

Министерство образования и науки Мурманской области  
Государственное автономное профессиональное образовательное учреждение  
Мурманской области «Мурманский колледж экономики и информационных технологий»  
Центр цифрового образования детей «IT – куб»

РАССМОТРЕНА  
Методическим советом  
ГАПОУ МО «МКЭиИТ»  
Протокол № 4 от 24.06.2022 г.

УТВЕРЖДЕНА  
Приказом и.о. директора  
ГАПОУ МО «МКЭиИТ»  
№ 142 от «24» июня 2022 г.

## **Дополнительная общеобразовательная общеразвивающая программа технической направленности**

### **«ПРОГРАММИРОВАНИЕ НА PYTHON»**

Направленность: техническая  
Уровень программы: стартовый  
Срок реализации: 1 год (144 часа)  
Возраст детей: 15-17 лет

**Разработчики:**  
Комарова Т. В., педагог  
дополнительного образования,  
Кузнецова К. В, методист

г. Мурманск  
2022 г.

## СОДЕРЖАНИЕ

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА.....	3
2. ПРОГНОЗИРУЕМЫЕ РЕЗУЛЬТАТЫ.....	8
3. СОДЕРЖАНИЕ ПРОГРАММЫ.....	10
4. ФОРМЫ АТТЕСТАЦИИ И ОЦЕНОЧНЫЕ МАТЕРИАЛЫ .....	18
5. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ .....	27
6. СПИСОК ЛИТЕРАТУРЫ .....	29

## 1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

В настоящее время мы переживаем большие изменения в развитии общества. В современную жизнь человека все больше и больше внедряются компьютеры и информационные технологии. Всё большее значение приобретает умение человека грамотно обращаться с компьютером, причем зачастую не на пользовательском уровне, а на уровне начинающего программиста.

В обязательном школьном курсе информатики программирование нередко представлено лишь на элементарном уровне, на это выделяется недостаточное количество часов. Лишь немногие школы могут себе позволить преподавать программирование на достойном уровне. Следствием этого является формальное восприятие учащимися основ современного программирования и неумение применять полученные знания на практике.

Изучение основных принципов программирования невозможно без регулярной практики написания программ на каком-либо языке. В данной программе выбран язык программирования Python. Данный выбор обусловлен тем, что синтаксис языка достаточно прост и интуитивно понятен, а это понижает порог вхождения и позволяет сосредоточиться на логических и алгоритмических аспектах программирования, а не на выучивании тонкостей синтаксиса. При этом 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. Программа разработана на основе:

- Федерального закона от 29.12.2012 № 273-ФЗ (ред. от 31.07.2020) «Об образовании в Российской Федерации» (с изм. и доп., вступ. в силу с 01.09.2020);

- Паспорта национального проекта «Образование» (утв. президиумом Совета при Президенте РФ по стратегическому развитию и национальным проектам, протокол от 24.12.2018 № 16);

-Государственной программы Российской Федерации «Развитие образования» (Утверждена Постановлением Правительства РФ от 26.12.2017 № 1642 (ред. от 22.02.2021) «Об утверждении государственной программы Российской Федерации «Развитие образования»;

- Методических рекомендаций по созданию и функционированию центров цифрового образования «IT-куб» (утверждены распоряжением Министерства просвещения Российской Федерации от 12 января 2021 № Р-5);

- Рекомендаций Министерства просвещения Российской Федерации по реализации дополнительной общеобразовательной программы по направлению «Разработка виртуальной и дополненной реальности» с использованием оборудования Центра цифрового образования детей «IT-куб»;

- Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам, утвержденным приказом Министерства просвещения Российской Федерации от 09.11.2018 № 196;

- Постановления Главного государственного санитарного врача Российской Федерации от 28.09.2020 № 28 «Об утверждении санитарных правил СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи»;

- Постановления Главного государственного санитарного врача Российской Федерации от 28.01.2021 № 2 «Об утверждении санитарных правил и норм СанПиН 1.2.3685-21 Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания»;

- Распоряжения Правительства Российской Федерации от 31 марта 2022 года № 678-р «Концепция развития дополнительного образования детей до 2030 года»;

- Распоряжения Правительства Российской Федерации от 29 мая 2015 года № 996-р «Стратегия развития воспитания в Российской Федерации на период до 2025 года»;

- Приказа Министерства труда и социальной защиты Российской Федерации от 22.09.2021 № 652н «Об утверждении профессионального стандарта «Педагог дополнительного образования детей и взрослых».

**Направленность программы:** техническая.

**Уровень программы:** стартовый.

**Новизна, актуальность, педагогическая целесообразность программы.** Данная программа составлена с учётом современных потребностей рынка в специалистах в области информационных технологий, особенно в области программирования. Научившись программировать на языке Python, учащиеся получают мощный и удобный инструмент для решения как учебных, так и прикладных задач. Знания и умения, приобретенные в результате освоения курса, могут быть использованы обучающимися 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.

**Отличительные особенности программы.** Особенность программы «Программирование на Python» заключается в изучении основ программирования на языке Python, основных приёмов написания программ на современном языке программирования, развитие алгоритмического мышления учащихся, творческих способностей, аналитических и логических компетенций.

**Цель программы:** создание условий для изучения методов программирования на языке Python; подготовка к использованию языка и методов программирования на Python в учебной и последующей деятельности в различных предметных областях.

Для достижения цели планируется решить следующие задачи программы:

**Обучающие:**

- познакомить с правилами безопасного использования цифровыми инструментами и компьютерным оборудованием, организации рабочего места;

- формирование навыков работы в интегрированной среде разработки на языке Python;
- изучение конструкций языка программирования Python;
- формирование навыков разработки эффективных алгоритмов и программ на языке программирования Python;
- формирование представления о значении информационных технологий в развитии общества и в изменении характера труда человека.

**Развивающие:**

- развитие навыков алгоритмического и логического мышления, грамотной разработки программ;
- развитие навыков поиска информации в сети Интернет, анализа выбранной информации на соответствие запросу, использования информации при решении задач;
- способствовать формированию познавательных и регулятивных универсальных учебных действий;
- формирование творческого подхода к поставленной задаче;
- развитие навыков эффективной деятельности в проекте;
- формирование навыков рефлексивной деятельности.

**Воспитательные:**

- формирование стремления к получению качественного законченного результата в проектной деятельности;
- воспитание социально-значимых качеств личности человека: ответственность, коммуникабельность, добросовестность, взаимопомощь, доброжелательность.

**Адресат программы:** дети в возрасте от 15 до 17 лет.

**Наполняемость группы:** от 10 до 12 человек.

**Содержание и объем стартовых знаний, необходимых для начального этапа освоения программы:** особых знаний не требуется.

**Срок реализации программы:** 1 год

**Объем программы:** 144 часа

**Режим занятий:** 2-3 раза в неделю, число и продолжительность занятий в день: 2 занятия по 45 минут.

**Формы организации учебной деятельности:** групповая, индивидуальная, коллективная.

## 2. ПРОГНОЗИРУЕМЫЕ РЕЗУЛЬТАТЫ

### Предметные результаты:

В результате освоения программы, учащиеся будут:

#### Знать:

- правила работы с компьютером и технику безопасности;
- основные предметные понятия («информация», «алгоритм», «исполнитель», «модель») и их свойства;
- основы языка Python;
- особенности работы с интегрированной средой разработки;
- базовые и сложные конструкции, способы организации процедур и функций в языке программирования Python;
- технику ведения проектной деятельности и принципов тайм-менеджмента.

#### Уметь:

- работать с информацией: находить с применением правил поиска в компьютерных сетях, оценивать и использовать информацию из различных источников при выполнении заданий и проектов по различным темам;
- определять результат выполнения алгоритма при заданных исходных данных, узнавать изученные алгоритмы обработки чисел и числовых последовательностей, читать и понимать программы, написанные на языке программирования высокого уровня Python;
- самостоятельно ставить и формулировать для себя новые задачи, развивать мотивы своей познавательной деятельности;
- самостоятельно планировать пути решения поставленной проблемы для получения эффективного результата;
- критически оценивать правильность решения учебно-исследовательской задачи;
- корректировать свои действия, вносить изменения в программу и отлаживать её в соответствии с изменяющимися условиями;



- организовывать учебное сотрудничество и совместную деятельность с учителем и сверстниками в процессе проектной и учебно-исследовательской деятельности.

**Владеть:**

- навыками разработки эффективных алгоритмов и программ на основе изучения языка программирования Python;
- навыками использования специальных средств и библиотек языка Python;
- навыками работы в интегрированной среде разработки на языке программирования Python;
- навыками алгоритмического и логического мышления, грамотной разработки программ.

**Метапредметные результаты:**

- формирование умения выбирать наиболее эффективные способы решения задач на компьютере в зависимости от конкретных условий;
- формирование приёмов проектной деятельности, включая умения видеть проблему, формулировать тему и цель проекта, составлять план своей деятельности, осуществлять действия по реализации плана, результат деятельности соотносить с целью, классифицировать, наблюдать, проводить эксперименты, делать выводы и заключения, доказывать, защищать свои идеи, оценивать результаты своей работы;
- формирование умения распределять время;
- формирование умений успешной самопрезентации.

**Личностные результаты:**

- формирование умения самостоятельной деятельности;
- формирование умения работать в команде;
- формирование коммуникативных навыков;
- формирование навыков анализа и самоанализа;
- формирование целеустремлённости и усидчивости в процессе творческой, исследовательской работы и учебной деятельности.

### 3. СОДЕРЖАНИЕ ПРОГРАММЫ

#### Учебно-тематический план

№ п/п	Название разделов, тем	Количество часов			Форма аттестации/ контроля
		Всего	Теория	Практика	
<b>1</b>	<b>Введение в программирование</b>	<b>26</b>	<b>10</b>	<b>16</b>	
1.1	Вводное занятие. Инструктаж по ТБ. Введение в программу 1-го года обучения. Этапы жизненного цикла программы.	2	2	-	Опрос
1.2	Основы алгоритмизации	2	2	-	Опрос
1.3	Базовые алгоритмические структуры: следование, ветвление	4	1	3	Опрос, наблюдение
1.4	Базовая алгоритмическая структура «цикл»	4	1	3	Опрос, наблюдение
1.5	Разработка алгоритмов с использованием базовых алгоритмических структур	2	-	2	Беседа
1.6	Знакомство со средой PyCharm.	2	2	-	Опрос
1.7	Организация ввода данных в программе. Функция Input. Создание программ линейной структуры	4	2	2	Наблюдение
1.8	Создание программ линейной структуры	6	-	6	Демонстрация решений
<b>2</b>	<b>Базовые конструкции языка Python</b>	<b>70</b>	<b>22</b>	<b>48</b>	
2.1	Условный оператор IF. Создание программ разветвляющейся структуры	4	1	3	Наблюдение
2.2	Операторы циклов while и for.	6	2	4	Опрос, наблюдение
2.3	Создание программ циклической и разветвляющейся структуры	6	-	6	Демонстрация решения
2.4	Строки. Операции со строками. Методы split и join.	6	2	4	Наблюдение
2.5	Списки. Индексы элементов списка. Ввод/вывод элементов списка. Методы для работы со списком.	4	1	3	Наблюдение

2.6	Работа со списком: поиск, замена, перестановка элементов. Сортировка элементов списка	4	1	3	Наблюдение
2.7	Исключения. Механизм поиска и обработки исключительных ситуаций (try/except).	2	1	1	Наблюдение
2.8	Понятие массива. Индексы элементов массива. Ввод/вывод элементов массива.	4	2	2	Наблюдение
2.9	Работа с массивом (вектором, матрицей)	4	2	2	Наблюдение
2.10	Работа с диагональными элементами матрицы. Сортировка элементов массива	6	2	4	Демонстрация решения
2.11	Решение задач по теме «Работа с массивами»	2	-	2	Демонстрация решений
2.12	Кортежи. Множества. Словари.	4	2	2	Наблюдение
2.13	Понятие подпрограммы. Создание процедур и функций.	4	2	2	Наблюдение
2.14	Модульный принцип компоновки программы. Подключение и использование модулей. Создание модулей	4	1	3	Наблюдение
2.15	Работа с внешними библиотеками Python и утилитой pip. Модуль Collections	4	1	3	Демонстрация решения
2.16	Решение задач по теме «Создание модулей, процедур и функций»	2	-	2	Беседа
2.17	Автоматизированное тестирование в Python. Создание unit test	4	2	2	Наблюдение
<b>3</b>	<b>Работа с графикой</b>	<b>22</b>	<b>7</b>	<b>15</b>	
3.1	Методы модуля Graph	4	2	2	Наблюдение
3.2	Решение задач на тему «Рисование простыми примитивами»	6	-	6	Демонстрация решения
3.3	Использование процедур в графике	4	2	2	Наблюдение
3.4	Штриховка	2	1	1	Наблюдение
3.5	Анимация. Создание анимации.	6	2	4	Наблюдение
<b>4</b>	<b>Создание графического интерфейса</b>	<b>26</b>	<b>3</b>	<b>23</b>	
4.1	Кроссплатформенная библиотека для разработки графического интерфейса TKInter	6	3	3	Наблюдение

4.2	Создание приложений с графическим интерфейсом	6	-	6	Демонстрация решения
4.3	Выбор темы, планирование работы над проектом, изучение предметной области, проектирование	4	-	4	Наблюдение
4.4	Разработка итогового проекта	6	-	6	Наблюдение
4.5	Подготовка к защите проектов	2	-	2	Беседа
4.6	Защита итоговых проектов	2	-	2	Демонстрация проектов
	<b>Итого</b>	<b>144</b>	<b>42</b>	<b>102</b>	

## Содержание учебно-тематического плана

### 1. Введение в программирование

#### **Тема 1.1** Вводное занятие.

Теория (2 часа). Инструктаж по ТБ. Введение в программу 1-го года обучения. Этапы жизненного цикла программы.

#### **Тема 1.2** Основы алгоритмизации.

Теория (2 часа). Понятие алгоритма, свойства алгоритма (дискретность, понятность, результативность, массовость, определенность), понятие исполнителя алгоритма. Различные формы представления алгоритмов, правила оформления блок-схем алгоритмов.

#### **Тема 1.3** Базовые алгоритмические структуры: следование, ветвление

Теория (1 час). Базовые алгоритмические структуры: следование, ветвление. Виды ветвлений.

Практика (3 часа). Построение линейных и разветвляющихся алгоритмов в виде блок-схем.

#### **Тема 1.4** Базовая алгоритмическая структура «цикл»

Теория (1 час). Цикл. Виды циклов

Практика (3 часа). Построение линейных, разветвляющихся и циклических алгоритмов в виде блок-схем.

**Тема 1.5** Разработка алгоритмов с использованием базовых алгоритмических структур

Практика (2 часа). Построение алгоритмов в виде блок – схем с использованием всех базовых алгоритмических структур

#### **Тема 1.6** Знакомство со средой PyCharm.

Теория (2 часа). Знакомство со средой PyCharm. Понятия кода, интерпретатора, программы. Интегрированные среды, исполнение кода. Переменные. Константы. Арифметические операторы в Python. Простейшие программы с выводом на экран. Функция Print. Рассмотрение примеров.

#### **Тема 1.7** Организация ввода данных в программе. Функция Input.

Создание программ линейной структуры

Теория (2 часа). Организация ввода данных в программе. Функция Input.

Создание программ линейной структуры. Рассмотрение примеров.

Практика (2 часов). Программирование алгоритмов линейной структуры.

#### **Тема 1.8** Создание программ линейной структуры

Практика (6 часов). Разработка программ линейной структуры.

### **Базовые конструкции языка Python**

**Тема 2.1** Условный оператор IF. Создание программ разветвляющейся структуры.

Теория (1 час). Условный оператор IF. Создание программ разветвляющейся структуры. Рассмотрение примеров.

Практика (3 часа). Программирование алгоритмов разветвляющейся структуры.

#### **Тема 2.2** Операторы циклов While и For.

Теория (2 часа). Операторы циклов While и For. Создание программ циклической структуры. Рассмотрение примеров.

Практика (4 часа). Программирование алгоритмов циклической структуры.

#### **Тема 2.3** Создание программ циклической и разветвляющейся структуры

Практика (6 часов). Решение задач с использованием операторов ветвлений и циклов.

#### **Тема 2.4** Строки. Операции со строками. Методы split и join.

Теория (2 часа). Строки. Операции со строками. Методы split и join

Практика (4 часа). Написание программ на тему «Работа со строками».

**Тема 2.5** Списки. Индексы элементов списка. Ввод/вывод элементов списка. Методы для работы со списком.

Теория (1 час). Списки. Индексы элементов списка. Ввод/вывод элементов списка. Методы для работы со списком.

Практика (3 часа). Решение задач на ввод, вывод элементов списка.

**Тема 2.6** Работа со списком: поиск, замена, перестановка элементов. Сортировка элементов списка.

Теория (1 час). Работа со списком: поиск, замена, перестановка элементов. Сортировка элементов списка.

Практика (3 часа). Решение задач на обработку списков.

**Тема 2.7** Исключения. Механизм поиска и обработки исключительных ситуаций (try/except).

Теория (1 час). Понятие исключения. Механизм поиска и обработки исключительных ситуаций (try/except).

Практика (1 час). Обработка исключений.

**Тема 2.8** Понятие массива. Индексы элементов массива. Ввод/вывод элементов массива.

Теория (2 часа). Понятие массива. Индексы элементов массива. Ввод/вывод элементов массива. Рассмотрение примеров работы с массивом.

Практика (2 часа). Решение задач на ввод, вывод элементов списка.

**Тема 2.9** Работа с массивом (вектором, матрицей): поиск, замена, перестановка элементов. Формирование массива из элементов другого массива

Практика (4 часа) Работа с массивом (вектором, матрицей): поиск, замена, перестановка элементов. Формирование массива из элементов другого массива.

**Тема 2.10** Работа с диагональными элементами матрицы. Сортировка элементов массива

Теория (2 часа). Диагональные элементы матрицы. Алгоритмы сортировок массивов.

Практика (4 часа). Работа с диагональными элементами матрицы.  
Сортировка элементов массива.

**Тема 2.11** Решение задач по теме «Работа с массивами».

Практика (2 часа). Решение задач по теме «Работа с массивами».

**Тема 2.12** Кортежи. Множества. Словари

Теория (2 часа). Кортежи. Словари (dict) и работа с ними. Методы словарей. Множества (set и frozenset).

Практика (2 часа). Решение задач на тему «Словари и множества».

**Тема 2.13.** Понятие подпрограммы. Создание процедур и функций.

Теория (2 часа). Понятие подпрограммы. Процедуры и функции. Правила составления и вызова процедур и функций в программном коде. Локальные и глобальные переменные.

Практика (2 часа). Создание процедур и функций.

**Тема 2.14.** Модульный принцип компоновки программы. Подключение и использование модулей. Создание модулей

Теория (1 часа). Понятие модульного программирования. Понятие модуля. Подключение модуля инструкциями import и from. Создание модуля.

Практика (3 часов). Создание модулей.

**Тема 2.15.** Работа с внешними библиотеками Python и утилитой pip.  
Модуль Collections

Теория (1 часа). Внешние библиотеки Python, утилита pip. Модуль Collections.

Практика (3 часов). Работа с внешними библиотеками Python и утилитой pip. Модуль os,sys. Модуль abc. Модуль datetime. Модуль Collections.

**Тема 2.16** Решение задач по теме «Создание модулей, процедур и функций»

Практика (2 часа). Решение задач по теме «Создание модулей, процедур и функций».

**Тема 2.17** Автоматизированное тестирование в python. Создание unit test.

Теория (2 часа). Автоматизированное тестирование в python.

Практика (2 часа). Создание unit test.

### **3. Работа с графикой**

**Тема 3.1** Методы модуля Graph.

Теория (2 часа). Модуль Graph. Основные методы модуля.

Практика (2 часов). Решение задач с использованием основных методов модуля Graph.

**Тема 3.2** Решение задач на тему «Рисование простыми примитивами»

Практика (6 часов). Решение задач на тему «Рисование простыми примитивами».

**Тема 3.3** Использование процедур в графике.

Теория (2 часа). Разбор примеров использования процедур в графике.

Практика (2 часа). Решение задач на тему «Использование процедур в графике».

**Тема 3.4** Штриховка.

Теория (1 час). Разбор примеров использования метода штриховки при рисовании.

Практика (1 час). Решение задач на тему «Штриховка».

**Тема 3.4** Анимация. Создание анимации

Теория (2 часа). Понятие анимации. Алгоритм реализации анимации. Рассмотрение примеров.

Практика (4 часа). Решение задач на тему «Анимация».

### **4. Создание графического интерфейса**

**Тема 4.1** Кроссплатформенная библиотека для разработки графического интерфейса TKInter

Теория (3 часа). Кроссплатформенная библиотека для разработки графического интерфейса TKInter.

Практика (3 часа). Решение задач на тему «Модуль TKInter».

**Тема 4.2** Создание приложений с графическим интерфейсом

Практика (6 часов). Разработка приложений с графическим интерфейсом.



**Тема 4.3** Выбор темы, планирование работы над проектом, изучение предметной области, проектирование

Практика (4 часа). Выбор темы, планирование работы над проектом, изучение предметной области, проектирование

**Тема 4.4** Разработка итогового проекта

Практика (6 часов). Создание программы для итогового проекта.

**Тема 4.5** Подготовка к защите проектов

Практика (2 часа). Обсуждение плана выступления защиты индивидуального проекта.

**Тема 4.6** Защита индивидуальных итоговых проектов.

Практика (2 часа). Защита индивидуальных итоговых проектов.

#### 4. ФОРМЫ АТТЕСТАЦИИ И ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

Формы аттестации: наблюдение, тестирование, защита проекта, демонстрация решения, беседа, опрос.

Входной контроль осуществляется в начале реализации программы в форме опроса и имеет диагностические задачи. Цель входной диагностики – зафиксировать начальный уровень подготовки обучающихся, имеющиеся знания, умения и навыки, связанные с предстоящей деятельностью.

Наблюдение осуществляется в течение реализации программы.

Промежуточный контроль осуществляется в целях диагностики теоретических знаний и практических умений и навыков по итогам освоения одного из разделов курса. Проводится в форме демонстрации решений по теме.

Итоговый контроль проводится с целью определения степени достижения результатов обучения и получения сведений для совершенствования программы и методов обучения – представляет из себя защиту проекта.

##### Критерии оценки проекта

№	Название критерия	Максимальный балл
1.	Актуальность и проработанность проблемы	До 5 баллов
2.	Четкость формулировки целей и задач	До 5 баллов
3.	Технологическая сложность проекта	До 5 баллов
4.	Новизна и оригинальность решения	До 5 баллов
5.	Качество разработанного продукта	До 5 баллов
5.	Защита проекта: <ul style="list-style-type: none"><li>• качество презентации;</li><li>• четкость и ясность изложения, умение взаимодействовать с аудиторией, отвечать на вопросы</li></ul>	До 5 баллов
6.	Наличие самооценки и перспектив дальнейшей разработки проекта	До 5 баллов
7.	Умение работать в команде	До 5 баллов
Итого		35 баллов

Анализ результатов итоговой аттестации – защиты проекта.

Высокий уровень – учащийся набрал не менее 28 баллов по итогам защиты проекта.

Средний уровень – учащийся набрал от 17 до 27 баллов по итогам защиты проекта.

Низкий уровень – учащийся набрал менее 17 баллов по итогам защиты проекта.

Общими критериями оценки результативности обучения являются:

- оценка уровня теоретических знаний: свобода восприятия теоретической информации, осмысленность и свобода использования специальной терминологии, свобода ориентации в теоретическом материале;
- оценка уровня практической подготовки: соответствие развития уровня практических умений и навыков программным требованиям, свобода владения специальным оборудованием и оснащением;
- оценка уровня достижения личностных результатов: культура организации самостоятельной деятельности, культура работы с информацией, аккуратность и ответственность при работе.

Оценка итоговых результатов освоения программы осуществляется по трем уровням:

Высокий уровень – достижение 80- 100% показателей освоения программы.

Средний уровень – достижение 50- 79% показателей освоения программы.

Низкий уровень - достижение менее чем 50% показателей освоения программы.

Достигнутые обучающимся знания, умения и навыки заносятся в сводную таблицу результатов обучения.

### Оценка уровней освоения программы

Уровни	Параметры	Показатели
<b>Высокий уровень (80-100%)</b>	Теоретические знания	Обучающийся освоил материал в полном объеме. Знает и понимает значение терминов, употребляет их осознанно и в полном соответствии с содержанием. Самостоятельно ориентируется в содержании материала по темам.
	Практические умения и навыки	Обучающийся овладел 80-100% умений и навыков, предусмотренных программой за конкретный период. Умет работать самостоятельно, применяя

		<p>практические умения и навыки.  Правильно и по назначению применяет инструменты.  Умеет выполнять основные логические действия (анализ, синтез, установление причинно-следственных связей).  Способен планировать и регулировать свою деятельность по реализации проекта.  Умеет осуществлять поиск информации, в том числе в сети Интернет; выслушивать собеседника и вести диалог; выражать свои мысли в соответствии с задачами и условиями коммуникации.</p>
	Личностные результаты	<p>Обучающийся обладает внутренней мотивацией.  Способен самостоятельно организовать собственную деятельность.  Сформирована культура работы с информацией.  Работу выполняет аккуратно, доводит до конца.  Может оценить результаты выполнения своего задания и дать оценку работы своего товарища.</p>
<b>Средний уровень (50-79%)</b>	Теоретические знания	<p>Учащийся освоил базовые знания, ориентируется в содержании материала по темам, иногда обращается за помощью к педагогу.  Использует специальную терминологию, однако сочетает её с бытовой</p>
	Практические умения и навыки	<p>Владеет базовыми навыками и умениями, но не всегда может выполнить задание самостоятельно, просит помощи педагога.  В основном выполняет задания на основе образца.  Способен разработать проект с помощью преподавателя.  Встречаются отдельные случаи неправильного применения инструментов.  Делает ошибки в работе, но может устранить их после наводящих вопросов или самостоятельно  Испытывает незначительные затруднения при выполнении основных логических действий (анализ, синтез, установление причинно-следственных связей).  Способен планировать и регулировать свою деятельность по реализации проекта с помощью педагога.  Испытывает незначительные сложности в осуществлении коммуникации с педагогом и сверстниками.</p>
	Личностные результаты	<p>Внутренняя мотивация к обучению сочетается с внешней.  В работе допускает небрежность.  Работу не всегда выполняет аккуратно и/или доводит до конца.  Оценить результаты своей деятельности может с подсказкой педагога.</p>
<b>Низкий уровень (меньше)</b>	Теоретические знания	<p>Владеет минимальными знаниями, ориентируется в содержании материала по темам только с помощью педагога.  Избегает употреблять специальные термины.</p>

50%)	Практические умения и навыки	<p>Владеет минимальными начальными навыками и умениями.</p> <p>Учащийся способен выполнять каждую операцию только с подсказкой педагога или товарищей.</p> <p>Часто неправильно применяет необходимый инструмент или не использует его вовсе. В работе допускает грубые ошибки, не может их найти даже после указания преподавателя.</p> <p>В состоянии выполнять лишь простейшие практические задания педагога.</p> <p>Испытывает существенные затруднения при выполнении основных логических действий (анализ, синтез, установление причинно-следственных связей).</p> <p>Не способен планировать и регулировать свою деятельность по реализации проекта.</p> <p>Испытывает значительные сложности в осуществлении коммуникации с педагогом и сверстниками.</p>
	Личностные результаты	<p>Преобладает внешняя мотивация к обучению.</p> <p>Работу часто выполняет неаккуратно и/или не доводит до конца.</p> <p>Не способен самостоятельно и объективно оценить результаты своей работы.</p>

### Сводная таблица результатов обучения по программе

№ п/п	Фамилия, имя обучающегося	Оценка теоретических знаний	Оценка практических умений и навыков (предметных и метапредметных)	Личностные результаты	Итоговая оценка
1.					
2.					
3.					

### Примеры заданий для проведения промежуточной аттестации

#### 1. Решение задач по теме «Базовые алгоритмические структуры»

Задание №1: Разработайте алгоритм, записав его в виде блок-схемы, для вычисления значения функции  $N$ :

$$N = \begin{cases} (k - c), & \text{если } 5 \leq a < 11 \\ (k + c), & \text{во всех остальных случаях} \end{cases}$$

Задание №2: Разработайте алгоритм, записав его в виде блок-схемы, для вычисления суммы всех чисел от 3 до 50.

## 2. Решение задач по теме «Создание программ линейной структуры»

Задание: Услуги телефонной сети оплачиваются по следующему правилу: за 1 минуту разговора необходимо заплатить  $A$  рублей. Составьте программу, вычисляющую плату за пользование телефоном для введенного времени разговора.

## 3. Решение задач по теме «Создание программ циклической и разветвляющейся структуры»

Задание №1: Даны три стороны треугольника. Необходимо определить, существует ли такой треугольник. Если «да», то уточнить: является ли он равносторонним или равнобедренным.

Задание №2: Напишите программу, позволяющую найти сумму всех четных чисел из указанного пользователем диапазона.

## 4. Решение задач по теме «Работа с массивами»

Задание №1: Организуйте ручное заполнение целочисленного массива  $A$ , размерностью 8. Найдите и выведите на экран количество нулевых элементов вектора и сумму элементов, равных числу  $T$ . Выведите на экран массив  $A$ .

Задание №2: Организуйте заполнение целочисленного двумерного квадратного массива случайными числами. Замените все элементы матрицы, находящиеся ниже главной диагонали на максимальный элемент строки  $S$ . Выведите на экран матрицу до и после изменения.

## 5. Решение задач по теме «Создание модулей, процедур и функций»

Задание: Создайте модуль с функциями сложения и умножения двух целых чисел. Напишите программу, позволяющую вычислять и выводить на экран результат сложения или перемножения двух целых чисел, введенных пользователем, по выбору пользователя. Используйте предварительно созданный модуль.

## **Примеры тем итоговых проектов**

### 1. Разработка игры «Змейка» с графическим интерфейсом

2. Создание анимационного логотипа

3. Создание анимационного графического приложения

## **5. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ**

### **Материально-техническое обеспечение**

Рабочее место учащегося:

- ноутбук с выходом в сеть Интернет.

Рабочее место наставника:

- ноутбук с выходом в сеть Интернет;
- технические средства обучения (ТСО) (мультимедийное устройство).

### **Методическое обеспечение программы**

Методы обучения:

- объяснительно-иллюстративный (беседы, объяснения);
- репродуктивный (деятельность обучаемых носит алгоритмический характер, выполняется по инструкциям, предписаниям, правилам в аналогичных, сходных с показанным образцом ситуациях);
  - метод проблемного изложения;
  - эвристический (метод обучения заключается в организации активного поиска решения выдвинутых в обучении (или самостоятельно сформулированных) познавательных задач в ходе подготовки и реализации творческих проектов);
  - исследовательский.

Педагогические технологии: проектная технология, здоровьесберегающие технологии, технология проблемного обучения.

Проектная технология дает возможность самостоятельно конструировать свои знания, ориентироваться в информационном пространстве, развивает критическое и творческое мышление, создаёт условия для формирования и развития внутренней мотивации учащихся к более качественному овладению знаниями, повышения мыслительной активности и приобретения навыков логического мышления.

Здоровьесберегающие технологии позволяют создать максимально возможные условия для сохранения, укрепления и развития эмоционального,



интеллектуального и физического здоровья, в том числе в условиях работы с компьютерной техникой.

Проблемное обучение — это тип развивающего обучения, содержание которого представлено системой проблемных задач различного уровня сложности, в процессе решения которых учащиеся овладевают новыми знаниями и способами действия, а через это происходит формирование творческих способностей: продуктивного мышления и познавательной мотивации.

#### **Учебно-методические средства обучения:**

- специализированная литература по направлению, инструкции;
- образцы выполненные обучающимися и педагогом;
- учебно-методические пособия для педагога и обучающихся, включающие дидактический, информационный, справочный материалы.

Образовательные технологии: проектная технология, здоровьесберегающие технологии, технология проблемного обучения.

#### **Кадровое обеспечение**

Программу реализуют педагоги структурного подразделения Центр цифрового образования детей «IT-куб».

## **6. СПИСОК ЛИТЕРАТУРЫ**

### **Список литературы для педагогов**

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. Лутц М. Изучаем Python, пер. с англ. 3-е изд. — СПб.: Символ Плюс, 2009. — 848 с.
4. Рафгарден Т. Совершенный алгоритм. Жадные алгоритмы и динамическое программирование. — СПб.: Питер, 2020. — 256 с.
5. Рейтц К., Шлюссер Т. Автостопом по Python. — СПб. : Питер, 2017. — 336 с.
6. Фёдоров Д. Ю. Программирование на языке высокого уровня Python: учебное пособие для прикладного бакалавриата. — М. : Издательство Юрайт, 2019. — 161 с.

### **Список литературы для обучающихся**

1. Бхаргава А. Грокаем алгоритмы: иллюстрированное пособие для программистов и любопытствующих. — СПб.: Питер, 2017. — 288 с.
2. Гэддис Т. Начинаем программировать на Python / пер. с англ. 4-е изд. — СПб.: БХВ-Петербург, 2019. — 768 с.
3. Луридас П. Алгоритмы для начинающих: теория и практика для разработчика. — М. : Эксмо, 2018. — 608 с.
4. Мюллер Дж. Python для чайников. — СПб.: Диалектика, 2019. — 416 с.