

**ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
СРЕДНЯЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ШКОЛА № 693  
НЕВСКОГО РАЙОНА САНКТ-ПЕТЕРБУРГА**

ПРИНЯТА  
Педагогическим советом  
ГБОУ школы № 693  
Невского района Санкт-Петербурга  
Протокол от \_\_\_\_\_ № \_\_\_\_\_

УТВЕРЖДЕНА  
Приказом директора  
ГБОУ школы № 693  
Невского района Санкт-Петербурга  
от \_\_\_\_\_ № \_\_\_\_\_

  
Н.А. Соколова  


**ДОПОЛНИТЕЛЬНАЯ  
ОБЩЕОБРАЗОВАТЕЛЬНАЯ  
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА  
«ШКОЛА ПРОГРАММИРОВАНИЯ»**

Возраст учащихся: 13-14 лет  
Срок реализации: 1 год

Разработчик:

Тихомирова Светлана Анатольевна,

Педагог дополнительного образования

## **ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

### **Направленность программы**

Дополнительная общеразвивающая программа «Школа программирования» имеет техническую направленность и предназначена для организации внеурочной деятельности по нескольким взаимосвязанным направлениям развития личности, таким как общеинтеллектуальное, общекультурное, социальное. Программа предлагает ее реализацию в факультативной или кружковой форме.

Программа ориентирована на обобщение и углубление знаний и умений по курсу информатики основной школы.

### **Уровень программы**

Данная дополнительная общеразвивающая программа ориентирована на один год обучения.

### **Актуальность программы**

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

### **Отличительные особенности данной программы**

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

Занятия начинаются с практического знакомства со средой программирования Python, далее идет непосредственное изучение синтаксических конструкций языка и отработка навыков применения элементов программирования при решении задач и создании игр. Каждая новая тема завершается практическими задачами, способствующими овладению методики программирования и изучению языка Python. На втором году обучения закрепляются полученные знания и навыки программирования, осваиваются новые методы, способы решения задач, рассматриваются более сложные

задачи, новые технологии программирования.

### **Адресат программы. Условия набора**

Данная дополнительная общеобразовательная программа предназначена для детей 13-15 лет. Набирается 1 группа. В коллектив принимаются все желающие. Набор производится, начиная с 1 сентября текущего года. Возможен дополнительный набор отдельных обучающихся в течении года.

### **Педагогическая целесообразность**

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

### **Цель программы**

Основной целью данного учебного курса является ознакомление слушателя с объектно-ориентированным языком программирования Python, с возможностями, синтаксисом языка, технологией и методами программирования в среде Python, обучение практическим навыкам программирования на языке Python для решения типовых задач математики и информатики, а также при разработке простейших игр.

### **Задачи программы**

Реализация поставленной цели предусматривает решение следующих задач:

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

- формирование навыков алгоритмического и логического мышления;
- формирование навыков грамотной разработки программ;
- формирование практических навыков решения прикладных задач;
- формирование практических навыков разработки игр.

### **Срок реализации программы**

Срок реализации программы «Школа программирования» составляет 1 год (72 недели, 144 часа). Период обучения: сентябрь - май.

### **Адресат программы**

Программа ориентирована на детей, склонных связать свою будущую жизнь с профессией программиста.

Набор производится по желанию учащихся и их родителей.

Возрастная категория обучающихся по программе от 13 до 14 лет.

### **Формы и режим занятия**

**Форма занятий** – групповая. Наполняемость группы – от 5 до 15 человек (набор осуществляется без предварительного отбора, по желанию и интересу учащегося).

**Режим занятий:** Обучающиеся занимаются 2 раза в неделю по 2 часа.

**Цель Программы:** Способствовать формированию творческой личности, обладающей информационными компетенциями, владеющей базовыми понятиями теории алгоритмов, умеющей разрабатывать эффективные алгоритмы и реализовывать их в виде программы, написанной на языке программирования Python.

#### **Задачи Программы**

##### ***Обучающие:***

- обучить языку программирования Python и созданию программ на его основе;
- научить создавать прикладное программное обеспечение;
- расширять кругозор обучающихся в области программирования;
- научить дизайнерскому оформлению созданного ПО.

##### ***Развивающие:***

- развивать память и внимание, познавательную и творческую активность;
- развивать творческие способности, эстетическое и эргономическое восприятие объектов труда;
- развивать логическое мышление.

##### ***Воспитательные:***

- прививать интерес к активному творческому самовыражению, культуре труда;
- воспитывать упорство в достижении желаемого результата;
- воспитывать эстетический вкус;
- воспитывать чувство взаимопомощи, доверия, коллективизма.

## Условия реализации программы

### **Материально-техническое обеспечение:**

- столы для компьютера;
- компьютерные стулья;
- шкафы для дидактических материалов, пособий;
- специальная и научно-популярная литература для педагога и учащихся;
- канцтовары;

### **Информационное обеспечение:**

- персональный компьютер (на каждого участника);
- интерактивная доска;
- видеоматериалы разной тематики по программе;
- оргтехника;
- выход в сеть Internet;

### **Аппаратное обеспечение:**

- Процессор не ниже Core2 Duo;
- Объем оперативной памяти не ниже 4 Гб DDR3;
- Дисковое пространство на менее 128 Гб;
- Монитор диагональю на мене 13”;

### **Программное обеспечение:**

- Операционная система Windows 10 Pro Профессиональная или выше;
- Интерпретатор Python версии 3.8 и выше;
- IDE JupyterLab/PyCharm;
- FoxitReader или другой просмотрщик PDF файлов;
- WinRAR;
- Пакет офисных программ;
- Растровый графический редактор;
- Любой браузер для интернет серфинга.

## Планируемые (ожидаемые) результаты

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

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

- умение самостоятельно планировать пути достижения цели, в том числе альтернативные, осознанно выбирать наиболее эффективные способы решения задач;

- умение соотносить свои действия с планируемыми результатами, осуществлять контроль всей деятельности в процессе достижения результата, корректировать свои действия в соответствии с изменяющейся ситуацией;
- умение оценивать правильность выполнения учебной задачи, собственные возможности ее решения;
- владение основами самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в учебной и познавательной деятельности;
- умение организовывать сотрудничество и совместную деятельность с учителем и сверстниками; работать индивидуально в группе: находить общее решение и разрешать конфликты на основе согласования позиций и учета интересов; формулировать, аргументировать и отстаивать свое мнение;
- формирование и развитие компетентности в области использования информационно-коммуникационных технологий.

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

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

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

- формирование знаний, умений и навыков при решении задач информатики и программирования разных видов;
- приобретение опыта использования информационных технологий в индивидуальной и коллективной учебной и познавательной, в том числе проектной деятельности.
- формирование информационной и алгоритмической культуры; формирование представления о компьютере как универсальном устройстве обработки информации; развитие основных навыков и умений использования компьютерных устройств;
- формирование представления об основных изучаемых понятиях: информация,

алгоритм, модель – и их свойства;

- развитие алгоритмического мышления, необходимого для профессиональной деятельности в современном обществе; развитие умений составить и записать алгоритм для конкретного исполнителя; формирование знаний об алгоритмических конструкциях, логических значениях и операциях; знакомство с одним из языков программирования и основными алгоритмическими структурами – линейной, условной и циклической;
- владение системой базовых знаний, отражающих вклад информатики в формирование современной научной картины мира;
- овладение понятием сложности алгоритма, знание основных алгоритмов обработки числовой и текстовой информации, алгоритмов поиска и сортировки;
- владение универсальным языком программирования высокого уровня Python, представлениями о базовых типах данных и структурах данных; умением использовать основные управляющие конструкции;
- владение навыками и опытом разработки программ в среде программирования Python, включая тестирование и отладку программ; владение элементарными навыками формализации прикладной задачи и документирования программ;
- формирование умения работать с библиотеками программ; получение опыта использования компьютерных средств представления и анализа данных.

В результате изучения дисциплины учащиеся должны знать:

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

– основы разработки простых игр в системе программирования Python.

**Учащиеся должны уметь:**

– записывать алгоритмические структуры на языке программирования Python;

– использовать Python для решения задач из области математики, физики;

– строить алгоритмы методом последовательного уточнения (сверху вниз), изображать эти алгоритмы в виде блок-схем;

– использовать основные алгоритмические приемы при решении математических задач;

– решать простые, сложные и нестандартные задачи;

– создавать простые игры;

– анализировать текст чужих программ, находить в них неточности, оптимизировать алгоритм, создавать собственные варианты решения.

### **Формы подведения итогов реализации программы**

Основной формой подведения итогов дополнительной общеразвивающей программы «Школа программирования» является решения задач (разработка и отладка программ на языке программирования Python).

#### **Учебный план 1 года обучения**

№ п/п	Название темы	Кол-во часов			Формы контроля
		Всего	Теория (в т. ч. контроль)	Практика (в т. ч. контроль)	
1	Инструктаж по ТБ	1	1	0	Текущий контроль
2	Основы программирования на языке Python.	29		16	Представление выполненной работы
3	Методы программирования на языке Python. Введение в олимпиадное программирование	42	14	15	Представление выполненной работы
4	Методы программирования на языке Python. Решение олимпиадных задач	48	24	24	Демонстрация и защита модели



5	Объектно-ориентированное программирование. Решение задач повышенной сложности	24	12	12	Демонстрация и защита модели
---	---	----	----	----	------------------------------

## **ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ**

Успешная реализация предлагаемой программы учебной дисциплины ориентирована на существующую информационно-образовательную среду школы. Информационно-образовательная среда образовательной организации включает комплекс информационно-образовательных ресурсов, в том числе цифровые образовательные ресурсы, совокупность технологических средств информационных и коммуникационных технологий: компьютеры и иное ИКТ-оборудование, коммуникационные каналы.

### **ОЦЕНОЧНЫЕ МАТЕРИАЛЫ**

Промежуточный контроль: решение задач

Итоговый контроль: учащимся предлагается самостоятельно выбрать тему творческого проекта и на основании темы разработать программу, пояснительную записку, презентацию.

### **МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ**

Программа разработана на основе концептуальных положений Стратегии инновационного развития Российской Федерации на период до 2020 года.

Методы обучения:

1. Словесный: объяснение нового материала; рассказ обзорный для раскрытия новой темы; беседы с учащимися в процессе изучения темы.
2. Наглядный: применение демонстрационного материала, наглядных пособий, презентаций по теме.
3. Практический: индивидуальная и совместная продуктивная деятельность, выполнение учащимися определенных заданий, решение задач.
4. Интерактивный: создание специальных заданий, моделирующих реальную жизненную ситуацию, из которой учащимся предлагается найти выход.

### **СПИСОК ЛИТЕРАТУРЫ, ИСПОЛЬЗУЕМЫЙ ПРИ НАПИСАНИИ ПРОГРАММЫ**

1. Майк МакГрат «Программирование на Python для начинающих» Эксмо, 2015.
2. Федоров Д. Ю. Основы программирования на примере языка Python. //Учебное пособие. – Санкт-Петербург: 2016.
3. Сэнд У., Сэнд К. «Hello World! Занимательное программирование на языке Python» - М.: – 2016.
4. Долинский М.С. Решение сложных и олимпиадных задач по программированию - Учебное пособие - М.: – 2006.
5. Россум Г., Дж. Дрейк Ф.Л., Откидач Д.С. Язык программирования Python. 2001.
6. Щерба А.В. Изучение языка программирования Python на основе задач УМК авторов И.А. Калинин и Н.Н. Самылкина. //Учебное пособие. –М.: МПГУ, 2015.
7. <http://informatics.mccme.ru/course/view.php?id=156>
8. [https://inf5.ru/podgotovka\\_k\\_olympiad/olym\\_zadachi\\_s\\_resheniyami.htm](https://inf5.ru/podgotovka_k_olympiad/olym_zadachi_s_resheniyami.htm)

9. <http://anngorg.ru/olimp/materials>
10. <http://kpolyakov.spb.ru/school/probook/python.htm>
11. <http://dist-olimpiada.krasnogorka.edusite.ru/p4aa1.html>

#### **СПИСОК ЛИТЕРАТУРЫ, РЕКОМЕНДУЕМЫЙ ДЕТЯМ**

1. Майк МакГрат «Программирование на Python для начинающих» Эксмо, 2015.
2. Федоров Д. Ю. Основы программирования на примере языка Python. //Учебное пособие. – Санкт-Петербург: 2016.
3. Сэнд У., Сэнд К. «Hello World! Занимательное программирование на языке Python» - М.: – 2016.
4. <http://kpolyakov.spb.ru/school/probook/python.htm>
5. <http://informatics.mccme.ru/course/view.php?id=156>

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
СРЕДНЯЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ШКОЛА № 693  
НЕВСКОГО РАЙОНА САНКТ-ПЕТЕРБУРГА

ПРИНЯТА  
Педагогическим советом  
ГБОУ школы № 693  
Невского района Санкт-Петербурга  
Протокол от \_\_\_\_\_ № \_\_\_\_\_

УТВЕРЖДЕНА  
Приказом директора  
ГБОУ школы № 693  
Невского района Санкт-Петербурга  
от \_\_\_\_\_ № \_\_\_\_\_  
Н.А. Соколова



РАБОЧАЯ ПРОГРАММА  
к дополнительной общеобразовательной общеразвивающей программе  
«Школа программирования»

Возраст учащихся 13-14 лет  
1 год обучения

Составитель: Тихомирова С.А.,  
педагог дополнительного образования

## **Особенности организации образовательного процесса первого года обучения.**

### **Цели программы «Школа программирования»:**

- Исследование, моделирование и конструирование решений.
- Вовлечение обучающихся в изучение предметов естественно-научного цикла с помощью практико-ориентированного подхода.
- Развитие базовых навыков программирования и алгоритмического мышления.
- Развитие навыков совместной работы, коммуникативных и презентационных компетенций, умения аргументированно представить свою точку зрения.
- Развитие критического мышления, навыков поиска решений поставленных задач. Использование научного подхода при изучении физических явлений и законов.

### **Данные цели реализуются через решение следующих задач:**

- Стимулировать мотивацию обучающихся к получению знаний, помогать формировать творческую личность.
- Способствовать развитию интереса к технике, конструированию, программированию, высоким технологиям, формировать навыки коллективного труда.
- Прививать навыки программирования через разработку программ в визуальной среде программирования, развивать алгоритмическое мышление.

### **В основу программы положены:**

- Тематический принцип планирования учебного материала, что отвечает задачам нравственного, трудового и эстетического воспитания обучающихся, учитывает их интересы, их возрастные особенности;
- Обучение в порядке усложнения;
- Взаимосвязь развития творческих и познавательных способностей обучающихся;
- Построение занятий таким образом, чтобы обучающиеся имели возможность сменить типы и ритмы работы;
- Создание на занятиях эмоционально-комфортной среды.

## **Содержание дополнительной общеразвивающей программы «Школа программирования» 1 года обучения.**

### **Раздел 1. Основы программирования на языке Python.**

Тема 1.1. История языков программирования. Введение в язык программирования Python. Среда программирования Python. Установка программы. Теория (1 ч.)

Тема 1.2. Типы данных и функции вывода. Определение переменной. Переменные и арифметические выражения. Теория (1 ч.). Практика (1 ч.) Тренировочные задания.

Тема 1.3. Чтение данных. Операции над строками. Примеры решения задач. Теория (1 ч.). Практика (1 ч.) Тренировочные задания.

Тема 1.4. Отработка навыков решения простейших задач. Практика (4 ч.) Форма подведения итогов: практические задания.

Тема 1.5. Логический тип данных и операции. Примеры использования логических выражений. Теория (1 ч.).

Тема 1.6. Условный оператор. Вложенный условный оператор. Примеры решения задач. Теория (1 ч.). Практика (1 ч.) Тренировочные задания.

Тема 1.7. Отработка навыков решения простейших задач. Практика (4 ч.) Форма подведения итогов: практические задания.

Тема 1.8. Цикл WHILE. Примеры решения задач. Теория (1 ч.). Практика (1 ч.)

Тренировочные задания.

Тема 1.9. Подсчет суммы и оператор CONTINUE. Примеры решения задач. Теория (1 ч.). Практика (1 ч.) Тренировочные задания.

Тема 1.10. Отработка навыков решения простейших задач. Практика (4 ч.) Форма подведения итогов: практические задания.

Тема 1.11. Вещественные числа. Основы работы с вещественными числами. Округление вещественных чисел. Примеры решения задач. Теория (1 ч.). Практика (1 ч.) Тренировочные задания.

Тема 1.12. Отработка навыков решения простейших задач. Практика (4 ч.) Форма подведения итогов: практические задания.

## **Раздел 2. Методы программирования на языке Python. Введение в олимпиадное программирование.**

Тема 2.1. Срезы строк. Использование срезов. Метод FIND. Примеры решения задач. Теория (1 ч.). Практика (1 ч.) Тренировочные задания.

Тема 2.2. Отработка навыков решения простейших задач.

Тема 2.3. Методы RFIND, REPLACE и COUNT. Примеры решения задач.

Тема 2.4. Отработка навыков решения простейших задач. Практика (2 ч.) Форма подведения итогов: практические задания.

Тема 2.5. Решение несложных олимпиадных задач. Практика (4 ч.) Форма подведения итогов: практические задания.

Тема 2.6. Функции. Использование функций. Примеры решения задач. Теория (1 ч.). Практика (1 ч.) Тренировочные задания.

Тема 2.7. Возврат значений. Локальные и глобальные переменные. Примеры решения задач. Теория (1 ч.). Практика (1 ч.) Тренировочные задания.

Тема 2.8. Отработка навыков решения задач. Практика (4 ч.) Форма подведения итогов: практические задания.

Тема 2.9. Рекурсия. Использование рекурсии. Примеры решения задач. Теория (1 ч.). Практика (1 ч.) Тренировочные задания.

Тема 2.10. Отработка навыков решения задач. Практика (4 ч.) Форма подведения итогов: практические задания.

Тема 2.11. Кортежи. Функция RANGE, цикл FOR. Примеры решения задач. Теория (1 ч.). Практика (1 ч.) Тренировочные задания.

Тема 2.12. Отработка навыков решения задач. Практика (4 ч.) Форма подведения итогов: практические задания.

Тема 2.13. Списки. Метод SPLIT и JOIN. Примеры решения задач. Теория (1 ч.). Практика (1 ч.) Тренировочные задания.

Тема 2.14. Отработка навыков решения задач. Практика (2 ч.) Форма подведения итогов: практические задания.

Тема 2.15. Решение олимпиадных задач. Практика (4 ч.) Форма подведения итогов: практические задания.

## **Тема 3. Методы программирования на языке Python. Решение олимпиадных задач.**

Тема 3.1. Повторение. Основные конструкции языка программирования Python. Типы данных. Арифметические выражения. Условный оператор. Циклы. Решение задач. Теория (2 ч.). Практика (2 ч.) Тренировочные задания.

Тема 3.2. Срезы строк. Кортежи. Методы. Функции и рекурсия в Python. Решение задач. Теория (2 ч.). Практика (2 ч.) Тренировочные задания.

Тема 3.3. Списки. Методы работы со списками. Обработка списка. Примеры решения задач. Теория (1 ч.). Практика (1 ч.) Тренировочные задания.

Тема 3.4. Отработка навыков решения задач. Практика (4 ч.) Форма подведения итогов: практические задания.

Тема 3.5. Сортировка. Сравнение списков и кортежей. Именованный параметр KEY.

Примеры решения задач. Теория (1 ч.). Практика (1 ч.) Тренировочные задания.

Тема 3.6. Отработка навыков решения задач. Практика (4 ч.) Форма подведения итогов: практические задания.

Тема 3.7. Структуры в Python. Лямбда-функции. Именованные параметры. Чтение из файла. Теория (1 ч.). Практика (1 ч.) Тренировочные задания.

Тема 3.8. Сортировка подсчетом. Примеры решения задач. Теория (1 ч.). Практика (1 ч.) Тренировочные задания.

Тема 3.9. Отработка навыков решения задач. Практика (4 ч.) Форма подведения итогов: практические задания.

Тема 3.10. Множества и хеш-функции. Создание и работа с множествами. Теория (1 ч.). Практика (1 ч.) Тренировочные задания.

Тема 3.11. Отработка навыков решения задач. Практика (4 ч.) Форма подведения итогов: практические задания.

Тема 3.12. Решение олимпиадных задач. Практика (4 ч.) Форма подведения итогов: практические задания.

Тема 3.13. Словари. Методы строк. Примеры решения сложных задач. Теория (2 ч.). Практика (2 ч.) Тренировочные задания.

Тема 3.14. Отработка навыков решения сложных задач. Практика (6 ч.) Форма подведения итогов: практические задания.

**Тема 4. Объектно-ориентированное программирование. Решение задач повышенной сложности.**

Тема 4.1. Парадигмы программирования и функциональное программирование. Встроенные функции для работы с последовательностями. Примеры решения задач в функциональном стиле. Теория (1 ч.). Практика (1 ч.) Тренировочные задания.

Тема 4.2. Итераторы и генераторы. Генерация комбинаторных объектов ITERTOOLS. Функции PARTIAL, REDUCE, ACCUMULATE. Примеры решения задач. Теория (1 ч.). Практика (1 ч.) Тренировочные задания.

Тема 4.3. Отработка навыков решения сложных задач. Практика (4 ч.) Форма подведения итогов: практические задания.

Тема 4.4. Объектно-ориентированное программирование. Комплексные числа. Инкапсуляция и конструкторы. Примеры решения задач. Теория (1 ч.). Практика (1 ч.) Тренировочные задания.

Тема 4.5. Классы объектов. Обработка ошибок. Наследование и полиморфизм. Примеры решения задач. Теория (1 ч.). Практика (1 ч.) Тренировочные задания.

Тема 4.6. Решение задач повышенной сложности. Практика (4 ч.) Форма подведения итогов: практические задания.

Тема 4.7. Решение сложных олимпиадных задач. Практика (6 ч.) Форма подведения итогов: практические задания.

Тема 4.8. Заключительное занятие Практика (2 ч.) Форма подведения итогов: зачетное задание.

### **Планируемые результаты**

В процессе освоения содержания программы обучающиеся должны **знать/понимать**:

- правила безопасного поведения и гигиены при работе с конструктором и компьютером;
- основных понятия, основные технические термины, связанные с процессами конструирования и программирования на языке Python;
- знание основных алгоритмов в программировании;
- знание операторов, виды информации и способы её представления.

**Уметь:**

- составлять алгоритмы по задачам;
- правильно писать программы;
- уметь самостоятельно решать технические задачи.
- вести индивидуальные и групповые работы.

**Использовать/применять** приобретенные знания и умения в практической деятельности и повседневной жизни для:

- поиска, преобразования, хранения и применения информации (в том числе с использованием компьютера) для решения различных задач;
- соблюдения правил личной гигиены и безопасности приёмов работы со средствами информационных и коммуникационных технологий.

**ПРИНЯТ**

Педагогическим советом  
ГБОУ школа №693  
Невского района г.Санкт-Петербурга  
Протокол № \_\_\_\_\_  
«\_\_» \_\_\_\_\_ 202\_г.

**УТВЕРЖДЕН**

Приказом от «\_\_» \_\_\_\_\_ 202\_г.  
Директор ГБОУ Школа №693  
Невского района г.Санкт-Петербурга  
Протокол № \_\_\_\_\_  
\_\_\_\_\_ Соколова Н.А.

**Календарно-тематический план  
рабочей программы «Школа программирования»  
первый год обучения, количество часов 144 часа  
На 2020-2021 учебный год**

№ занятия	Планируемая дата	Фактическая дата	Раздел	Тема	Кол-во часов			Формы/Способы контроля
					Всего	Теория	Практика	
			1	Основы программирования на языке Python	30	8	22	
1			1.1.	История языков программирования. Введение в язык программирования Python. Среда программирования Python. Установка программы	1	1		
2			1.2.	Типы данных и функции вывода. Определение переменной. Переменные и арифметические выражения.	2	1	1	
3			1.3.	Чтение данных. Операции над строками. Примеры решения задач	2	1	1	
4			1.4.	Отработка навыков решения простейших задач.	4		4	Практические задания
5			1.5.	Логический тип данных и операции. Примеры использования логических выражений.	1	1		
6			1.6.	Условный оператор. Вложенный условный оператор. Примеры решения	2	1	1	



№ занятия	Планируемая дата	Фактическая дата	Раздел	Тема	Кол-во часов			Формы/Способы контроля
					Всего	Теория	Практика	
				задач.				
7			1.7.	Отработка навыков решения простейших задач.	4		4	Практические задания
8			1.8.	Цикл WHILE. Примеры решения задач.	2	1	1	
9			1.9.	Подсчет суммы и оператор CONTINUE. Примеры решения задач.	2	1	1	
10			1.10.	Отработка навыков решения простейших задач	4		4	Практические задания
11			1.11.	Вещественные числа. Основы работы с вещественными числами. Округление вещественных чисел. Примеры решения задач.	2	1	1	
12			1.12.	Отработка навыков решения простейших задач.	4		4	Практические задания
			2	<b>Методы программирования на языке Python. Введение в олимпиадное программирование.</b>	42	7	35	
13			2.1	Срезы строк. Использование срезов. Метод FIND. Примеры решения задач.	2	1	1	
14			2.2.	Отработка навыков решения простейших задач.	4		4	Практические задания
15			2.3.	Методы RFIND, REPLACE и COUNT. Примеры решения задач.	2	1	1	
16			2.4.	Отработка навыков решения простейших задач.	2		2	Практические задания
17			2.5.	Решение несложных олимпиадных задач	4		4	Практические задания
18			2.6.	Функции. Использование функций. Примеры решения задач.	2	1	1	
19			2.7.	Возврат значений.	2	1	1	

№ занятия	Планируемая дата	Фактическая дата	Раздел	Тема	Кол-во часов			Формы/Способы контроля
					Всего	Теория	Практика	
				Локальные и глобальные переменные. Примеры решения задач.				
20			2.8.	Отработка навыков решения задач.	4		4	Практические задания
21			2.9.	Рекурсия. Использование рекурсии. Примеры решения задач.	2	1	1	
22			2.10.	Отработка навыков решения задач	4		4	Практические задания
23			2.11.	Кортежи. Функция RANGE, цикл FOR. Примеры решения задач.	2	1	1	
24			2.12.	Отработка навыков решения задач.	4		4	Практические задания
25			2.13.	Списки. Метод SPLIT и JOIN. Примеры решения задач.	2	1	1	
26			2.14.	Отработка навыков решения задач.	2		2	Практические задания
27			2.15.	Решение олимпиадных задач.	4		4	Практические задания
			3.	<b>Методы программирования на языке Python. Решение олимпиадных задач.</b>	48	11	37	
28			3.1.	Повторение. Основные конструкции языка программирования Python. Типы данных. Арифметические выражения. Условный оператор. Циклы. Решение задач.	4	2	2	
29			3.2.	Срезы строк. Кортежи. Методы. Функции и рекурсия в Python. Решение задач.	4	2	2	
30			3.3.	Списки. Методы работы со	2	1	1	

№ занятия	Планируемая дата	Фактическая дата	Раздел	Тема	Кол-во часов			Формы/Способы контроля
					Всего	Теория	Практика	
				списками. Обработка списка. Примеры решения задач.				
31			3.4.	Отработка навыков решения задач.	4		4	Практические задания
32			3.5.	Сортировка. Сравнение списков и кортежей. Именованный параметр KEY. Примеры решения задач.	2	1	1	
33			3.6.	Отработка навыков решения задач.	4		4	Практические задания
34			3.7.	Структуры в Python. Лямбда-функции. Именованные параметры. Чтение из файла.	2	1	1	
35			3.8.	Сортировка подсчетом. Примеры решения задач.	2	1	1	
36			3.9.	Отработка навыков решения задач.	4		4	Практические задания
37			3.10.	Множества и хеш-функции. Создание и работа с множествами.	2	1	1	
38			3.11.	Отработка навыков решения задач.	4		4	Практические задания
39			3.12.	Решение олимпиадных задач.	4		4	Практические задания
40			3.13.	Словари. Методы строк. Примеры решения сложных задач.	4	2	2	
41			3.14.	Отработка навыков решения сложных задач.	6		6	Практические задания
			4.	<b>Объектно-ориентированное программирование. Решение задач повышенной сложности.</b>	24	4	20	
42			4.1.	Парадигмы программирования и функциональное программирование. Встроенные функции для	2	1	1	

№ занятия	Планируемая дата	Фактическая дата	Раздел	Тема	Кол-во часов			Формы/Способы контроля
					Всего	Теория	Практика	
				работы с последовательностями. Примеры решения задач в функциональном стиле.				
43			4.2.	Итераторы и генераторы. Генерация комбинаторных объектов ITERTOOLS. Функции PARTIAL, REDUCE, ACCUMULATE. Примеры решения задач.	2	1	1	
44			4.3.	Отработка навыков решения сложных задач.	4		4	Практические задания
45			4.5.	Объектно-ориентированное программирование. Комплексные числа. Инкапсуляция и конструкторы. Примеры решения задач.	2	1	1	
46			4.6.	Классы объектов. Обработка ошибок. Наследование и полиморфизм. Примеры решения задач.	2	1	1	
47			4.7.	Решение задач повышенной сложности.	4		4	Практические задания
48			4.8.	Решение сложных олимпиадных задач.	6		6	Практические задания
49			4.9.	Заключительное занятие	2		2	Практические задания
				Итого:	144	30	114	

**ПРИНЯТ**

Педагогическим советом  
ГБОУ школа №693  
Невского района г.Санкт-Петербурга  
Протокол № \_\_\_\_\_  
«\_\_» \_\_\_\_\_ 202\_г.

**УТВЕРЖДЕН**

Приказом от «\_\_» \_\_\_\_\_ 202\_г.  
Директор ГБОУ Школа №693  
Невского района г.Санкт-Петербурга  
Протокол № \_\_\_\_\_  
\_\_\_\_\_ Соколова Н.А.

**КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК**

На 2020-2021 учебный год

Дополнительная общеобразовательная общеразвивающая программа  
«Школа программирования»

Год обучения, номер группы	Дата обучения по программе	Дата окончания обучения по программе	Количество учебных недель	Количество учебных дней	Количество учебных часов	Режим занятий
1 год, группа №1	01.09.2020	31.05.2021	36	72	144	2 часа 2 раза в неделю

**Лист корректировки учебно-тематического плана**

<b>№ п/п</b>	<b>Дата внесения изменений</b>	<b>Содержание внесенных изменений</b>	<b>Обоснование внесения корректировки</b>	<b>Результат внесенных изменений</b>