программиста.

Изучение основных принципов программирования невозможно без регулярной практики написания программ на каком-либо языке программирования. Для обучения в рамках программы выбран язык Python, который является достаточно эффективным и доступным инструментом достижения задач в области создания программных продуктов.

Синтаксис языка достаточно прост и интуитивно понятен, что понижает порог вхождения и позволяет сосредоточиться на алгоритмических аспектах программирования.

**Уровень освоения программы: базовый**

**Объем ДОП: 72 часа**

**Срок освоения: 1 год**

**Целью реализации программы** является изучение основ программирования на языке Python, основных приемов написания программ на современном языке программирования, развитие алгоритмического мышления обучающихся, творческих способностей, аналитических и логических компетенций.

**Задачи программы:**

*Образовательные (предметные):*

- Познакомить с основными алгоритмическими конструкциями и правилами их записи, с основными способами организации данных.
- Научить составлять и записывать алгоритмы с использованием соответствующих алгоритмических конструкций.
- Научить распознавать необходимость применения той или иной алгоритмической конструкции при решении задач.
- Научить организовывать данные для эффективной алгоритмической обработки.
- Научить разрабатывать алгоритмы и реализовывать их на языке программирования Python.
- Научить осуществлять отладку и тестирование программы.

*Личностные:*

- повышение общекультурного уровня;
- вооружение правильным методологическим подходом к познавательной и практической деятельности;
- выделение и раскрытие роли информационных технологий и компьютеров в развитии современного общества;
- привитие навыков сознательного и рационального использования компьютера в своей учебной, а затем и профессиональной деятельности;
- воспитание стремления к овладению техникой исследования;
- воспитание трудолюбия, инициативности и настойчивости в преодолении трудностей.

*Метапредметные:*

- владение умениями организации собственной учебной деятельности, включающими: целеполагание как постановку учебной задачи на основе соотнесения того, что уже известно, и того, что требуется установить;
- планирование – определение последовательности промежуточных целей с учетом конечного результата, разбиение задачи на подзадачи, разработка последовательности и структуры действий, необходимых для достижения цели при помощи фиксированного набора средств;

- прогнозирование – предвосхищение результата

## **Планируемые результаты**

### *Предметные:*

- умение определять результат выполнения алгоритма при заданных исходных данных, узнавать
- изученные алгоритмы обработки чисел и числовых последовательностей, создавать на их основе несложные программы анализа данных, читать и понимать несложные программы, написанные на выбранном для изучения универсальном алгоритмическом языке высокого уровня;
- формирование представлений об основных предметных понятиях («информация», «алгоритм», «исполнитель», «модель») и их свойствах;
- развитие логических способностей и алгоритмического мышления, умения составить и записать алгоритм для конкретного исполнителя, знакомство с основными алгоритмическими структурами — линейной, условной и циклической;
- умение выполнять пошагово (с использованием компьютера или вручную) несложные алгоритмы управления исполнителями и анализа числовых и текстовых данных;
- навыки и опыт разработки программ в выбранной среде программирования, включая тестирование и отладку программ;
- умение использовать основные управляющие конструкции объектно-ориентированного программирования и библиотеки прикладных программ, выполнять созданные программы;
- умение разрабатывать и использовать компьютерно-математические модели, оценивать числовые параметры моделируемых объектов и процессов, интерпретировать результаты, получаемые в ходе моделирования реальных процессов, анализировать готовые модели на предмет соответствия реальному объекту или процессу;
- формирование умения соблюдать нормы информационной этики и права.

### *Метапредметные:*

- умение самостоятельно ставить и формулировать для себя новые задачи, развивать мотивы своей познавательной деятельности;
- умение самостоятельно планировать пути решения поставленной проблемы для получения эффективного результата; понимание, что в программировании длинная программа не всегда лучшая;
- умение критически оценивать правильность решения учебно- исследовательской задачи;
- умение корректировать свои действия, вносить изменения в программу и отлаживать её в соответствии с изменяющимися условиями;
- владение основами самоконтроля, способность к принятию решений;
- умение создавать, применять и преобразовывать знаки и символы, модели и схемы для решения учебно-исследовательских и проектных работ;
- формирование и развитие компетентности в области использования информационно-коммуникационных технологий (ИКТ-компетенция);
- умение организовывать учебное сотрудничество и совместную деятельность с учителем и сверстниками в процессе проектной и учебно- исследовательской деятельности.

### *Личностные:*

- формирование ответственного отношения к учению, способности довести до конца начатое дело аналогично завершённым творческим учебным проектам;
- формирование способности к саморазвитию и самообразованию средствами информационных технологий на основе приобретённой благодаря иллюстрированной среде программирования мотивации к обучению и познанию;
- развитие опыта участия в социально значимых проектах, повышение уровня самооценки благодаря реализованным проектам;
- формирование коммуникативной компетентности в общении и сотрудничестве со сверстниками в процессе образовательной, учебно- исследовательской и проектной деятельности, участия в конкурсах и конференциях различного уровня;
- формирование целостного мировоззрения, соответствующего современному уровню развития информационных технологий;
- формирование осознанного позитивного отношения к другому человеку, его мнению, результату его деятельности;

- формирование ценности здорового и безопасного образа жизни; усвоение правил индивидуального и коллективного безопасного поведения при работе с компьютерной техникой.

## **Организационно-педагогические условия реализации программы**

### **Формы обучения**

Форма обучения – очная.

### **1.6. Режим занятий**

Режим занятий определяется календарным учебным графиком.

Согласно СанПиН 2.4.3648-20 занятия могут проводиться два академических часа, т.е. по 45 минут с перерывом 10 мин.

Занятия проводятся в группах численностью 12 человек

Условия набора обучающихся в группу: программа рассчитана на обучающихся в возрасте от 12 до 18 лет, не требует предварительных знаний и входного контроля.

### **1.7. Методы обучения**

Для достижения поставленной цели и реализации задач программы используются следующие методы обучения:

- словесный: объяснение, беседа, рассказ.
- наглядный: показ, наблюдение, демонстрация приемов работы.
- практический: выполнение практических заданий, направленных на овладение комплексом специальных умений.

**2. УЧЕБНЫЙ ПЛАН**  
**дополнительной общеобразовательной (общеразвивающей) программы**  
**«Программирование на Python»**

Категория обучающихся – возраст от 12 до 18 лет

Срок обучения – 72 часа

Форма обучения – очная

Форма аттестации – защита индивидуальных или групповых проектов

№ п/п	Тема занятий	Количество часов			Формы аттеста- ции/ контроля
		Всего	Теория	Практика	
1	Раздел 1. Знакомство с языком Python	5	2	3	Тестирование,
2	Раздел 2. Переменные и выражения	8	3	5	Тестирование,
3	Раздел 3. Условные предложения	7	3	4	Тестирование,
4	Раздел 4. Циклы	10	4	6	Тестирование,
5	Раздел 5. Функции	10	4	6	Тестирование,
6	Раздел 6. Строки - последовательности символов	7	3	4	Решение практических задач
7	Раздел 7. Сложные типы данных	17	6	11	Тестирование,
8	Раздел 8. Стиль программирования и отладка программ	8	2	6	Решение практических задач, презентация проекта, рефлексия.
	ВСЕГО	72	27	45	

ДО от 29.08.2025г.



на 2025-2026 учебный год

Год обучения	Дата начала занятий	Дата окончания занятий	Количество учебных недель	Количество учебных дней	Количество учебных часов	Режим занятий
1 год	11.09.2025	25.05.2025	36	36	72	2 раза в неделю по 1 учебному часу

**Государственное бюджетное общеобразовательное учреждение  
средняя общеобразовательная школа № 571 с углублённым изучением английского  
языка Невского района Санкт-Петербурга**

ПРИНЯТА

Педагогическим советом ГБОУ  
школы № 571 с углубленным изу-  
чением английского языка  
Невского района Санкт-  
Петербурга  
Протокол №7 от «29» августа 2025г.

УТВЕРЖДЕНА

Директором  
ГБОУ школы № 571 с углубленным  
изучением английского языка Невского  
района Санкт-Петербурга  
В.Е.Григорьев  
Приказ № 49-ДО от 25.08.2025г.



**Рабочая программа к дополнительной  
общеразвивающей программе**

**«Программирование на языке Python»**

**Год обучения: 2025-2026**

Разработчик:  
педагог дополнительного образования  
Ложкина Ольга Владимировна

## **Задачи программы:**

### *Образовательные (предметные):*

- Познакомить с основными алгоритмическими конструкциями и правилами их записи, с основными способами организации данных.
- Научить составлять и записывать алгоритмы с использованием соответствующих алгоритмических конструкций.
- Научить распознавать необходимость применения той или иной алгоритмической конструкции при решении задач.
- Научить организовывать данные для эффективной алгоритмической обработки.
- Научить разрабатывать алгоритмы и реализовывать их на языке программирования Python.
- Научить осуществлять отладку и тестирование программы.

### *Личностные:*

- повышение общекультурного уровня;
- вооружение правильным методологическим подходом к познавательной и практической деятельности;
- выделение и раскрытие роли информационных технологий и компьютеров в развитии современного общества;
- привитие навыков сознательного и рационального использования компьютера в своей учебной, а затем и профессиональной деятельности;
- воспитание стремления к овладению техникой исследования;
- воспитание трудолюбия, инициативности и настойчивости в преодолении трудностей.

### *Метапредметные:*

- владение умениями организации собственной учебной деятельности, включающими: целеполагание как постановку учебной задачи на основе соотнесения того, что уже известно, и того, что требуется установить;
- планирование – определение последовательности промежуточных целей с учетом конечного результата, разбиение задачи на подзадачи, разработка последовательности и структуры действий, необходимых для достижения цели при помощи фиксированного набора средств;
- прогнозирование – предвосхищение результата

## **Планируемые результаты**

### *Предметные:*

- умение определять результат выполнения алгоритма при заданных исходных данных, узнавать
- изученные алгоритмы обработки чисел и числовых последовательностей, создавать на их основе несложные программы анализа данных, читать и понимать несложные программы, написанные на выбранном для изучения универсальном алгоритмическом языке высокого уровня;
- формирование представлений об основных предметных понятиях («информация», «алгоритм», «исполнитель», «модель») и их свойствах;
- развитие логических способностей и алгоритмического мышления, умения составить и записать алгоритм для конкретного исполнителя, знакомство с основными алгоритмическими структурами — линейной, условной и циклической;
- умение выполнять пошагово (с использованием компьютера или вручную) несложные алгоритмы управления исполнителями и анализа числовых и текстовых данных;
- навыки и опыт разработки программ в выбранной среде программирования, включая тестирование и отладку программ;
- умение использовать основные управляющие конструкции объектно-ориентированного программирования и библиотеки прикладных программ, выполнять созданные программы;
- умение разрабатывать и использовать компьютерно-математические модели, оценивать числовые параметры моделируемых объектов и процессов, интерпретировать результаты, получаемые в ходе моделирования реальных процессов, анализировать готовые модели на предмет соответствия реальному объекту или процессу;



- формирование умения соблюдать нормы информационной этики и права.

*Метапредметные:*

- умение самостоятельно ставить и формулировать для себя новые задачи, развивать мотивы своей познавательной деятельности;
- умение самостоятельно планировать пути решения поставленной проблемы для получения эффективного результата; понимание, что в программировании длинная программа не всегда лучшая;
- умение критически оценивать правильность решения учебно- исследовательской задачи;
- умение корректировать свои действия, вносить изменения в программу и отлаживать её в соответствии с изменяющимися условиями;
- владение основами самоконтроля, способность к принятию решений;
- умение создавать, применять и преобразовывать знаки и символы, модели и схемы для решения учебно-исследовательских и проектных работ;
- формирование и развитие компетентности в области использования информационно-коммуникационных технологий (ИКТ-компетенция);
- умение организовывать учебное сотрудничество и совместную деятельность с учителем и сверстниками в процессе проектной и учебно- исследовательской деятельности.

*Личностные:*

- формирование ответственного отношения к учению, способности довести до конца начатое дело аналогично завершённым творческим учебным проектам;
- формирование способности к саморазвитию и самообразованию средствами информационных технологий на основе приобретённой благодаря иллюстрированной среде программирования мотивации к обучению и познанию;
- развитие опыта участия в социально значимых проектах, повышение уровня самооценки благодаря реализованным проектам;
- формирование коммуникативной компетентности в общении и сотрудничестве со сверстниками в процессе образовательной, учебно- исследовательской и проектной деятельности, участия в конкурсах и конференциях различного уровня;
- формирование целостного мировоззрения, соответствующего современному уровню развития информационных технологий;
- формирование осознанного позитивного отношения к другому человеку, его мнению, результату его деятельности;
- формирование ценности здорового и безопасного образа жизни; усвоение правил индивидуального и коллективного безопасного поведения при работе с компьютерной техникой.

### Учебно-тематический план

№ п/п	Тема занятий	Количество часов			Формы атте- стации/ кон- троля
		Всего	Теория	Практика	
Раздел 1. Знакомство с языком Python					
1	Инструктаж по технике безопасности в компьютерном классе. Общие сведения о языке		1		Тестирование, решение практических задач
2	Общие сведения о языке			1	
3	Установка программыPython		1		
4	Режимы работы с Python			1	
5	Режимы работы с Python			1	

### Раздел 2. Переменные и выражения

6	Переменные		1		Тестирование, решение практических задач
7	Работа со справочной системой			1	
8	Выражения		1		
9	Выражения			1	
10	Ввод и вывод		1		
11	Ввод и вывод			1	
12	Задачи на элементарные действия с числами			1	
13	Задачи на элементарные действия с числами			1	

### Раздел 3. Условные предложения

14	Логические выражения и операторы.		1		Тестирование, решение практических задач
15	Логические выражения и операторы.			1	
16	Условный оператор		1		
17	Условный оператор			1	
18	Множественное ветвление		1		
19	Множественное ветвление			1	
20	Множественное ветвление			1	

### Раздел 4. Циклы

21	Оператор цикла с условием		1		Тестирование, решение практических задач
22	Оператор цикла с условием			1	
23	Оператор цикла с условием			1	
24	Оператор цикла for		1		
25	Решение задач с циклом for.			1	
26	Решение задач с циклом for.			1	
27	Вложенные циклы		1		
28	Реализация циклических алгоритмов			1	
29	Случайные числа		1		
30	Случайные числа			1	

### Раздел 5. Функции

31	Создание функций		1		Тестирование, решение практических задач
----	------------------	--	---	--	---------------------------------------------------

32	Создание функций			1	
33	Создание функций			1	
34	Локальные переменные		1		
35	Локальные переменные			1	
36	Примеры решения задач с использованием функций		1		
37	Решение задач с использованием функций			1	
38	Решение задач с использованием функций			1	
39	Рекурсивные функции		1		
40	Рекурсивные функции			1	

#### Раздел 6. Строки - последовательности символов

41	Строки		1		
42	Строки			1	
43	Срезы строк		1		
44	Срезы строк			1	
45	Примеры решения задач со строками		1		
46	Решение задач со строками.			1	
47	Решение задач со строками.			1	

Решение  
практических  
задач

#### Раздел 7. Сложные типы данных

48	Списки		1		
49	Списки			1	
50	Срезы списков			1	
51	Срезы списков			1	
52	Решение задач со списками		1		
53	Решение задач со списками			1	
54	Решение задач со списками			1	
55	Матрицы		1		
56	Матрицы			1	
57	Матрицы			1	
58	Кортежи		1		
59	Кортежи			1	
60	Кортежи			1	
61	Введение в словари		1		
62	Введение в словари			1	
63	Множества в языке Python		1		
64	Множества в языке Python			1	

Тестирование,  
решение  
практических  
задач

#### Раздел 8. Стил программирования и отладка программ

65	Стиль программирования		1		Решение практических задач, презентация проекта, рефлексия.
66	Стиль программирования			1	
67	Стиль программирования			1	
68	Стиль программирования			1	
69	Отладка программ		1		
70	Отладка программ			1	
71	Отладка программ			1	
72	Зачет по курсу «Программирование на языке Python»			1	
	ВСЕГО	72	27	45	

### Содержание учебного плана

#### Вводное занятие

##### Раздел 1. Знакомство с языком Python

Теория: Знакомство с планом работы объединения, инструктаж по ТБ. 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

*Учащиеся должны знать/ понимать:*

- понятие программы;
- структура программы на Python;
- режимы работы с Python.

*Учащиеся должны уметь:*

- выполнить установку программы;
- выполнить простейшую программу в интерактивной среде;
- написать комментарии в программе.

##### Раздел 2. Переменные и выражения (8 часов)

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

Выражения. Операции. Порядок выполнения операций. Разделительные 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.

*Учащиеся должны знать/ понимать:*

- общую структуру программы;
- типы данных;
- целые, вещественные типы данных и операции над ними;

– оператор присваивания;

### Раздел 3. Условные предложения (7 часов)

Теория: Логический тип дан-

ных. 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 те-

ме "Условные операторы". Зачетная работа №1. "Составление программ с ветвлением".

Тест №3. "Условные операторы".

*Учащиеся должны знать/понимать:*

- назначение условного оператора;
- способ записи условного оператора;
- логический тип данных;
- логические операторы `or`, `and`, `not`;
- Учащиеся должны уметь:
- использовать условный оператор;
- создавать сложные условия с помощью логических операторов.

### Раздел 4. Циклы (10 часов)

Теория: Понятие цик-

ла. Тело цикла. Условия выполнения тела цикла. Оператор цикла с условием. Оператор цикла `while`. Бесконечные цик-

лы. Альтернативная ветка цикла `while`. Обновление переменной. Краткая форма записи обновления. Примеры использования циклов.

Оператор цикла с парамет-

ром `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"

*Учащиеся должны знать/понимать:*

- циклы с условием и их виды;
- правила записи циклов с условием;
- назначение и особенности использования цикла с параметром;
- формат записи цикла с параметром;
- примеры использования циклов различных типов.

*Учащиеся должны уметь:*

- определять вид цикла, наиболее удобный для решения поставленной задачи;
- использовать цикл с условием;
- определять целесообразность применения и использовать цикл с параметром для решения поставленной задачи.

### Раздел 5. Функции (10 часов)

Теория: Создание функций. Параметры и аргументы. Локальные и глобальные переменные.

Поток выполнения. Функции, возвращающие результат. Анонимные функ-

ции, 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

*Учащиеся должны знать/понимать:*

- понятие функции;
- способы описания функции;
- принципы структурного программирования;
- понятие локальных переменных подпрограмм;
- понятие формальных и фактических параметров подпрограмм;
- способ передачи параметров.

*Учащиеся должны уметь:*

- создавать и использовать функции;
- использовать механизм параметров для передачи значений.

#### Раздел 6. Строки- последовательности символов (7 часов)

Теория: Составной тип данных - строка. Доступ по индексу. Длина строки и отрицательные индексы. Преобразование типов. Применение цикла для обхода строки.

Срезы-

трок. 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.

*Учащиеся должны знать/понимать:*

- назначение строкового типа данных;
- операторы для работы со строками;
- процедуры и функции для работы со строками;
- операции со строками.

*Учащиеся должны уметь:*

- описывать строки;
- соединять строки;
- находить длину строки;
- вырезать часть строки;
- находить подстроку в строке;
- находить количество слов в строке.

#### Раздел 7. Сложные типы данных (17 часов)

Теория: Спис-

ки. Тип списка (list). Индексы. Обход списка. Проверка вхождения в список. Добавление в список. Суммирование или изменение спис-

ка. 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

*Учащиеся должны знать/понимать:*

- сложные типы данных;
- способ описания списка;
- способ доступа к элементам списка;
- способ описания кортежа;
- способ описания словаря;
- операции, выполняемые со списками, кортежами и словарями;
- понятие множества;
- способы описания множества;
- операторы работы с множествами.

*Учащиеся должны уметь:*

- описывать списки;
- вводить элементы списка;
- выводить элементы списка;
- выполнять поиск элемента в списке, поиск минимума и максимума, нахождение суммы элементов списка;
- использовать вложенные списки;
- приводить примеры использования вложенных списков (матриц);
- описывать множества;
- определять принадлежность элемента множеству;
- вводить элементы множества;
- выводить элементы множества.

#### Раздел 8. Стил ь программирования и отладка программ (8 часов)

Стил ь программирования. Отладка программ.

Зачет по курсу «Программирование на языке Python»

*Учащиеся должны знать/ понимать:*

- что такое стил ь программирования;
- правила именования объектов;
- основные рекомендации при написании программ.

*Учащиеся должны уметь:*

- определять вид ошибок и находить ошибки в программе.
- выполнять тестирование и отладку программ.

## 5. ОРГАНИЗАЦИОННО – ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

### 5.1. Материально-технические условия реализации программы

Реализация дополнительной общеобразовательной (общеразвивающей) программы обеспечена лабораторией «Программирование на Python».

Оборудование лаборатории и рабочих мест:

Рабочеемestoобучающегося	
Ноутбук	<p>Форм-фактор: ноутбук.</p> <p>Жесткая неотключаемая клавиатура. Русская раскладка клавиатуры.</p> <p>Диагональ экрана: не менее 15,6 дюйма. Разрешение экрана: не менее 1920 × 1080 пикселей. Количество ядер процессора: не менее 4.</p> <p>Количество потоков: не менее 8.</p> <p>Базовая тактовая частота процессора: не менее 1 ГГц. Максимальная тактовая частота процессора: не менее 2,5 ГГц. Кэш-память процессора: не менее 6 Мбайт.</p> <p>Объём установленной оперативной памяти: не менее 8 Гбайт.</p> <p>Объём поддерживаемой оперативной памяти (для возможности расширения): не менее 24 Гбайт.</p> <p>Объём накопителя SSD: не менее 240 Гбайт.</p> <p>Время автономной работы от батареи: не менее 6 часов.</p> <p>Вес ноутбука с установленным аккумулятором: не более 1,8 кг.</p> <p>Внешний интерфейс USB стандарта не ниже 3.0: не менее трёх свободных.</p> <p>Внешний интерфейс LAN (использование переходников не предусмотрено).</p> <p>Наличие модулей и интерфейсов (использование переходников не предусмотрено): VGA, HDMI.</p> <p>Беспроводная связь Wi-Fi: наличие с поддержкой стандарта IEEE 802.11n или современнее.</p> <p>Веб-камера. Манипулятор «мышь».</p> <p>Предустановленная операционная система с графическим пользовательским интерфейсом, обеспечивающая работу распространённых образовательных и общесистемных приложений</p>
Наушники	Тип: полноразмерные
Презентационное оборудование.	
Моноблочное интерактивное устройство	<p>Интерактивный моноблочный дисплей, диагональ экрана: не менее 65 дюймов,</p> <p>разрешение экрана: не менее 3840 × 2160 пикселей. Встроенная акустическая система.</p> <p>Количество одновременно распознаваемых касаний сенсорным экраном: не менее 20 касаний.</p> <p>Высота срабатывания сенсора экрана: не более 3 мм от поверхности экрана.</p> <p>Встроенные функции распознавания объектов касания (палец или без- батарейный стилус). Количество поддерживаемых без батарейных стилусов одновременно: не менее 2 шт.</p> <p>Возможность использования ладони в качестве инструмента стирания либо игнорирования касаний экрана ладонью.</p> <p>Интегрированный датчик освещённости для автоматической кор-</p>



	<p>рекции яркости подсветки. Наличие функции графического комментирования поверх произвольного изображения, в том числе от физически подключённого источника видеосигнала. Интегрированные функции вывода изображений с экранов мобильных устройств (на платформе распространённых ОС), а также с возможностью интерактивного взаимодействия (управления) с устройством-источником.</p> <p>Интегрированный в пользовательский интерфейс функционал просмотра и работы с файлами основных форматов с USB-накопителей или сетевого сервера.</p> <p>Поддержка встроенными средствами дистанционного управления рабочих параметров устройства через внешние системы.</p> <p>Предустановленная операционная система с графическим пользовательским интерфейсом, обеспечивающая работу распространённых образовательных и общесистемных приложений.</p> <p>Интегрированные средства, обеспечивающие следующий функционал: создание многостраничных уроков с использованием медиаконтента различных форматов, создание надписей и комментариев поверх запущенных приложений; распознавание фигур и рукописного текста (русский, английский языки); наличие инструментов рисования геометрических фигур и линий; встроенные функции: генератор случайных чисел, калькулятор, экранная клавиатура, таймер, редактор математических формул; электронные математические инструменты: циркуль, угольник, линейка, транспортир; режим «белой доски» с возможностью создания заметок, рисования, работы с таблицами и графиками; импорт файлов форматов: PDF, PPT</p>
Напольная мобильная стойка для интерактивных досок или универсальное настенное крепление.	Совместимость с моноблочным интерактивным устройством. Максимальный вес, выдерживаемый креплением: не менее 60 кг
<b>Дополнительное оборудование</b>	
Доска магнитно-маркерная настенная	Тип: полимерная, сухостираемая
Флипчарт магнитно-маркерный на треноге	Размер рабочей области: не менее 700 × 1000 мм. Тип опоры: тренога
Комплект кабелей и переходников.	Кабели, переходники для подключения и коммутации оборудования. Сетевой удлинитель для подключения оборудования к сети электропитания и др. (по выбору)
Учебная и методическая литература	Для реализации образовательных программ.
Комплект комплектующих и расходных материалов	Для реализации образовательных программ
<b>Мебель</b>	
Комплект мебели	Учебная мебель: столы для всех учеников, стулья/кресла для всех учеников, пуфы.

	Мебель для педагога: стол, стул (кресло). Системы хранения: тумбы, шкафы, стеллажи (по выбору)
--	---------------------------------------------------------------------------------------------------

## 5.2. Учебно-методическое обеспечение программы

1. Бэрри П. Изучаем программирование на Python. - М., 2017. - 624 с.
2. Буйначев С. К. Основы программирования на языке Python: учебное пособие. – Екатеринбург: Изд-во Урал. ун-та, 2014. – 91 с.
3. Бхаргава А. Грокаем алгоритмы: иллюстрированное пособие для программистов и любопытствующих. - СПб.: Питер, 2017. – 288 с.
4. Гэддис Т. Начинаем программировать на Python / пер. с англ. 4-е изд. – СПб.: БХВ-Петербург, 2019. - 768 с.
5. Мюллер Дж. Python для чайников. – СПб.: Диалектика, 2019. — 416 с.
6. Луридас П. Алгоритмы для начинающих: теория и практика для разработчика. - М.: Эксмо, 2018. – 608 с.
7. Лутц М. Изучаем Python, – СПб.: Символ Плюс, 2009. - 848 с.
8. Рафгарден Т. Совершенный алгоритм. Жадные алгоритмы и динамическое программирование. - СПб.: Питер, 2020. – 256 с.
9. Рейтц К., Шлюссер Т. Автостопом по Python. — СПб.: Питер, 2017.
10. Фёдоров Д. Ю. Программирование на языке высокого уровня Python: учебное пособие для прикладного бакалавриата. – М.: Издательство Юрайт, 2019. – 161 с.
11. Python 3 для начинающих: <https://pythonworld.ru/samouchitel-python>
12. Учебник по языку программирования Python (хэбраиндекс): <https://habr.com/ru/post/61905/>
13. Python для начинающих 2021 – уроки, задачи и тесты: <https://pythonru.com/uroki/python-dlja-nachinajushhih>

## 5.3. Кадровое обеспечение образовательной программы

Требования к квалификации педагогических кадров, обеспечивающих реализацию дополнительной общеобразовательной программы:

Высшее профессиональное образование, направленность (профиль) которого, как правило, соответствует направленности дополнительной общеобразовательной программы в рамках укрупненной группы 09.00.00 «Информатика и вычислительная техника»

Дополнительное профессиональное образование - профессиональная переподготовка, направленность которой соответствует направленности дополнительной общеобразовательной программы, осваиваемой обучающимися.

При отсутствии педагогического образования – дополнительное профессиональное педагогическое образование; дополнительная профессиональная программа может быть освоена после трудоустройства. Обязательно обучение по дополнительным профессиональным программам по профилю педагогической деятельности не реже одного раза в три года.

## 6. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ПРОГРАММЫ

Контроль и оценка результатов освоения программы осуществляется преподавателем в процессе проведения практических занятий, а также выполнения обучающимися индивидуальных заданий.

Объектами контроля и оценки являются: знания, умения, навыки по изучаемому курсу; уровень освоения программирования на языке Python; мастерство, техника исполнения работ; степень самостоятельности и уровень способностей.

Результаты обучения (освоенные умения, освоенные знания)	Формы и методы контроля и оценки
<b>Уметь</b>	
– Вырабатывать индивидуальную стратегию решения задач	Беседа, опрос наблюдение за деятельностью обучающегося. Проекты, конкурсы, выставки, фестивали, соревнования.
– Безопасно работать за компьютером в сети Интернет	Беседа, опрос, наблюдение за деятельностью обучающегося.
– Систематизировать и расширять знания в области программирования	Беседа, опрос, наблюдение за деятельностью обучающегося. Оценка результатов выполнения практических работ
– Работать с базами данных	Беседа, опрос, наблюдение за деятельностью обучающегося. Оценка результатов выполнения практических работ
– Разрабатывать собственные проекты	Творческие задания, контрольные работы, проекты.
– Выступать перед аудиторией, отстаивать свою точку зрения	Творческие задания, контрольные работы, проекты.
<b>Знать</b>	
– Структуру и содержание основ логики в работе программного кода	Беседа, опрос, наблюдение за деятельностью обучающегося. Оценка результатов выполнения практических работ.
– Язык программирования Python и все его особенности	Беседа, опрос,наблюдение за деятельностью обучающегося. Оценка результатов выполнения практических работ.
– Алгоритмическое мышление, необходимое для дальнейшей профессиональной деятельности	Беседа, опрос,наблюдение за деятельностью обучающегося. Оценка результатов выполнения практических работ. контрольные работы, проекты.
– Методы проектирования и разработки приложений	Беседа, опрос,наблюдение за деятельностью обучающегося. Оценка результатов выполнения практических работ.
– Структуру и составные элементы программного кода	Беседа, опрос,наблюдение за деятельностью обучающегося. Оценка результатов выполнения практических работ.

## 7. ОЦЕНОЧНЫЕ И МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

### 7.1. Оценочные материалы

### 7.1.1. Цель итоговой аттестации:

Итоговая аттестация по программе проводится в целях определения уровня теоретических знаний и практических умений по основам программирования на языке Python, основных приемов написания программ на современном языке программирования, развитие алгоритмического мышления обучающихся, творческих способностей.

### 7.1.2. Форма аттестации:

Обязательной формой аттестации по итогам освоения дополнительной образовательной (общеразвивающей) программы является итоговая работа, включающая в себя разработку и защиту индивидуального или группового проекта на языке программирования Python.

## 7.2. Методические материалы

Проект по программированию представляет собой продукт, результатом которого является программа для решения той или иной задачи. Особенностью является то, что одна и та же задача в зависимости от уровня проработки может быть решена как начинающим, так и опытным программистом.

При выполнении проекта по программированию обучающиеся имеют следующие возможности: выработать умение самостоятельно формулировать цели и задачи проекта, планировать свою деятельность, повысить уровень программирования на языке Python, получить умение представлять результаты своей деятельности.

Проект может разрабатываться индивидуально или группой обучающихся. Если задача достаточно сложная, то проект может быть разбит на подзадачи, подпроекты. Каждую подзадачу будут выполнять различные группы участников проекта. Например, одна группа занимается разработкой алгоритма, другая группа — непосредственно написанием и отладкой кода на языке Python, третья — подготовкой к презентации проекта.

План работы над проектом по программированию



В помощь участникам проекта можно предложить заполнить следующий учётный лист. Проект по программированию

Тема проекта:

Творческое название (при наличии):

Основополагающий вопрос:

Авторы:

- 1.
- 2.
- 3.

...

Предметная область:

Краткая аннотация:

Этапы выполнения проекта:

При подготовке к защите проекта обучающимся необходимо сделать презентацию и доклад, в котором отражаются основные этапы разработки программы, представлен алгоритм решения задачи, дан листинг программы, сформулированы основные результаты работы. Можно предложить в помощь обучающимся заполнить следующий чек-лист.

Аннотация.

Содержание.

Постановка задачи:

а) возможности использования программы; б) описание интерфейса.

Формализация алгоритма:

а) перечень подпрограмм (при наличии);

б) описание алгоритма (блок-схема или подробное словесное описание алгоритма).

Листинг программы (текст программы).

Тестовые примеры:

а) результаты работы;

б) скриншоты результатов работы.

Описание размещения.

Требования к программным и аппаратным средствам.

Для оценивания проекта могут быть разработаны специальные оценочные листы.

В таблице представлен пример оценочного листа.

Лист оценивания проекта

Критерий оценивания	1-я группа	2-я группа	...
Актуальность темы			
Соответствие содержания проекта заявленной теме			
Техническая сложность разработанной программы			
Оригинальность алгоритма			
Дизайн интерфейса			
Степень разработанности программы			
Применение программы для решения аналогичных задач			
Итоговое количество баллов			

Ниже приведены возможные темы исследовательских проектов обучающихся.

Конвертор чисел (перевод числа в n-ичную систему счисления).

- Шифровальщик текста (реализация шифра Цезаря).
- Компьютерный тест.
- Реализация игры «Камень, ножницы, бумага».
- Калькулятор для ипотеки.

