

Индивидуальный предприниматель Майоров Рустем Марванович

«Утверждаю»

Индивидуальный предприниматель  
Майоров Р.М.

« 04 » 10 2018 г.

Приказ № 10 от 04.10.2018 г.

«Рассмотрено»

педагогическим советом

ЦРШ «Код Успеха»

Протокол № 1 от 04.10.2018 г.

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

Возраст обучающихся: 13-16 лет

Срок реализации: 1 год

Форма обучения – очная

Уровень программы - базовый

Автор-составитель:

Майоров Рустем Марванович

г. Ульяновск, 2018

## Оглавление

1.	Пояснительная записка.....	3
1.1.	Направленность.....	3
1.2.	Актуальность программы.....	3
1.3.	Педагогическая целесообразность.....	4
1.4.	Цель и задачи программы.....	4
1.5.	Планируемые результаты.....	5
1.6.	Отличительные особенности программы.....	7
1.7.	Условия реализации программы.....	7
1.8.	Формы организации занятий.....	8
2.	Календарный учебный график.....	8
3.	Учебный план.....	8
4.	Содержание программы.....	9
5.	Оценочные и методические материалы.....	12
5.1.	Формы аттестации.....	12
5.2.	Методическое обеспечение.....	12
6.	Материально-техническое обеспечение программы.....	14
6.1.	Перечень необходимого оборудования.....	14
6.2.	Учебно-методическое обеспечение.....	15
7.	Кадровое обеспечение.....	15
8.	Список литературы.....	15
8.1.	Литература для педагога.....	15
8.2.	Литература для учащихся.....	16
8.3.	Интернет-ресурсы.....	16

## 1 Пояснительная записка

Дополнительная общеобразовательная программа «Программирование на языке Python» разработана согласно требованиям следующих нормативных документов:

- Федеральный Закон «Об образовании в Российской Федерации» от 29.12.2012 № 273-ФЗ.
- Концепция развития дополнительного образования детей (утверждена распоряжением Правительства РФ от 04.09.2014 № 1726-р).
- СанПин к устройству, содержанию и организации режима работы образовательных организаций дополнительного образования детей (утверждено постановлением Главного государственного санитарного врача РФ от 04.07.2014 № 41)
- Порядок организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам (утвержден приказом Министерства просвещения РФ от 9 ноября 2018 г. N 196).
- Стратегия развития воспитания в Российской Федерации на период до 2025 года (утверждена Распоряжением Правительства Российской Федерации от 29 мая 2015 г. N 996-р)

В школьном курсе информатики вопросы программирования рассматриваются лишь в ознакомительном плане, и на это выделяется недостаточное количество часов, как следствие – формальное восприятие учащимися основ современного программирования. Дополнительная общеобразовательная программа «Программирование на языке Python» направлена на устранение данного пробела. Это особенно актуально в условиях построения цифровой экономики, при высокой востребованности в программистах.

### Направленность

Направленность дополнительной образовательной программы «Программирование на языке Python» – научно-техническая. Она является прикладной, носит практико-ориентировочный характер и направлена на овладение воспитанниками основных приемов программирования. Обучение по данной программе создает благоприятные условия для интеллектуального и духовного воспитания личности ребенка, социально - культурного и профессионального самоопределения, развития познавательной активности и творческой самореализации учащихся.

### Актуальность программы

Данная дополнительная общеобразовательная программа соотносится с тенденциями развития дополнительного образования и согласно Концепции развития дополнительного образования способствует:

- созданию необходимых условий для личностного развития обучающихся, позитивной социализации и профессионального самоопределения;

- удовлетворению индивидуальных потребностей обучающихся в интеллектуальном развитии, а также в занятиях научно-техническим творчеством;
- формированию и развитию творческих способностей учащихся, выявлению, развитию и поддержке талантливых учащихся;
- обеспечению духовно-нравственного, гражданского, патриотического, трудового воспитания учащихся;

Кроме того, актуальность программы обуславливается тем, что при обычном обучении информатике, темы «алгоритмы» и «программирование» изучаются очень мало и поздно, это замедляет формирование алгоритмического мышления, не способствует развитию интереса учащихся в области программирования, учащиеся, как правило, не готовы успешно выступать на олимпиадах по информатике, теряют интерес к предмету. Обучение по данной программе создает благоприятные условия для более раннего «погружения» учащихся в мир логики, математического моделирования, для интеллектуального и духовного воспитания личности ребенка, социально - культурного и профессионального самоопределения, развития познавательной активности и творческой самореализации учащихся. Она рассчитана на сотворчество и сотрудничество педагога и воспитанников. Данная программа дает возможность детям творчески мыслить, находить самостоятельные индивидуальные решения, а полученные умения и навыки применять в жизни. Развитие творческих способностей помогает также в профессиональной ориентации подростков.

### **Педагогическая целесообразность**

Педагогическая целесообразность заключается в том, что сделанный в представленной программе акцент на практическую работу позволяет повысить результативность обучения информатике и ИКТ при совместном преподавании школьного основного курса и данного дополнительного, расширить мировоззрение учащихся, повысить предметные и межпредметные ЗУН и УУН, подготовки учащихся успешно освоить учебный материал и участвовать в олимпиадах, осознанного выбора профиля дальнейшего обучения и будущей профессии.

### **Цель и задачи программы**

**Цель программы:** развитие творческих способностей учащихся, овладение теоретическими и практическими навыками решения алгоритмических задач программными средствами.

#### **Задачи:**

##### образовательные:

обучить основным видам вычислительных процессов и способам их реализации с помощью синтаксических конструкций языка программирования Python;

обучить приемам работы в основных интегрированных средах программирования (IDLE, Wing, Pycharm);

сформировать навыки тестирования, поиска и исправления ошибок в программном коде;

воспитательные:

воспитывать культуру написания «чистого», понятного для других программистов, кода, соблюдения рекомендаций по выбору имен для идентификаторов;

воспитать культуру проектной деятельности, в том числе умение планировать, работать в коллективе; чувство ответственности за результаты своего труда, используемого другими людьми;

формировать установку на позитивную социальную деятельность в информационном обществе, недопустимость действий, нарушающих правовые и этические нормы работы с информацией;

содействовать повышению привлекательности науки, научно-технического творчества для подрастающего поколения;

формировать у подрастающего поколения ответственное отношение к своему здоровью и потребности в здоровом образе жизни, соблюдении правил гигиены;

развивающие:

быстро ориентироваться в динамично развивающемся и обновляющемся информационном пространстве; получать, использовать и создавать разнообразную информацию; принимать обоснованные решения и решать проблемы на основе полученных знаний, умений и навыков;

овладение умениями строить математические объекты, в том числе логические формулы и программы на формальном языке, удовлетворяющие заданному описанию; использовать общепользовательские инструменты и настраивать их для нужд пользователя; применять алгоритмы и приёмы программирования;

развитие алгоритмического мышления, способностей к формализации, элементов системного мышления;

приобретение опыта создания, редактирования, оформления, сохранения, передачи информационных объектов различного типа с помощью современных программных средств; построение компьютерных моделей, коллективной реализации информационных проектов, преодоление трудностей в процессе интеллектуального проектирования, информационной деятельности в различных сферах, востребованных на рынке труда; решения сложных задач и олимпиадных задач программирования.

## Планируемые результаты

Планируемые результаты обучения по дополнительной общеобразовательной программе «Программирование на языке Python»:

**Предметные:**

- умение анализировать поставленные задачи и создавать соответствующие информационные модели;
- умение разрабатывать алгоритмы решения поставленных задач и выбирать оптимальные структуры данных;
- умение применять основные конструкции языка Python для реализации алгоритмов решения поставленных задач;
- умение тестировать и выполнять отладку программного кода, применять инструментальные средства отладки.

**Личностные:**

- чувство патриотизма, уважения к своему народу, гордости за свой край;
- ответственное отношение к труду, качеству своей деятельности;
- овладение основами саморазвития и самовоспитания в соответствии с общечеловеческими ценностями и идеалами гражданского общества; готовность и способность к самостоятельной, творческой и ответственной деятельности;
- умение общаться и сотрудничать со сверстниками, детьми старшего и младшего возраста, взрослыми в процессе образовательной, общественно полезной, творческой и других видов деятельности;
- навыки сотрудничества с взрослыми и сверстниками в разных социальных ситуациях, умение не создавать конфликтов и находить выходы из спорных ситуаций;
- сформированность познавательных интересов и творческая активность в данной области.

**Метапредметные:**

- умение определять и формулировать цель деятельности на занятии, проговаривать последовательность действий с помощью педагога;
- умение планировать, контролировать и оценивать свои действия в соответствии с поставленной задачей и условиями её реализации, определять наиболее эффективные способы достижения результата;
- навыки самостоятельного выполнения практических работ по предложенному плану с опорой блок-схемы;
- способность самостоятельно оценивать и принимать решения, определяющие стратегию поведения, с учетом гражданских и нравственных ценностей;
- умение понимать причины успеха/неуспеха учебной деятельности и способности конструктивно действовать даже в ситуациях неуспеха.

**В результате обучения выпускники должны знать:**

- Основные типы данных языка программирования Python;
- Основные конструкции языка программирования Python;
- Обозначения, применяемые для оформления блок-схем алгоритмов;
- Порядок применения инструментальных средств для разработки программ на языке Python;

- Порядок применения инструментальных средств для отладки программ. **должны уметь:**
- Выполнять блок-схемы алгоритмов;
- Составлять программный код, реализующий тот или иной алгоритм;
- Определять необходимые для решения задачи данные структуры данных;
- Оформлять программный код с соответствием с общепринятыми правилами;
- Осуществлять поиск исходных данных, приводящих к неверному результату, определять причину неверного результата и принимать решение по оптимальному исправлению ошибок в программном коде;
- Работать в команде над решением задачи;
- Проявлять чувство коллективизма, взаимопомощи, взаимопонимания;
- Уважать труд своих товарищей;
- Стремиться к совершенствованию своих навыков.

### Отличительные особенности программы

Отличительные особенности данной дополнительной программы от существующих образовательных программ в том, что она содержит дополнительный изучаемый материал, значительно расширяет возможности формирования универсальных учебных и предметных навыков. В программе большое внимание уделяется практическим заданиям. Проверка заданий осуществляется комбинированным способом. На первом этапе автоматически с помощью тестирующей системы. На этом этапе отсекается большая часть неверных решений. Второй этап контроля – проверка посланного кода преподавателем. На этом этапе проверяется, выполнены ли ограничения по используемым конструкциям языка, оптимальность решения и правильность оформления кода. Допущенные ошибки разбираются на занятии. Такой подход приучает детей к самостоятельной работе, освобождает преподавателя от рутинной проверки заведомо неверных решений, тем самым повышая эффективность учебного процесса.

Все учебные материалы представлены в формате презентаций. Их удобно использовать с помощью мультимедийного проектора. Обучаемые имеют к ним постоянный доступ в сети Интернет и могут использовать их в качестве опорного конспекта.

### Условия реализации программы

В освоении настоящей учебной программы участвуют дети в возрасте 13-16 лет. Условиями отбора в группу для детей 13-14 лет являются уверенные навыки решения алгоритмических задач в средах визуального программирования (например Scratch). Для детей в возрасте 15-16 лет наличие начальных навыков работы на ПК.

Форма обучения – очная.

Формирование групп целесообразно осуществлять с небольшим разбросом возраста и со схожим уровнем навыков решения олимпиадных задач по

математике. Оценка указанного уровня возможна на основании рекомендаций педагогов, ведущих математику или на основе собеседования.

### Формы организации занятий

Основной формой организации деятельности детей на занятии является групповое занятие. При изучении теоретических занятий педагог объясняет учебный материал с использованием слайдов и магнитно-маркерной доски.

Закрепление изученного материала осуществляется самостоятельной работой с помощью педагога по решению задач на компьютере с использованием специализированного сайта [test.kod-u.ru](http://test.kod-u.ru).

Периодически проводятся олимпиады, предполагающие самостоятельную работу по решению задач на компьютере без помощи педагога.

После олимпиад, как правило, педагогом осуществляется разбор задач. Для разбора могут привлекаться обучаемые, успешно решившие те или иные задачи в ходе олимпиады.

## 2 Календарный учебный график

Режим организации занятий по данной дополнительной общеобразовательной программе определяется календарным учебным графиком и соответствует нормам, утвержденным «СанПин к устройству, содержанию и организации режима работы образовательных организаций дополнительного образования детей» № 41 от 04.07.2014 (СанПин 2.4.43172 -14, пункт 8.3, приложение № 3).

Начало занятий первого года обучения – 1 сентября.

Окончание занятий первого года обучения – 31 мая.

Всего учебных недель (продолжительность учебного года) – 34 недели.

Количество учебных дней: – 68 дней.

Объем учебных часов:– 136 часов.

Режим работы: 2 раза в неделю по 2 часа.

Продолжительность учебного времени: 2 урока по 45 мин

Перерыв между уроками – 10 мин.

## 3 Учебный план

№ п/п	Название раздела, темы	Количество часов			Формы аттестации, контроля
		Всего	Теория	Практика	
1.	Введение	2	2		
2.	Основные понятия	2	2		
3.	Вычисления	10	0	10	Текущий контроль
4.	Условная инструкция	10	2	8	Текущий контроль
5.	Цикл while	10	1	9	Текущий



6.	Цикл for	10	1	9	Текущий контроль
7.	Списки	10	1	9	Текущий контроль
8.	Линейные алгоритмы	10	1	9	Текущий контроль
9.	Строки	10	2	8	Текущий контроль
10.	Матрицы	12	1	11	Текущий контроль
11.	Методы строк и операции над строками	8	2	6	Текущий контроль
12.	Функции	8	2	6	Текущий контроль
13.	Встроенная сортировка	6	1	5	Текущий контроль
14.	Файлы	10	1	9	Текущий контроль
15.	Множества	8	1	7	Текущий контроль
16.	Словари	10	1	9	Текущий контроль
	<b>Итого</b>	<b>136</b>			

## 4 Содержание программы

### Тема 1. Введение

#### Цели и задачи:

Показать обучаемым место изучаемого материала в общей системе информационных технологий. Дать минимально необходимые сведения для начала работы.

#### Результаты освоения:

Умение запускать интегрированную среду разработки в режимах калькулятора и редактора кода. Создание первой работающей программы.

#### Краткое содержание занятий:

Применение компьютеров, алгоритмы, устройство ЭВМ, машинные команды, взаимодействие программ, компиляция и интерпретация, понятие языка программирования, среда IDLE, режим калькулятора и запуск файлов. Работа в

режиме калькулятора: повторный запуск команд. Функция `print()`, Практика: Hello world.

## Тема 2. Основные понятия

### Цели и задачи:

Ввести основные понятия языка программирования и показать взаимосвязь между ними. Изучить целочисленное деление и нахождение остатка от него.

### Результаты освоения:

Знать основные понятия языка. Уметь строить и разбирать выражения. Понимать различие между типами данных. Уметь решать задачи на целочисленное деление.

### Краткое содержание занятий:

Вычисления, переменная (Модель «коробки»), присвоение, Множественное присваивание. Обмен значений переменных. Последовательное присвоение (цепочка). Идентификаторы. Резервированные слова. Взаимозаменяемость переменных и значений.

Понятие функции (Пример `min`). Стандартные функции: `abs`. Подпрограмма, скобки для запуска, без скобок - взятие адреса функции, понятие параметра (аргумента), возврат значения, пример `input()`. Термины оператор, операция, операнд. Выражение. Использование значений, переменных и функций в выражениях. Практика: Ввод двух чисел и сложение. Понятие типа. Типы `str`, `int`, `float`. Преобразование типов. Вложенный вызов функций. (`int+int`, `str+str`, `int+str`). `int()` для округления. Ввод нескольких значения. Комментарии в программах.

Тип результата в разных операциях. Использование в циклических процессах операций `//` и `%` (неполное частное и взятие по модулю). Увеличение числа на 1. Сокращенная операция `+=`.

## Тема 3. Вычисления

### Цели и задачи:

Изучить особенности работы с тестирующей системой, закрепить знания по применению темы целочисленное деление.

### Результаты освоения:

Умение самостоятельной работы в тестирующей системе.

### Краткое содержание занятий:

Важность самостоятельной работы (не гуглить и работать дома). Регистрация на сайте. Поиск курса и добавление в избранное. Доступ к слайдам. По-

рядок сдачи задач. Вердикты проверяющей системы. Мои посылки. Разбор задачи МКАД на примере развертки на оси движения по МКАД и взятия шариков из коробки. Взятие по модулю от отрицательного числа.

#### Тема 4. Условная инструкция

##### Цели и задачи:

Познакомиться с видами ветвящихся процессов и изучить способы их реализации. Изучить логический тип данных.

##### Результаты освоения:

Умение применять для решения задач условные инструкции, строить комбинированные логические выражения.

##### Краткое содержание занятий:

Последовательность выполнения инструкций. Ветвления. Условная инструкция. Любой тип, как логическое выражение. Использование отступов. Блок схема. Условия. Тип bool. Вычисление логических выражений. else. Вложенные условия. Каскадные условия. Пошаговая отладка программ в IDE.

#### Тема 5. Цикл while

##### Цели и задачи:

Познакомиться с циклическими вычислительными процессами, изучить синтаксис и порядок работы инструкции while.

##### Результаты освоения:

Умение применять для решения задач инструкцию while.

##### Краткое содержание занятий:

Необходимость повторения действий. Структура циклов. Порядок работы цикла while. Зацикливание и его предотвращение.

#### Тема 6. Цикл for

##### Цели и задачи:

Изучить синтаксис и порядок работы инструкции for, изучить функцию range.

##### Результаты освоения:

Умение применять для решения задач инструкцию for и функцию range.

##### Краткое содержание занятий:

Понятие итерируемого объекта. Цикл for как перебор элементов итерируемого объекта. Функция range - создание итерируемого объекта из последовательности целых чисел. Else в циклах.

## Тема 7. Списки

### Цели и задачи:

Изучить принципы организации списков, способы их создания, в том числе с помощью генераторов. Изучить методы списков. Получить представление о понятии объекта.

### Результаты освоения:

Умение применять списки для решения задач, умение использовать генераторы для создания списка.

### Краткое содержание занятий:

Списки - аналогия со строками: Состоит из элементов, индексация элементов (в т.ч. отрицательная), срезы (Разная роль квадратных скобок: задание списка и индексы и срезы). Различия со строками: что может быть элементом списка. (нет find, возможность присвоения элементу). Понятие объекта. Встроенные средства справки dir и help. Методы: append, count, index, insert, pop, remove. Уточнение модели переменной (не ящик, а ссылка) вывод: может быть две ссылки на один ящик. Пример с копированием списков. Генераторы списков. Удаление и добавление элементов с помощью присвоения срезу. Изменяемые и неизменяемые объекты. Преобразования в список (строка по буквам, split, join). Распаковка списка при передаче в функцию.

## Тема 8. Линейные алгоритмы

### Цели и задачи:

Изучить простейшие линейные алгоритмы.

### Результаты освоения:

Умение применять для решения задач линейные алгоритмы. Умение программно реализовать изученные алгоритмы.

### Краткое содержание занятий:

Почему называются линейными. Нахождение суммы списка. Нахождение минимума и максимума. Различные начальные кандидаты: индекс первого элемента предпочтительно, начальное значение, когда невозможно по индексу. Поиск элемента в списке. В несортированном списке: первого и последнего вхождения, поиск с барьером (задача шеренга). Поиск в отсортированном списке: досрочное прерывание, бинарный поиск будет изучаться позже.

## Тема 9. Строки

### Цели и задачи:

Изучить способы задания строк, порядок работы с индексами и срезами строк.

**Результаты освоения:**

Умение применять строки для представления информации. Умение манипулировать с фрагментами строк.

**Краткое содержание занятий:**

Аналогия со списками: Состоит из элементов, индексация элементов (в т.ч. отрицательная), срезы. Различия со списками: что может быть элементом списка. Неизменяемость: невозможность присвоения элементу. Задание строк. Использование кавычек. Обращение к символу по индексу. Срезы. in, сцепление. Экранирование символов. Служебные символы, таблица кодировки, вывод обратного следа. Управление в функции print() (sep, end). Понятие объекта и метода. Способ получения справки по методам. Метод format.

**Тема 10. Матрицы****Цели и задачи:**

Изучить порядок представления информации в виде матриц, способы создания матриц.

**Результаты освоения:**

Знать область применения матриц. Уметь создавать матрицы и использовать их для решения задач.

**Краткое содержание занятий:**

Матрица, как двумерный список. Создание двумерных списков. Неправильный способ. Заполненный нулями и ввод с клавиатуры. Сравнение кортежей со списками. Именованный кортеж.

**Тема 11. Методы строк и операции над строками****Цели и задачи:**

Изучить методы объекта типа str. Изучить функции, работающие со строками.

**Результаты освоения:**

Умение применять методы строк для выполнения различных манипуляций со строками и их фрагментами.

**Краткое содержание занятий:**

Повторение понятия объект. Функции и операторы для работы со строками. Обзор методов.

## Тема 12. Функции

### Цели и задачи:

Изучить принципы построения и порядок создания пользовательских функций. Изучить различные способы передачи информации в функцию.

### Результаты освоения:

Знание структуры функции, порядка передачи и возврата данных, различия между глобальными и локальными переменными. Умение выделять границы функции и проектировать ее интерфейс.

### Краткое содержание занятий:

Для чего нужны функции. Виды функций. (методы, лямбда). Оформление функций, рекомендации по именованию. Параметры функции (позиционные, именованные), значения по умолчанию. Распаковка параметров. Различия передачи переменной и списка в качестве параметра. Возврат значения функции. Проектирование функций (универсальность, интерфейс). Локальные и глобальные переменные. Оформление основной программы в виде функции `main()` (Вопросы быстрого действия). Рекурсия. Условие завершения рекурсии. Ограничение на количество вызовов рекурсии в Python.

## Тема 13. Встроенная сортировка

### Цели и задачи:

Научиться применять встроенные функцию и метод сортировки для списков структур.

### Результаты освоения:

Умение решать задачи, требующие сортировки данных, представляемых с помощью списков структур.

### Краткое содержание занятий:

Понятие сортировки. В основе сортировки – сравнение. Функция `sorted` и метод `sort`. Сортировка по убыванию. Устойчивость сортировки. Сортировка матрицы. Пример – турнирная таблица. Функция, возвращающая сравниваемое значение. Именованный параметр `key`.

## Тема 14. Файлы

### Цели и задачи:

Получить представление об области применения файлов и концепции последовательного доступа к информации. Изучить порядок работы с файлами.

**Результаты освоения:**

Умение использовать файловый ввод-вывод для решения широкого круга задач.

**Краткое содержание занятий:**

Назначение файлов в информационных системах. Отличие внешней памяти от оперативной. Особенность работы с файлами – последовательное чтение. Применение файлов в олимпиадах школьного и муниципального этапов. Понятие файлового объекта. Открытие файла. Способы чтения из файла. Способы записи в файл. Чтение файла циклом `for`. Перенаправление стандартных потоков ввода-вывода в файлы.

**Тема 15. Множества****Цели и задачи:**

Изучить действия, выполняемые над множествами. Дать представление об устройстве и принципе работы множеств.

**Результаты освоения:**

Умение выделять задачи, решение которых эффективно с использованием множеств, применять множества для решения таких задач.

**Краткое содержание занятий:**

Классификация коллекций. Деление на изменяемые и неизменяемые типы. Понятие множества. Устройство оперативной памяти, адресация. Вычисление адреса по индексу массива. Отличие списка от массива с точки зрения адресации. Понятие хеширования. Принцип размещения элементов множества в памяти. Преимущества и ограничения множеств, связанные с особенностью размещения в памяти. Задание множеств. Работа с множествами. Методы множеств. Операции над множествами.

**Тема 16. Словари****Цели и задачи:**

Получить представление об устройстве и принципе работы словарей. Изучить методы словарей.

**Результаты освоения:**

Уметь использовать словари для решения задач, требующих быстрого поиска по произвольному значению, в том числе, задачи подсчета определенных значений.

**Краткое содержание занятий:**

Задача о подсчете частоты употребления слов в файле. Устройство и принцип работы словарей. Сравнение словарей с множествами. Задание слова-

рей. Методы словарей. Работа со словарями. Применение словарей для решения задачи о подсчете частоты употребления слов файле.

## 5 Оценочные и методические материалы

### Формы аттестации

Для оценки результативности обучения по дополнительной общеобразовательной программе «Программирование на языке Python» применяется входной, текущий, промежуточный контроль.

#### Входной контроль

Предназначен для определения исходного уровня знаний и умений, проводится в форме собеседования.

#### Текущий контроль

Осуществляется автоматизировано с помощью автоматизированной системы на базе системы управления курсами Moodle, размещенной по адресу [test.kod-u.ru](http://test.kod-u.ru). Задачи для решения находятся в разделе «Курсы / Программирование / ru.rgr»». Наборы задач делятся на обязательные и дополнительные и имеют различный уровень сложности, начиная с 1 (простые задачи) и далее по возрастанню.

Раздел считается освоенным, если задачи из обязательных наборов решены в объеме не менее 80% от числа задач, входящих в набор. Если задачи обязательных наборов решались с существенной помощью педагога, обучаемому предлагается прорешать задачи дополнительных наборов.

Фиксация результатов осуществляется автоматически в разделе разделе «Курсы / Программирование / ru.rgr / Управление курсом / Отчеты / Завершение элементов курса».

#### Промежуточный контроль

Проводится раз в три месяца в форме олимпиады по программированию. Порядок проведения описан в документе «Положение о закрытой олимпиаде по программированию ЦРШ "Код Успеха"».

### Методическое обеспечение

#### Дидактический материал

№ п/п	Тема	Формы занятий	Дидактический материал
1.	Введение	Лекция	Введение. Слайды <sup>1</sup>
2.	Основные понятия	Лекция	Основные понятия. Слайды <sup>1</sup>
3.	Вычисления	Практика	

<sup>1</sup> Сайт [test.kod-u.ru](http://test.kod-u.ru). Раздел «Курсы / Программирование / ru.rgr»



4.	Условная инструкция	Лекция, Практика	Ветвления. Слайды <sup>1</sup>
5.	Цикл while	Лекция, Практика	Цикл while. Слайды <sup>1</sup>
6.	Цикл for	Лекция, Практика	Цикл for (слайды) <sup>1</sup> Функция print (слайды) <sup>1</sup>
7.	Списки	Лекция, Практика	Списки. Слайды <sup>1</sup>
8.	Линейные алгоритмы	Лекция, Практика	
9.	Строки	Лекция, Практика	Строки (слайды) <sup>1</sup>
10.	Матрицы	Лекция, Практика	Матрицы. Слайды <sup>1</sup>
11.	Методы строк и операции над строками	Лекция, Практика	Методы строк. Слайды <sup>1</sup>
12.	Функции	Лекция, Практика	Слайды. Функции <sup>2</sup>
13.	Встроенная сортировка	Лекция, Практика	
14.	Файлы	Лекция, Практика	Файлы. Слайды <sup>2</sup>
15.	Множества	Лекция, Практика	Множества. Слайды <sup>2</sup>
16.	Словари	Лекция, Практика	Словари. Слайды <sup>2</sup>

### Присмы и методы организации образовательного процесса

При проведении лекционных занятий рекомендуется делать их максимально интерактивными, используя активизирующие вопросы, ставя перед обучающимися проблемы и обсуждая возможные пути их решения, анализируя их достоинства и недостатки. В ходе изложения материала избегать объяснения алгоритма приведением конкретного кода. Вместо этого целесообразно использовать блок-схемы алгоритма. Объяснение алгоритма необходимо сопровождать примерами его работы на конкретных исходных данных.

В первых двух лекциях необходимо сформировать основной понятийный аппарат: переменная, тип, выражение, функция.

При проведении практических занятий добиваться максимально самостоятельного решения задач. Обращать внимание на осознание необходимости

<sup>1</sup> Сайт [informatics.msk.ru](http://informatics.msk.ru). Раздел «Кружки и уроки / Ульяновск / Код Успеха / Ульяновск. Код Успеха Python»

самостоятельной работы. Сложность решаемых задач должна быть в зоне ближайшего развития.

При возникновении длительных затруднений давать небольшие подсказки в форме наводящих вопросов, графических схем, иллюстрирующих суть задачи. Начать помощь стоит с того, что нужно убедиться, что обучаемый правильно понял условие задачи. Для этого можно попросить пересказать условие и объяснить, как получается ответ на одном из примеров исходных данных.

Следующим шагом может быть предложить понаблюдать обучаемому над своими рассуждениями при решении задачи и словами описать алгоритм. Далее можно предложить ему составить блок-схему алгоритма. Если при этом допущены ошибки, попросить выполнить действия алгоритма, записывая при этом значения переменных на доске. В качестве исходных данных предложить такие, на которых предлагаемый учеником алгоритм дает неверный ответ. Далее совместно ищется способ исправить ошибку в блок-схеме. После этого ее предлагается закодировать.

В зависимости от задачи, иногда имеет смысл сразу писать код программы. В этом случае так же можно попросить подробно рассказать, как она будет работать, записывая на доске значения переменных. Так же ищется и исправляется ошибка. Составленный общими усилиями код на доске целесообразно стереть, с тем, чтобы обучаемый воспроизвел его повторно самостоятельно. Исключение могут составлять очень слабые ученики.

Если код обучаемым уже написан и в нем «глупая ошибка», часто хорошо работает метод «duck debug». Обучаемому предлагается объяснить, как работает его программа. В ходе объяснения он находит ошибку.

При возникновении ситуации, когда часть тестов проходит, а часть получает вердикт «Неправильный ответ» необходимо ознакомить, а при повторных ситуациях напоминать правила составления тестов (проверка крайних значений исходных данных и пограничных значений в условиях и циклах; тесты должны заставить вычислительный процесс пройти по всем возможным маршрутам). Если тест неочевидный и плохо подбирается по изложенным выше правилам, допускается подсказать, на каких исходных данных программа выдает неправильный ответ.

После нахождения теста, дающего ошибку, необходимо научить вдумчиво и системно ее исправлять. Объяснить необходимость избегать применения «костылей» для исправления программы.

## 6 Материально-техническое обеспечение программы

### Перечень необходимого оборудования

#### Техническое оснащение:

- Кабинет, имеющий хорошее освещение.
- Компьютеры (ноутбуки) из расчета один на обучаемого, с доступом к сети Интернет.

- Стулья, рабочие столы.
- Стол и стул педагога.
- Доска магнитно-маркерная.
- Проектор

#### Программные средства:

- Операционная система Windows 7 (или выше)
- Интерпретатор Python 3.5 (или выше)
- IDE: IDL, Wing 101, PyCharm

#### Учебно-методическое обеспечение:

1. Слайды по изучаемым темам.
2. Сайты <http://test.kod-u.ru/> и <http://informatics.mccme.ru> с подготовленными разделами для реализации данной программы.

## 7 Кадровое обеспечение

Согласно Профессиональному стандарту «Педагог дополнительного образования детей и взрослых» по данной программе может работать педагог дополнительного образования с уровнем образования и квалификации, соответствующим обозначениям таблицы пункта 2 Профессионального стандарта (Описание трудовых функций, входящих в профессиональный стандарт), а именно: коды А и В с уровнями квалификации 6.

## 8 Список литературы

### Литература для педагога

1. Федеральный закон «Об образовании в Российской Федерации» №273-ФЗ от 29.12.2012 (ст.2, п.9, п. 14; ст.12, п.5; ст.33, п.2; ст.75, п.2, п.4)
2. Приказ Министерства просвещения РФ от 9 ноября 2018 г. N 196 "Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам"
3. «Санитарно-эпидемиологические требования к устройству, содержанию и организации режима работы образовательных организаций дополнительного образования детей» от 04.07.2014 № 41
4. Концепция развития дополнительного образования детей от 04.09.2014 № 1726-р (раздел IV)
5. Методические рекомендации по проектированию дополнительных общеразвивающих программ (Приложение к письму Департамента государственной политики в сфере воспитания детей и молодежи Министерства образования и науки РФ от 18.11.2015 № 09-3242)
6. Python для детей. Самоучитель по программированию / Джейсон Бриггс; пер. с англ. Станислава Ломакина; [науч. ред. Д. Абрамова]. – 2-е изд. – М.: Манн, Иванов и Фербер, 2018. -320 с.
7. Программируем с Minecraft. Создай свой мир с помощью Python / Крэйг Ричардсон; пер. с англ. Станислава Ломакина; [науч. ред. Г. Гаджиев]. –М.

- Мани, Ивонна-Мария, 2007. - 288 с. ил.
8. Олимпиадные задачи по информатике / Антти Лааксонен; пер. с англ. А.А. Слинкин - М.: Манн, Иванов и Фербер, 2018. - 288 с. ил.

### Литература для учителя

9. Python для детей. Самоучитель по программированию / Джейсон Бриггс; пер. с англ. Станислав Лавринович [науч. ред. Д. Абрамова]. - 2-е изд. - М.: Манн, Иванов и Фербер, 2018. - 288 с.

### Интернет-ресурсы

1. <https://python.org>
2. <http://python.org>
3. <https://www.dbooks.org/tutorial/python>