

Управление образования администрации  
Киселевского городского округа  
Муниципальное бюджетное общеобразовательное учреждение  
«Средняя образовательная школа №3»

**ПРИНЯТА** на заседании  
педагогического совета ШКОЛЫ 3  
от «\_30\_»\_августа\_\_\_\_\_ 2024\_\_ г.  
Протокол № \_\_1\_\_\_\_\_

**УТВЕРЖДАЮ:**  
Директор ШКОЛЫ 3  
\_\_\_\_\_ О.И. Филиппова  
Приказ № \_59\_  
от «\_18\_»\_сентября\_ 2024\_\_ г.



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

**Стартовый уровень**

**Возраст обучающихся: 12-17 лет**

**Срок реализации: 1 год**

*Разработчик программы:*  
**Андреяшина Ольга Викторовна,**  
учитель информатики

## СОДЕРЖАНИЕ

<b>РАЗДЕЛ 1. КОМПЛЕКС ОСНОВНЫХ ХАРАКТЕРИСТИК ПРОГРАММЫ.....</b>	<b>...3</b>
1.1. Пояснительная записка .....	...3
1.2. Цель и задачи программы .....	...8
1.3. Учебно-тематический план и содержание программы .....	...9
1.4. Планируемые результаты .....	...17
<b>РАЗДЕЛ 2. КОМПЛЕКС ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИХ УСЛОВИЙ .....</b>	<b>...20</b>
2.1. Календарный учебный график .....	...20
2.2. Условия реализации программы .....	...20
2.3. Формы контроля/аттестации.....	...22
2.4. Оценочные материалы .....	...23
2.5. Методические материалы .....	...23
2.6. Список литературы .....	...24
<b>ПРИЛОЖЕНИЯ.....</b>	<b>...25</b>

# КОМПЛЕКС ОСНОВНЫХ ХАРАКТЕРИСТИК ПРОГРАММЫ

## 1.1. Пояснительная записка

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

- деятельности центра цифрового образования детей «IT-куб» для реализации дополнительных общеразвивающих программ федерального проекта «Цифровая образовательная среда»;
- выполнения *муниципального социального заказа* на услуги дополнительного образования на территории Киселевского городского округа.

**Нормативно-правовое обеспечение.** В настоящее время содержание, роль, назначение и условия реализации программ дополнительного образования закреплены в следующих нормативных документах:

- Федеральный Закон Российской Федерации от 29.12.2012 № 273 «Об образовании в Российской Федерации» (с изменениями и дополнениями);
- изменения в Федеральном законе «Об образовании в Российской Федерации» 273-ФЗ в части определения содержания воспитания в образовательном процессе с 01.09.2020 г., наименования и определения финансового обеспечения реализации образовательной программы, определенные в соответствии с бюджетным законодательством Российской Федерации и настоящим Федеральным законом, с 14.07.2023 г.;
- Указ Президента Российской Федерации «О национальных целях развития Российской Федерации на период до 2030 года», определяющего одной из национальных целей развития Российской Федерации предоставление возможности для самореализации и развития талантов;
- Указ Президента РФ от 07.05.2012 № 599 «О мерах по реализации государственной политики в области образования и науки»;
- Распоряжение Министерства просвещения РФ от 10 ноября 2021 г. № ТВ-1984/04 «Об утверждении методических рекомендаций по созданию и функционированию центров цифрового образования «IT-куб» в рамках региональных проектов, обеспечивающих достижение целей, показателей и результата федерального проекта «Цифровая образовательная среда» национального проекта «Образование»;
- Концепция развития дополнительного образования детей в РФ (Распоряжение Правительства РФ от 31.03.2022 г. № 678-р);
- Приказ Министерства просвещения РФ от 27.07.2022 № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по ДООП»;
- Приказ Министерства науки и высшего образования РФ и Министерства просвещения РФ от 30 июня 2020 г. № 845/369 «Об утверждении Порядка зачета организацией, осуществляющей образовательную деятельность, результатов освоения обучающимися учебных предметов, курсов, дисциплин»;

- плин (модулей), практики, дополнительных образовательных программ в других организациях, осуществляющих образовательную деятельность»;
- Национальный проект «Образование» (утвержден президиумом Совета при Президенте РФ по стратегическому развитию и национальным проектам (протокол от 03.09.2018 №10);
  - Стратегия развития информационного общества в Российской Федерации, утв. Указом Президента РФ от 09.05.2017 № 203 «О Стратегии развития информационного общества в Российской Федерации на 2017-2030 годы»;
  - Государственная программа РФ «Развитие образования» на 2018-2025 гг. (Постановление Правительства РФ от 26 декабря 2017 года № 1642);
  - Письмо Минобрнауки России от 18.11.2015 №09-3242. «Методические рекомендации по проектированию дополнительных общеразвивающих программ» (включая разноуровневые программы);
  - Постановление Правительства Российской Федерации от 11.10.2023 № 1678 «Об утверждении Правил применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ»;
  - Письмо Министерства Просвещения РФ от 29 сентября 2023 г. № АБ-3935/06 «Методические рекомендации по формированию механизмов обновления содержания, методов и технологий обучения в системе дополнительного образования детей, направленных на повышение качества дополнительного образования детей, в том числе включение компонентов, обеспечивающих формирование функциональной грамотности и компетентностей, связанных с эмоциональным, физическим, интеллектуальным, духовным развитием человека, значимых для вхождения Российской Федерации в число десяти ведущих стран мира по качеству общего образования, для реализации приоритетных направлений научно-технологического и культурного развития страны»;
  - Методические рекомендации по созданию и функционированию центров цифрового образования «IT-куб» (утверждены распоряжением Министерства просвещения Российской Федерации от 12 января 2021 г. № Р-5);
  - Постановление Государственного санитарного врача РФ от 28.09.2020 г. СП 2.4. 3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи»;
  - Постановление Государственного санитарного врача РФ от 28 января 2021 года № 2 СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания»;
  - Распоряжение Правительства Российской Федерации от 29 мая 2015 года № 996-р «Стратегия развития воспитания в Российской Федерации на период до 2025 года».
  - Приказ Министерства просвещения Российской Федерации от 13.03.2019 № 114 «Об утверждении показателей, характеризующих общие критерии

оценки качества условий осуществления образовательной деятельности организациями, осуществляющими образовательную деятельность по основным общеобразовательным программам, образовательным программам среднего профессионального образования, основным программам профессионального обучения, дополнительным общеобразовательным программам»;

- Приказ Министерства Кузбасса от 13 января 2023 г. № 102 «Правила персонифицированного финансирования дополнительного образования детей в Кемеровской области - Кузбассе»;
- Муниципальное «Положение о разработке, структуре и порядке утверждения дополнительной общеобразовательной общеразвивающей программы» (утв. УО 30.04.2023 г.);
- Устав Муниципального бюджетного общеобразовательного учреждения Киселевского городского округа «Средняя общеобразовательная школа 3» (ШКОЛА 3).

**Актуальность программы.** В 20-е г. XXI века в России, в том числе и в Кузбассе, возрастает потребность в специалистах с высоким уровнем владения информационными компетенциями, отвечающих *социальному заказу общества* по подготовке квалифицированных кадров в области программирования, а также обладающих высоким интересом к IT-сфере.

Одной из составляющих информационной компетентности является владение языком программирования, который отвечает современным требованиям к написанию программ, служит основой для дальнейшего развития и совершенствования программистских компетенций.

В настоящее время Python является очень востребованным языком. Он отлично подходит для знакомства с различными современными парадигмами программирования и активно применяется в самых разных областях от разработки веб-приложений до машинного обучения.

В связи с этим была разработана дополнительная общеобразовательная общеразвивающая программа «Программирование на языке Python», содержание которой позволяет обеспечить начальную подготовку учащихся в области программирования и формирует положительную мотивацию к языкам программирования;

**Отличительные особенности программы.** Программа «Основы программирования на языке Python» - *модифицированная*. Составлена с использованием элементов авторских программ и курсов:

- «Основы программирования», автор С.М. Окулов (г. Москва, 2012 г.);
- «Программирование на языке Python» педагога дополнительного образования Ефимова К. Н. (г. Калуга, 2019 г.);
- «Программирование аркадных игр на языке Python» педагога дополнительного образования Прядкиной Елены Владимировны (г. Москва, 2020 г.).

Особую роль в программе «Основы программирования на языке Python» отводится заданиям и упражнениям, направленным на формирование мысли-

тельных и психических процессов учащихся (внимание, память, логика), освоение приёмов умственных действий, самостоятельного нахождения и составления алгоритмов решения задач, умения строить модели, чётко и лаконично реализовывать этапы решения задач. Использование этих возможностей для формирования предметных и метапредметных результатов учащихся особенно важно, т.к. именно они активизируют процесс индивидуально-личностного становления учащихся.

Ключевой особенностью программы является его направленность на формирование у учащихся навыков поиска собственного решения поставленной задачи, составления алгоритма решения и его реализации с помощью средств программирования.

Основное количество часов в программе отводится практическому написанию учащимися компьютерных программ. При этом в результате каждый учащийся реализует индивидуальный проект. Поэтому продукт, полученный в результате освоения программы, имеет прикладной характер и может быть использован по назначению. Таким образом, программа имеет *воспитательную* составляющую - потенциальные предпосылки для развития коммуникативных и речевых действий в силу их действительно универсального, т.е. максимально обобщенного, характера.

Содержание данной программы служит средством внутри профильной специализации в области новых информационных технологий, что способствует созданию дополнительных условий для проявления индивидуальных образовательных интересов учащихся, их дальнейшей профессиональной ориентации. Осваивая данную программу, учащиеся овладеют навыками, которые будут востребованы в ближайшие десятилетия в различных специальностях, многие из которых включены в «Атлас профессий будущего».

***Уровень освоения программы – стартовый*** - позволяет обеспечить начальную подготовку учащихся в области программирования и формирует положительную мотивацию к языкам программирования и базовые конструкции языка Python.

***Адресат программы:*** учащиеся 12-17 лет.

Дети этого возраста отличаются внутренней уравновешенностью, стремлением к активной практической деятельности, поэтому основной формой проведения занятий выбраны практические занятия. Ребят также увлекает совместная, коллективная деятельность, так как резко возрастает значение коллектива, общественного мнения, отношений со сверстниками, оценки поступков и действий ребёнка со стороны не только старших, но и сверстников. Ребёнок стремится завоевать в их глазах авторитет, занять достойное место в коллективе.

Также следует отметить, что дети данной возрастной группы характеризуются такими психическими процессами, как изменение структуры личности и возникновение интереса к ней, развитие абстрактных форм мышления, становление более осознанного и целенаправленного характера деятельности, проявление стремления к самостоятельности и независимости, формирование самооценки. Эти процессы позволяют положить начало созданию начального про-

фессионального самоопределения учащихся.

**Наполняемость групп и особенности набора учащихся.** Наполняемость групп - до 15 человек. Группы формируются из учащихся разного возраста на добровольной внеконкурсной основе. Набор осуществляется на основании результатов собеседования, просмотра работ, наличия базовых знаний в области программирования.

*Минимальное количество групп для набора – 3 группы.*

Набор учащихся осуществляется на добровольной основе на основании письменного заявления родителей (законных представителей) или самого учащегося, достигшего 14 лет.

**Объем и срок освоения программы.** Срок освоения программы – 1 год. Общее количество часов в год – 144 часа.

**Режим занятий, периодичность и продолжительность.** Занятия проводятся 2 раза в неделю по 2 часа. Общее количество часов в неделю – 4 часа. Перерыв между занятиями групп не менее 10 минут.

В каникулярный период допускается изменение форм занятий, проведение воспитательных мероприятий, проведение и участие соревнований. А также возможно участие учащихся творческого объединения в общешкольных мероприятиях.

**Форма обучения** - очная. В условиях перехода на дистанционное обучение программа может быть реализована с использованием дистанционных образовательных технологий.

**Особенности организации образовательного процесса.** Программа реализуется в условиях общеобразовательного учреждения в рамках центра цифрового образования детей «IT-куб» для реализации дополнительных общеразвивающих программ федерального проекта «Цифровая образовательная среда».

Группы формируются из учащихся разного возраста, имеющих равные стартовые знания и умения в области информационных технологий. Содержание и условия реализации образовательной программы соответствуют возрастным и индивидуальным особенностям учащихся, адаптируется в соответствии с их возрастом и первоначальными навыками.

Занятия проводятся в соответствии с учебно-тематическим планом. Имеют практикоориентированный характер. Практикоориентированность программы выражена в подборе интерактивных и нестандартных форм занятий, способствующих формированию основных компетенций (информационных, коммуникативных, компетенций личного развития) и функциональной грамотности.

Особенностью организации образовательного процесса является использование нового современного оборудования для занятий и IT-технологий.

Формы организации образовательной деятельности: фронтальная – со всей группой; индивидуальная – самостоятельная работа учащегося над проектом, под руководством и с консультацией педагога; групповая – когда над одним проектом работают несколько человек. Важным условием совместных занятий

является получение навыка работы в коллективе и развитие самостоятельного технического творчества. Поэтому в программу включены практические занятия соревновательного характера, которые позволяют каждому проявить себя и найти своё место в детском коллективе.

В процессе реализации программы используются различные виды занятий: игра, квест, соревнование, квиз, викторины, интеллектуальные вопросы, конкурсы, и другие. Структура каждого занятия зависит от конкретной темы и решаемых задач.

Данная программа интегрирует в себе *воспитательный и образовательный компоненты*. Содержание программы предполагает формирование у учащихся моральных и нравственных установок и традиционных российских ценностных ориентиров, основанных на идее единства, коллективизма, солидарности, социального оптимизма и ценности труда. *Воспитательная работа* осуществляется как в процессе занятий, так и вне занятий в форме бесед, мастер-классов, экскурсий, участия в различных праздниках, творческих мероприятиях школы.

В условиях перехода на *дистанционное обучение* программа может быть реализована с использованием дистанционных образовательных технологий. В этом случае образовательный процесс организуется в форме видеозанятий, с использованием VK Мессенджера. Видеозанятия могут отправляться учащимся по электронной почте. При необходимости педагогом проводятся индивидуальные консультации с учащимися с использованием VK Мессенджера. Контроль выполнения заданий фиксируется посредством фотоотчетов, видеоотчетов, размещаемых учащимися и (или родителями) по итогам занятия в группе VK Мессенджера или на электронную почту педагога (по согласованию).

***Педагогическая целесообразность.*** На занятиях по данной программе учащиеся, научившись программировать на языке Python, получают мощный и удобный инструмент для решения как учебных, так и прикладных задач. Вместе с тем чистота и ясность конструкций этого языка программирования позволит им позже с лёгкостью выучить любой другой язык. Кроме того, знания и умения, приобретённые в результате освоения данной программы, могут быть использованы учащимися при сдаче ЕГЭ, при участии в олимпиадах по программированию, при решении задач по физике, химии, биологии, лингвистике и другим наукам, а также они являются фундаментом для дальнейшего совершенствования мастерства программирования и профессионального самоопределения.

## 1.2. Цель и задачи программы

***Цель программы:*** освоение учащимися основ программирования на языке Python.



### **Задачи программы:**

#### *Образовательные:*

- познакомить учащихся с возможностями, синтаксисом, технологией языка Python, с принципами и методами структурного, функционального и объектноориентированного программирования, с основными структурами данных и типовыми методами обработки этих структур;
- обеспечить освоение учащимися основных базовых алгоритмических конструкций и методов программирования для решения прикладных математических и информационных задач;
- формировать у учащихся умения работы на компьютере с использованием интегрированной среды IDLEPython и умения разработки эффективных алгоритмов и программ на основе изучения языка программирования Python.

#### *Развивающие:*

- развивать учебно-познавательные компетенции и функциональную грамотность у учащихся в процессе тренировки навыков, решения задач, касающихся написания программ различными методами;
- развивать у учащихся инженерно-техническое мышление, изобретательность, умение самостоятельно решать поставленную задачу;
- развивать у учащихся компетенции самоорганизации в процессе выработки и тренировки наиболее эффективной стратегии выполнения тестовых заданий;
- приобщать учащихся к проектно-творческой деятельности, развивать умения планирования проекта, умение работать в группе;

#### *Воспитательные:*

- воспитывать у учащихся ответственность за результаты учебного труда, понимание его значимости, соблюдение техники безопасности, санитарно-гигиенических условий труда;
- создать условия для профессионального самоопределения учащихся;
- воспитывать волю, умения преодолевать трудности, познавательную активность и самостоятельность, настойчивость в достижении результата в процессе решении учебных задач;
- воспитание интереса к предмету, к учению.

## **1.3. Учебно-тематический план и содержание программы**

### *Учебно-тематический план*

№ п/п	Наименование разделов и тем	Количество часов			Формы аттестации/контроля
		Всего	Теория	Практика	
	<b>Введение в программу «Основы программирования на языке Python».</b>	<b>2</b>	<b>1</b>	<b>1</b>	Опрос

<b>Раздел 1. Язык программирования Python.</b>		<b>6</b>	<b>2</b>	<b>4</b>	Тестирование, решение практических задач
1.1	Общие сведения о языке программы Python.	2	1	1	
1.2.	Режимы работы с Python.	2	1	1	
1.3	Решение практических задач.	2	-	2	
<b>Раздел 2. Переменные и выражения.</b>		<b>12</b>	<b>3</b>	<b>9</b>	Тестирование, решение практических задач
2.1	Переменные. Особенности работы со справочной системой.	4	1	3	
2.2	Выражения. Ввод и вывод.	2	1	1	
2.3	Выражения и операции.	2	1	1	
2.4	Задачи на элементарные действия с числами.	4	-	4	
<b>Раздел 3. Условные предложения.</b>		<b>18</b>	<b>4</b>	<b>14</b>	Тестирование, решение практических задач
3.1	Логические выражения и операторы.	2	1	1	
3.2	Условный оператор.	2	1	1	
3.3	Множественное ветвление.	2	1	1	
3.4	Реализация ветвления в языке Python.	2	1	1	
3.5	Самостоятельная работа №1 «Условные операторы».	4	-	4	
3.6	Составление программ с ветвлением.	6	-	6	
<b>Раздел 4. Циклы.</b>		<b>36</b>	<b>7</b>	<b>29</b>	Тестирование, творческая работа
4.1.	Оператор цикла с условием.	4	1	3	
4.2.	Оператор цикла while.)	2	1	1	
4.3.	Обновление переменной.	2	1	1	
4.4.	Оператор цикла for.	2	1	1	
4.5.	Вложенные циклы.	4	1	3	
4.6	Случайные числа.	4	1	3	
4.7	Примеры решения задач с циклом.	2	-	2	
4.8	Составление программ с циклом.	8	1	7	
4.9	Творческая работа «Циклы».	8	-	8	
<b>Раздел 5. Функции.</b>		<b>14</b>	<b>5</b>	<b>9</b>	Самостоятельная работа
5.1.	Создание функций.	2	1	1	
5.2.	Локальные переменные.	2	1	1	
5.3.	Встроенные функции.	2	1	1	

5.4	Примеры решения задач с использованием функций.	2	1	1	
5.5	Рекурсивные функции.	2	1	1	
5.6	Самостоятельная работа № 2 «Функции».	4	-	4	
<b>Раздел 6. Строки - последовательности символов.</b>		<b>6</b>	<b>2</b>	<b>4</b>	Решение практических задач
6.1.	Строки.	2	1	1	
6.2.	Срезы строк.	2	1	1	
6.3	Примеры решения задач со строками.	2	-	2	
<b>Раздел 7. Сложные типы данных.</b>		<b>20</b>	<b>10</b>	<b>10</b>	Презентация проекта
7.1.	Списки.	2	1	1	
7.2	Срезы списков.	2	1	1	
7.3	Матрицы.	2	1	1	
7.4	Кортежи.	2	1	1	
7.5	Введение в словари.	2	1	1	
7.6	Множества в языке Python.	2	1	1	
7.7	Модуль. Команды модуля.	2	1	1	
7.8	Классы и объекты.	2	1	1	
7.9	События.	2	1	1	
7.10	Работа с файлами.	2	1	1	
<b>Раздел 8. Стиль программирования и отладка программ.</b>		<b>28</b>	<b>3</b>	<b>25</b>	Презентация игры
8.1	Стиль программирования.	2	1	1	
8.2	Отладка программ.	2	1	1	
8.3	Игры на Python. Игра «Прыг-скок!»	8	1	7	
8.4.	Разработка игры.	16	-	16	
<b>Итоговое занятие. Презентация игры.</b>		<b>2</b>	<b>-</b>	<b>2</b>	
<b>ИТОГО:</b>		<b>144</b>	<b>37</b>	<b>107</b>	

### *Содержание программы*

**Введение в программу «Основы программирования на языке Python» (2 са).**

*Теория.* План работы объединения. Содержание программы. Инструктаж по ТБ. Общие сведения о языке Python.

*Практика.* Игры на знакомство и командообразование. Игры на развитие внимания, логического мышления, координацию движений. Установка Python на компьютер. Тренировочные упражнения.

*Форма контроля.* Опрос.

## **Раздел 1. Язык программирования Python (6 часов).**

### **1.1. Общие сведения о языке программы Python (2 ч.)**

*Теория.* Понятия «компьютерная программа», «алгоритм», «исполнитель», «язык программирования», «программа», «интерпретатор». Первая программа. История языка программирования Python и его возможности. Виды окон в IDLE: окно программы и окно консоли. Сравнение этих окон и их возможностей. Структура программы на языке Python. Виртуальная среда взаимодействия: регистрация, организация личного кабинета, поиск и выкладывание материалов. Система автоматизированной проверки задач.

*Практика.* Задания на знакомство с программой Python, её структурой.

### **1.2. Режимы работы с Python (2 ч.)**

*Теория.* Режимы работы Python.

*Практика.* Задания на знакомство с режимами работы Python.

### **1.3. Решение практических задач (2 ч.)**

*Практика.* Задания на выполнение простейших программ в интерактивной среде, написание комментариев в программе.

*Форма контроля.* Тестирование, решение практических задач.

## **Раздел 2. Переменные и выражения (12 часов).**

### **2.1. Переменные. Особенности работы со справочной системой (4 ч.)**

*Теория.* Типы данных. Преобразование типов. Понятие «переменная». Оператор присваивания. Имена переменных и ключевые слова. Правила именования переменных в языке Python. Оператор присваивания. Функция input().

*Практика.* Задания и упражнения на усвоение теоретического материала и работу со справочной системой. Проект «Аватар». В данном проекте отрабатывается функция input(), с помощью которой становится возможным ввести свои данные в программу и отобразить их.

### **2.2. Выражения. Ввод и вывод (2 ч.)**

*Теория.* Выражения. Операции. Порядок выполнения операций. Математические функции. Композиция. Ввод данных с клавиатуры. Вывод данных на экран. Пример скрипта, использующего ввод и вывод данных. Правила синтаксиса Python: правило начала, правило порядка, правило регистра. Понятие функции. Функция print().

*Практика.* Тренировочные упражнения. Решение кейсов. Проект «Символьная графика». Создание определённого рисунка с помощью символов. Отработка функции print().

### **2.3. Выражения и операции (2 ч.)**

*Теория.* Логический тип данных. Логические выражения и операторы. Сложные условные выражения (логические операции and, or, not). Условный оператор. Альтернативное выполнение. Примеры решения задач с условным оператором. Множественное ветвление. Реализация ветвления в языке Python. Арифметические операции с помощью математических операторов +, -, \*, /. Порядок выполнения операций. Понятие «выражение». Функции int() и str().

*Практика.* Проект «Сумматор». При написании данной программы отрабатываются математические операторы и функции int() и str().

## **2.4. Задачи на элементарные действия с числами (4 ч.)**

*Практика.* Решение задач на элементарные действия с числами.

*Форма контроля.* Тестирование, решение практических задач.

## **Раздел 3. Условные предложения (18 часов).**

### **3.1. Условный оператор (2 ч.)**

*Теория.* Понятие «условный оператор», «вложенные команды», «оператор сравнения». Условный оператор: понятие, назначение, способ записи. Альтернативное выполнение. Конструкция if и её синтаксис. Операторы сравнения: `>`, `>=`, `<`, `<=`, `!=`, `==`. Структура программы. Конструкция if-else. Команды if и elif

*Практика.* Решение задач на отработку условного оператора и операторов сравнения.

### **3.2. Логические выражения и операторы (2 ч.)**

*Теория.* Логический тип данных. Логические выражения и операторы. Сложные условные выражения (логические операторы: and, or, not). Порядок выполнения операций. Переменные без значения – None. Объединение условий.

*Практика.* Решение практических задач. Проект «Калькулятор»: создание приложения по определенным условиям.

### **3.3. Множественное ветвление (2 ч.)**

*Теория.* Множественное ветвление. Способы создания сложных условий с помощью логических операторов.

*Практика.* Задания на создание сложных условий с помощью логических операторов.

### **3.4. Реализация ветвления в языке Python (2 ч.)**

*Теория.* Способы реализации ветвления в языке Python.

*Практика.* Задания и упражнения на реализацию ветвления в языке Python.

### **3.5. Самостоятельная работа № 1 «Условные операторы» (4 ч.)**

*Практика.* Задания для самостоятельной работы.

### **3.6. Составление программ с ветвлением (6 ч.)**

*Практика.* Решение кейсов. Задания на составление программ с ветвлением.

*Форма контроля.* Тестирование, решение практических задач.

## **Раздел 4. Циклы (36 часов).**

### **4.1. Оператор цикла с условием (4 ч.)**

*Теория.* Понятие цикла. Тело цикла. Условия выполнения тела цикла. Виды циклов и их конструкции. Оператор цикла с условием.

*Практика.* Решение задач. Творческие задания.

### **4.2. Оператор цикла while (2 ч.)**

*Теория.* Понятие «цикл с предусловием». Оператор цикла while. Конструкция while и её синтаксис. Зацикливание и выход из цикла с помощью команды break. Бесконечные циклы. Альтернативная ветка цикла while.

*Практика.* Решение задач с использованием цикла while. Проект «Бомба взорвалась!». Написание программы по определенным условиям.

### **4.3. Обновление переменной (2 ч.)**

*Теория.* Обновление переменной. Краткая форма записи обновления. Примеры использования циклов.

*Практика.* Решение задач с циклами.

### **4.4. Оператор цикла for (2 ч.)**

*Теория.* Понятие «цикл со счётчиком». Конструкция for и её синтаксис. Оператор цикла с параметром for. Операторы управления циклом. Пример задачи с использованием цикла for.

*Практика.* Решение задач с использованием цикла for. Проект «Таблица умножения»: создание приложения по определенным условиям.

### **4.5. Вложенные циклы (4 ч.)**

*Теория.* Вложенные циклы. Циклы в циклах. Реализация циклических алгоритмов.

*Практика.* Практическая работа «Реализация циклических алгоритмов». Решение задач с вложенными циклами.

### **4.6. Случайные числа (4 ч.)**

*Теория.* Случайные числа. Функция randrange. Функция random.

*Практика.* Решение задач.

### **4.7. Примеры решения задач с циклом (2 ч.)**

*Практика.* Решение задач на применение циклов for и while.

### **4.8. Составление программ с циклом (8 ч.)**

*Теория.* Примеры составления программ с циклом.

*Практика.* Составление программ с циклом.

### **4.9. Творческая работа «Циклы» (8 ч.)**

*Практика.* Выполнение творческой работы «Циклы».

*Форма контроля.* Тестирование, творческая работа.

## **Раздел 5. Функции (14 часов).**

### **5.1. Создание функций (2 ч.)**

*Теория.* Понятия «функция», «параметр функции», «значение функции». Строение функции: имя, аргумент, тело. Создание и вызов функции. Переменные и область видимости. Параметры и аргументы. Способы описания функции. Принципы структурного программирования.

*Практика.* Задания и упражнения на создание и использование функций. Решение задач на отработку понятия «функция», её строение и синтаксис.

### **5.2. Локальные переменные (2 ч.)**

*Теория.* Локальные и глобальные переменные. Поток выполнения. Функции, возвращающие результат. Анонимные функции, инструкция lambda. Механизм параметров для передачи значений.

*Практика.* Задания на использование механизма параметров для передачи значений.

### **5.3. Встроенные функции (2 ч.)**

*Теория.* Понятие «встроенные функции». Функции: abs, bool, dir, eval, exec, float, int, len, max, min, range, sum.

*Практика.* Решение задач на отработку функций.

#### **5.4. Примеры решения задач с использованием функций (2 ч.)**

*Теория.* Примеры решения задач с использованием функций.

*Практика.* Решение задач с использованием функций.

#### **5.5. Рекурсивные функции (2 ч.)**

*Теория.* Рекурсивные функции. Вычисление факториала. Числа Фибоначчи.

*Практика.* Решение задач.

#### **5.6. Самостоятельная работа № 2 «Функции» (4 ч.)**

*Практика.* Задания по разделу для самостоятельной работы.

*Форма контроля.* Самостоятельная работа.

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

#### **6.1. Строки (2 ч.)**

*Теория.* Понятие «строка». Составной тип данных - строка. Доступ по индексу. Длина строки и отрицательные индексы. Преобразование типов. Применение цикла для обхода строки. Создание строк. Переменные внутри строк. Операции со строками.

*Практика.* Задания на описание, соединение строк, нахождение длины строки и др. Решение задач со строками.

#### **6.2. Срезы строк (2 ч.)**

*Теория.* Срезы строк. Строки нельзя изменить. Сравнение строк. Оператор in. Модуль string. Операторы для всех типов последовательностей (строки, списки, кортежи).

*Практика.* Групповая проблемная работа: решение кейсов.

#### **6.3. Примеры решения задач со строками (2 ч.)**

*Практика.* Решение задач со строками.

*Форма контроля.* Решение практических задач.

### **Раздел 7. Сложные типы данных (20 часов).**

#### **7.1. Списки (2 ч.)**

*Теория.* Понятие «списки», «элементы списка» и др. Тип список (list). Индексы. Обход списка. Проверка вхождения в список. Добавление в список. Суммирование или изменение списка. Операторы для списков. Понятие «список». Создание списков. Добавление/удаление элементов в/из список/списка. Операции со списками.

*Практика.* Задания и упражнения на умение описывать списки, вводить элементы списка, выводить элементы списка, выполнять поиск элемента в списке, поиск минимума и максимума, нахождение суммы элементов списка, использовать вложенные списки.

#### **7.2. Срезы списков (2 ч.)**

*Теория.* Срезы списков. Удаление списка. Клонирование списков. Списочные параметры. Функция range.

*Практика.* Решение задач на отработку операций со строками и списками.

#### **7.3. Матрицы (2 ч.)**

*Теория.* Матрицы. Вложенные списки. Матрицы. Строки и списки. Генераторы списков в Python.

*Практика.* Задания на поиск и приведение примеров использования вложенных списков (матриц).

#### **7.4. Кортежи (2 ч.)**

*Теория.* Понятие «кортеж». Создание кортежа. Операции с кортежем. Присваивание кортежей. Кортежи как возвращаемые значения.

*Практика.* Задания и упражнения на отработку полученных знаний и творческого характера.

#### **7.5. Введение в словари (2 ч.)**

*Теория.* Введение в словари. Понятие «словарь». Создание словаря. Тип словарь (dict). Словарные операции. Словарные методы.

*Практика.* Решение кейсов. Проект «Любимые вещи»: создание списка любимых развлечений и любимых лакомств.

#### **7.6. Множества в языке Python (2 ч.)**

*Теория.* Множества в языке Python. Множества. Множественный тип данных. Описание множеств. Операции, допустимые над множествами: объединение, пересечение, разность, включение. Оператор определения принадлежности элемента множеству.

*Практика.* Задания на описание множеств, определение принадлежности элемента множеству, введение элементов множества, вывод элементов множества.

#### **7.7. Модуль. Команды модуля (2 ч.)**

*Теория.* Понятие «модуль». Модуль turtle. Импортирование модуля в программу. Полезные модули: random, time, pickle. Создание холста. Перемещение черепашки.

*Практика.* Решение задач на применение модулей. Проект «Я рисую»: рисование изображения, состоящего из линий с помощью команд модуля на холсте.

#### **7.8. Классы и объекты (2 ч.)**

*Теория.* Понятие «объект». Концепция объектов. Понятие «класс». Использование классов в Python. Инициализация объектов.

*Практика.* Решение задач на отработку умения создавать классы и объекты.

#### **7.9. События (2 ч.)**

*Теория.* Понятие «событие», «подписка на событие», «обработка события». Особенности рисования фигур с использованием управляющих конструкций.

*Практика.* Проект «Поймай черепашку». Пишется приложение-игра, в котором создается холст и на нём прорисовывается черепашка, при нажатии на которую происходит ее перемещение в новое место на холсте.

#### **7.10. Работа с файлами (2 ч.)**

*Теория.* Понятие «файл». Классификация файлов в зависимости от видов информации. Создание текстового файла. Открытие файла в Python. Запись в файл.

*Практика.* Проект «Блокнот»: создание простейшего текстового редактора в виде приложения.

#### **7.11. Графика с модулем tkinter (2 ч.)**

*Теория.* Модуль tkinter и его возможности.



*Практика.* Проект «Анимация»: отрисовывание объекта на холсте и программирование анимации.

*Форма контроля.* Презентация проекта.

## **Раздел 8. Стил ь программирования и отладка программ (28 ч.)**

### **8.1. Стил ь программирования (2 ч.)**

*Теория.* Стил ь программирования. Правила именования объектов. Рекомендации при написании программ.

*Практика.* Написание программ по заданию педагога.

### **8.2. Отладка программ (2 ч.)**

*Теория.* Виды ошибок в программе. Способы определения разных видов ошибок и их нахождения в программе. Особенности тестирования и отладки программ.

*Практика.* Задания на тестирование и отладку программ.

### **8.3. Игры на Python. Игра «Прыг-скок!» (8 ч.)**

*Теория.* Обсуждение сюжета игры: объекты и их роль. Создание игрового холста. Создание главного цикла игры. Отрисовка мяча. Программирование мяча: перемещение, отскоки от границ игрового холста, первоначальное направление движения мяча. Отрисовка ракетки. Программирование ракетки: управление, установления границ. Обработка события столкновения мяча и ракетки. Обработка события «проигрыш». Доработка игры: задержка перед началом игры; экран «Конец игры»; ускорение мяча; счет в игре.

*Практика.* Приложение-игра «Прыг-скок!»: неуправляемый пользователем объект «мяч» и управляемые пользователем объект «ракетка».

### **8.4. Разработка игры (16 ч.)**

*Практика.* Работа над проектом: разработка игры (по заданию педагога).

*Форма контроля.* Презентация игры.

## **Итоговое занятие. Презентация игры (2 часа).**

*Практика.* Презентация игр учащимися.

*Форма контроля.* Презентация игры.

## **1.4. Планируемые результаты**

По окончании обучения учащиеся имеют следующие результаты.

### **Предметные результаты:**

*знают:*

- понятия в рамках программы;
- структуру, режимы работы программы на Python;
- целые, вещественные типы данных и операции над ними;
- оператор присваивания;
- операторы ввода-вывода.
- назначение и способ записи условного оператора;
- логический тип данных и логические операторы or, and, not;
- циклы с условием и их виды;

- правила записи циклов условием;
- назначение и особенности использования цикла с параметром;
- формат записи цикла с параметром;
- примеры использования циклов различных типов;
- назначение и способы описания функции;
- принципы структурного программирования;
- понятие локальных переменных подпрограмм, понятие формальных и фактических параметров подпрограмм;
- способ передачи параметров.
- назначение строкового типа данных;
- операции, операторы, процедуры и функции для работы со строками;
- сложные типы данных;
- способы описания списка и доступа к элементам списка;
- способ описания кортежа;
- способ описания словаря;
- операции, выполняемые со списками, кортежами и словарями;
- способы описания множества и операторы работы с множествами.
- о стилях программирования;
- правила именования объектов;
- основные рекомендации при написании программ;

**уметь:**

- выполнить установку программы Python;
- выполнить простейшую программу в интерактивной среде;
- писать комментарии в программе;
- пользоваться интерфейсом среды программирования Python;
- использовать команды редактора;
- организовывать ввод и вывод данных;
- записывать арифметические выражения.
- использовать условный оператор;
- создавать сложные условия с помощью логических операторов;
- 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;
- создавать и использовать функции;
- использовать механизм параметров для передачи значений;
- описывать, соединять строки, находить их длину, вырезать часть строки, находить подстроку в строке, находить количество слов в строке;
- описывать списки;
- вводить и 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 и выводить элементы множества;
- определять вид ошибок и находить ошибки в программе;
- выполнять тестирование и отладку программ;
- создавать простые игры;
- работать над проектами.

В результате обучения по программе учащиеся приобретут

**- личностные качества:**

- осмысление мотивов своих действий при выполнении заданий с жизненными ситуациями;
- критическое отношение к информации и избирательность её восприятия;
- осмысление мотивов своих действий при выполнении заданий;
- любознательность, сообразительность при выполнении разнообразных заданий;
- внимательность, настойчивость, целеустремленность, умение преодолевать трудности – качеств весьма важных в практической деятельности любого человека;
- самостоятельность суждений, независимость и нестандартность мышления;
- начало профессионального самоопределения, ознакомление с миром профессий, связанных с информационными и коммуникационными технологиями.

**- метапредметные компетенции:**

- принимать и сохранять учебную задачу;
- планировать последовательность шагов алгоритма для достижения цели;
- ставить цель, планировать и достигать этой цели;
- осуществлять итоговый и пошаговый контроль по результату;
- адекватно воспринимать оценку педагога;
- различать способ и результат действия;
- в сотрудничестве с педагогом ставить новые учебные задачи;
- проявлять познавательную инициативу в учебном сотрудничестве;
- оценивать получающийся творческий продукт и соотносить его с изначальным замыслом.

## РАЗДЕЛ 2. КОМПЛЕКС ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИХ УСЛОВИЙ

### 2.1. Календарный учебный график

Год обучения	Объем учебных часов	Всего учебных недель	Режим работы	Кол-во учебных дней	Даты начала и окончания учебных периодов/ Этапов	Продолжительность каникул
1 год	144	36	2 раза в неделю по 2 часа	72	01 сентября – 31 мая	30 декабря – 08 января

### 2.2. Условия реализации программы

#### *Методическое обеспечение.*

Для реализации программы используются следующие *методы и приёмы обучения*:

- по источнику полученных знаний: словесные, наглядные, практические;
- по способу организации познавательной деятельности:
  - развивающее обучение (проблемный, проектный, творческий, частично-поисковый, исследовательский, программированный);
  - дифференцированное обучение (уровневые, индивидуальные задания).
  - игровые методы (конкурсы, игры-конструкторы, турниры с использованием мультимедиа, дидактические).

#### *Формы организации работы учащихся:*

Индивидуальная – используется на всех этапах учебного занятия для решения различных дидактических задач: усвоения новых знаний и их закрепления, формирования и закрепления умений и навыков, для повторения и обобщения пройденного материала. Преимущества этой формы организации учебной работы в том, что она позволяет каждому учащемуся получить опыт познавательной проектной деятельности.

Групповая – используется для выполнения итоговых мини-проектов учащимися (по желанию, некоторые учащиеся могут выполнять проект индивидуально), где каждый член группы выполняет свою задачу, далее коллективно защищают свой проект.

Фронтальная - используются на этапе первичного усвоения нового материала. Эта форма позволяет привлечь к активной учебно-познавательной деятельности всех учащихся.

*Педагогические технологии:* ИКТ, проектная технология, здоровьесберегающие технологии. Использование интерактивных досок, веб-платформ, онлайн-занятий и других средств, которые позволяют учащимся активно участвовать в занятии и взаимодействовать с учебным материалом. Использование

элементов игры в учебных заданиях для увлекательности и мотивации учащимся.

*Алгоритм учебного занятия* в соответствии с моделью состоит из последовательности следующих этапов: организационного, проверочного, подготовительного, основного, контрольного, рефлексивного (самоанализ), итогового, информационного. Каждый этап отличается от другого сменой видов деятельности, содержанием и конкретной задачей. Основанием для выделения этапов может служить процесс усвоения знаний, который строится как смена видов деятельности учащихся: восприятие – осмысление – запоминание – применение – обобщение – систематизация.

*Воспитательная работа* в рамках реализации данной программы осуществляется в соответствии с направлениями воспитательной работы, которые описаны в «Программе воспитания школы № 3» (<https://school3.kuz-edu.ru/index.php?id=35327>).

### **Материально-техническое обеспечение.**

Занятия проходят в хорошо проветриваемой и освещённой аудитории, с мебелью, соответствующей санитарно-техническим требованиям и нормам возрастной физиологии (парты, стулья, стол и стул педагога) и с доступом в интернет.

На компьютерах, которые расположены в кабинете информатики, должна быть установлена операционная система Linux, а также необходимое программное обеспечение:

- текстовый редактор (Блокнот или Gedit) и текстовый процессор (MS Word или OpenOffice Writer);
- табличный процессор (MS Excel или OpenOffice Calc);
- средства для работы с базами данных (MS Access или OpenOffice Base);
- среда программирования Python 3+, интегрированная среда разработки.

Материалы и оборудование, имеющиеся в учреждении:

<b>Наименование</b>	<b>Кол-во (шт.)</b>
Парты двухместные	9
Стол�ы трапецевидные	6
Стулья ученические	30

В том числе при проведении занятий используется оборудование, приобретённое за счёт средств федерального бюджета в рамках функционирования центра цифрового образования детей «IT-куб» для реализации дополнительных общеразвивающих программ федерального проекта «Цифровая образовательная среда».

<b>Наименование</b>	<b>Кол-во (шт.)</b>
Ноутбук	16
Интерактивная панель	1
Доска флипчарт	1
Доска магнитно - маркерная	1
МФУ	1

***Информационное обеспечение.***

- Домашняя страница Python [www.python.org](http://www.python.org). Справочные материалы, официальная документация.
- Дистанционная подготовка. Курс «Основы языка программирования Python: легко и просто» на python «Д.П. Кириенко. «Основы языка программирования Python: легко и просто» на языке Python (школа 179 г. Москва)» (<http://informatics.msk.ru/course/view.php?id=156> )
- Сайт проекта Интуит: Национальный открытый университет, курс «Введение в программирование на Python», <http://www.intuit.ru/studies/courses/12179/1172/info>.
- Сайт проекта Интуит: Национальный открытый университет. Курс «Язык программирования Python» <http://www.intuit.ru/studies/courses/49/49/info>.
- Сайт проекта Open Book Project [openbookproject.net](http://openbookproject.net) содержит серию практических примеров на Python Криса Мейерса.
- Python. Подробный справочник Дэвида М. Бизли — книга со справочной информацией о языке Python и модулях стандартной библиотеки.
- Python. Справочник Марка Лутца. Справочник по наиболее часто используемым функциям и модулям.
- Компьютерный практикум в электронном виде с комплектом электронных учебных средств, размещённый на сайте: <http://kpolyakov.spb.ru/school/probook.htm>
- Материалы для подготовки к итоговой аттестации по информатике в форме ОГЭ, размещённые на сайте <http://kpolyakov.spb.ru/school/oge.htm>.

***Кадровое обеспечение.*** Кадровое обеспечение разработки и реализации дополнительной общеобразовательной общеразвивающей программы осуществляется учителем высшей квалификационной категории. Педагог имеет высшее педагогическое образование и прошел курсы повышения квалификации по программе «Использование современного учебного оборудования в центрах цифрового образования «IT-куб»

### 2.3. Формы аттестации/контроля

Результатом данной программы является определяемый уровень усвоения учащимися основных знаний и навыков работы в среде программирования Python.

Диагностика ***и контроль*** результатов обучения осуществляется в несколько этапов:

- *вводный контроль* проводится с целью выявления образовательного уровня учащихся на начало обучения и способствует подбору эффективных форм и методов деятельности для учащихся;
- *тематический контроль* позволяет определить степень сформированности знаний, умений и навыков по темам программы, это дает возмож-

ность своевременно выявлять пробелы в знаниях и оказывать учащимся помощь в усвоении содержания материала;

- *промежуточный* - проводится в конце первого полугодия с целью отслеживания уровня освоения программного материала;
- *итоговый контроль* призван определить конечные результаты обучения.

**Формы аттестации учащихся, предъявления и демонстрации образовательных результатов.** Формой аттестации учащихся и подведения итогов реализации программы является презентация игры.

За время обучения учащиеся могут принимать участие в конкурсах и конференциях научно-практических работ технической направленности различного уровня.

## 2.4. Оценочные материалы

Этапы диагностики/ контроля	Форма диагностики/аттестации
<i>вводный</i>	Опрос.
<i>тематический</i>	Тестирование, решение практических задач. Тестирование, творческая работа. Решение практических задач. Презентация проекта.
<i>промежуточный</i>	Самостоятельная работа.
<i>итоговый</i>	Презентация игры.

*Формы и методы оценивания результатов:* педагогическое наблюдение, создание своего оригинального продукта, умение эффективного взаимодействия в команде.

*Формы фиксации результатов:* протокол.

*Форма оценки:* уровень (высокий, средний, низкий).

## 2.5. Методические материалы

### *Дидактические материалы:*

- Правила по технике безопасности.
- Руководства для начинающих: Подробное введение в язык программирования Python, основные концепции программирования, создание первых проектов.
- Учебные планы: Структурированные планы занятий, охватывающие постепенное углубление в различные аспекты программирования с использованием Python.
- Практические задачи: Задачи разной сложности, нацеленные на развитие определенных навыков и понимания конкретных аспектов программирования.

### *Наглядные материалы:*

- Карточки с заданиями: Комплекты карточек с заданиями на бумаге, которые можно использовать для учебных игр и групповой работы.
- Видеоролики и анимации: Анимированные материалы, демонстрирующие процесс создания проектов и работу с интерфейсом Python.
- Иллюстрации к проектам: Графические описания проектов, созданных на Python, с подписями, пояснениями и скриншотами интерфейса.

## 2.6. Список литературы

### *Для педагога:*

#### *- основная литература:*

1. Бабанский Ю.К. Педагогика. - Москва: Просвещение, 2020. - 626 с.
2. Бордовская Н.В. Психология и педагогика: учебник. – Санкт-Петербург: Санкт-Петербург, 2019. - 320 с.
3. Васильев А.Н. Программирование на Python в примерах и задачах. – Москва: Бонбора, 2021. – 616 с. – (Российский компьютерный бестселлер).
4. Ефимов К.Н. Программирование на языке Python. - Калуга, 2019. – 41 с.
5. Задачник-практикум по информатике: Учебное пособие для средней школы/ Под ред. И. Г. Семакина, Е. К. Хеннера. — Москва: БИНОМ. Лаборатория, 2021. – 188 с.
6. Зимняя И.А. Педагогическая психология. – Ростов-на-Дону: Феникс, 2019. - 477 с.
7. Козлова В.А. Общая педагогика. Курс обзорных лекций. – Москва: Просвещение, 2019. – 285 с.
8. Прядкина Е.В. Программирование аркадных игр на языке Python. - Москва, 2020. – 35 с.

#### *- дополнительная литература:*

1. Бриггс Дж. Python для детей. Самоучитель по программированию. /Джейсон Бриггс; пер. с англ. Станислава Ломакина. – Москва: Манн, Иванов и Фербер, 2017. – 320 с.
2. Буйначев С.К. Основы программирования на языке Python [Электронный ресурс]: учебное пособие / С. К. Буйначев, Боклаг Н. Ю. – Электрон. текстовые дан. – Издательство Уральского университета, 2014.
3. Васильев А.Н. Python на примерах. Практический курс по программированию. Руководство. - Москва: Наука и техника, 2017. - 752 с.
4. Гин А.А. Приёмы педагогической техники: свобода выбора, открытость, деятельность, обратная связь, идеальность: Пособие для учителей / А.А. Гин. – Гомель: ИПП «Сож», 1999. – 88 с.
5. Златопольский Д.М. Основы программирования на языке Python. – Москва: ДМК Пресс, 2017. – 284 с.
6. Информатика. Задачник - практикум в 2т. / Под ред. И.Г. Семакина, Е.К. Хеннера; Том 1. – Москва: Бином. Лаборатория Знаний, 2002. – 136 с.
7. Крылов С.С., Ушаков Д.М. Информатика. Решение сложных задач. /



- ФИПИ. - Москва: Интеллект-Центр, 2010. – 152 с. - (Отличник ЕГЭ).
8. Лутц М. Изучаем Питон. - Санкт-Петербург: Символ, 2011. – 297 с.
  9. Михелькевич В.Н. «Метод проектов» и его использование в средней общеобразовательной и высшей инженерной школах: учебное пособие / под ред.: В.Н. Михелькевич, Н.В. Охтя. – Самара: Издательство Самарского государственного технического университета, 2004. – 48 с.
  10. Окулов С.М. Основы программирования. - Москва, 2012. – 37 с.
  11. Окулов С.М. Программирование в алгоритмах. – Москва: Бингм. Лаборатория знаний, 2002. – 190 с.
  12. Поляков К.Ю., Еремин Е.А.. Информатика: углубленный уровень. - Москва: Бином. Лаборатория Знаний, 2013. – 281 с.
  13. Федоров Д.Ю. Программирование на языке высокого уровня Python: учебное пособие для прикладного бакалавриата. – 2-е изд., перераб. и доп. – Москва: Юрайт, 2019. – 161 с.

***Для учащихся:***

1. Банкрашков А.В. Программирование для детей на языке Python. - Москва: АСТ, 2018. - 288 с.
2. Бэрри П. Изучаем программирование на Python. / Пол Бэрри; Пер. с англ. М.А. Райтман. - Москва: Издательство «Э», 2017. – 624 с. – (Мировой компьютерный бестселлер). [Электронный ресурс] / CoderNet – Режим доступа:  
[https://codenet.ru/books/python/izuchaem\\_programmirovanie\\_na\\_python/](https://codenet.ru/books/python/izuchaem_programmirovanie_na_python/)
3. Вордерман К. Программирование на Python. Иллюстрированное руководство для детей / К. Вордерман, К. Стили, К. Квигли. - Москва: Манн, Иванови Фербер, 2017. - 346 с.
4. Гуриков С.Р. Основы алгоритмизации и программирования на Python. Учебное пособие. Гриф МО РФ. - Москва: Инфра-М, Форум, 2018. - 707 с.
5. Пэйн, Б. Python для детей и родителей./ БрайсонПэйн; Пер. с англ. М.А. Райтман. – Москва: Издательство «Э», 2017. – 352 с.
6. Свейгарт Эл. Автоматизация рутинных задач с помощью Python: практическое руководство для начинающих. Пер. с англ. — Москва: Вильямс, 2016. – 592 с.

***Интернет-ресурсы:***

1. Руководство по языку программирования Python [Электронный ресурс] / URL: <https://metanit.com/python/> (дата обращения 18.09.2024г.)
2. Центр педагогического мастерства. МЦНМО. Система автоматической проверки решений задач [Электронный ресурс] / URL: <http://informatics.msk.ru/> (дата обращения 18.09.2024г.)

## ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

### *Критерии оценивания контрольных заданий*

Набранный балл	Оценка	Критерий
5	Высокий уровень	Получен полный и развёрнутый ответ на вопрос, приведены иллюстрирующие ответ примеры, получены ответы на дополнительные вопросы педагога.
4	Средний уровень	Получен полный и развёрнутый ответ на вопрос, приведены иллюстрирующие ответ примеры, но не получены ответы на дополнительные вопросы педагога.
3	Низкий уровень	Получен неполный ответ на вопрос, не приведены иллюстрирующие ответ примеры, получены неполные ответы на дополнительные вопросы педагога.

Материалы для организации и проведения учебно-исследовательской и проектной деятельности школьников

Проект по программированию представляет собой проект, 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 учащиеся 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 учащихся. 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.

### Лист оценивания проекта

<b>Критерий оценивания</b>	<b>1-я группа</b>	<b>2-я группа</b>
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		

## ГЛОССАРИЙ

**Аг-репорт** — подробное описание ошибки в работе программы с указанием, где именно произошла ошибка и как она повлияла на работу программы. Чаще всего составлением таких документов занимается тестировщик.

**База данных** — набор упорядоченной информации, которая хранится на каком-то носителе. Например, на жестком диске или на сервере.

**СУБД** — это система управления базами данных. Так называют сложное программное обеспечение, которое требуется, чтобы создавать базы данных, изменять их, получать из них информацию и контролировать версии.

**Компилятор** — программа, которая переводит код на языке программирования, понятном человеку, на язык машинного кода, понятном компьютеру.

**Массив** — тип данных, который представляет собой набор упорядоченных и однотипных элементов. Массивы есть и в Python, и в большинстве других языков программирования.

**Математическая статистика** — наука, которая помогает анализировать большие объемы данных, описывать их и строить прогнозы.

**Теория вероятностей** — раздел математики, который изучает закономерности случайных явлений. Он лежит в основе работы с большими данными.

**Объектно-ориентированное программирование (ООП)** — парадигма программирования, которая рассматривает программу не как единое целое, а как набор объектов, взаимодействующих друг с другом.

**Машинное обучение** — набор методов искусственного интеллекта, которые позволяют использовать его для разных практических целей. Например, для автоматизации процессов, перевода текстов, распознавания изображений или видео.

**Репозиторий** — хранилище данных, которое можно сравнить с каталогом информации. В разных его частях содержится код — данные, к которым он обращается и другие файлы. Чаще всего употребляется в контексте распределенных систем управления версиями, таких как GitHub.

**GitHub** — это онлайн-хранилище версий кода, в которой несколько разработчиков могут одновременно писать код. Одновременно это самая популярная соцсеть для программистов, где разработчики выкладывают код своих проектов, а их коллеги и рекрутеры могут оценить его качество.

**Фреймворк** — программная платформа, которая упрощает разработку. Она задает структуру, определяет правила и предоставляет необходимый набор инструментов для создания программы — какой, зависит от типа и назначения фреймворка.

**Open Source** — программное обеспечение с открытым исходным кодом. Чаще всего оно распространяется бесплатно, а принять участие его разработке или кастомизировать под свои нужды без нарушения авторских прав может любой разработчик.

**Data Science** — или наука о данных. Это раздел науки, связанной с разными аспектами работы с информацией: ее обработкой, анализом и использованием в практических целях.

**Deep learning** — глубокое машинное обучение. Это вид машинного обучения, при котором многослойные нейросети самостоятельно обучаются на больших массивах данных.