

**МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Комитет образования, науки и молодежной политики**

**Волгоградской области**

**Департамент по образованию администрации Волгограда**

**МОУ СШ №81**

**РАССМОТРЕНО**

Методическим  
объединением МОУ  
СШ №81

\_\_\_\_\_  
Степаненкова Н.П.  
Протокол № 1 от «30» 08.  
2024 г.

**СОГЛАСОВАНО**

Заместитель директора по  
УВР МОУ СШ №81

\_\_\_\_\_  
Чекомасова И.В.  
Протокол № 1 от «30» 08. 2024 г.

**УТВЕРЖДЕНО**

Директор  
МОУ СШ №81

\_\_\_\_\_  
Пономарева Е.А.  
Приказ № 204 от «30» 08.  
2024 г.

**РАБОЧАЯ ПРОГРАММА**

учебного курса

**«Алгоритмизация и программирование»**

11 класс

Составитель: Попова Татьяна Анатольевна,  
учитель информатики

## **Пояснительная записка**

Методологической основой федеральных государственных образовательных стандартов является системно-деятельностный подход, в рамках которого реализуются современные стратегии обучения, предполагающие использование информационных и коммуникационных технологий (ИКТ) в процессе изучения всех предметов, во внеурочной и внешкольной деятельности на протяжении всего периода обучения в школе. Организация учебно-воспитательного процесса в современной информационно-образовательной среде является необходимым условием формирования информационной культуры современного школьника, достижения им ряда образовательных результатов, прямо связанных с необходимостью использования информационных и коммуникационных технологий.

Средства ИКТ не только обеспечивают образование с использованием той же технологии, которую учащиеся применяют для связи и развлечений вне школы (что важно само по себе с точки зрения социализации учащихся в современном информационном обществе), но и создают условия для индивидуализации учебного процесса, повышения его эффективности и результативности. На протяжении всего периода существования школьного курса информатики преподавание этого предмета было тесно связано с информатизацией школьного образования: именно в рамках курса информатики школьники знакомились с теоретическими основами информационных технологий, овладевали практическими навыками использования средств ИКТ, которые потенциально могли применять при изучении других школьных предметов и в повседневной жизни.

Курс «Алгоритмизация и программирование» развивает алгоритмическое, операциональное мышление обучающегося. Умение разбить задачу на подзадачи, умение воспользоваться готовым алгоритмом более простой задачи при решении сложной – это общеучебные умения и навыки, которые формируются у каждого выпускника. Изучение программирования – как прагматическая цель заключается в освоении основ профессионального программирования. Такую цель можно ставить перед профильным или элективным предметом по информатике.

Рабочая программа дает последовательность изучения тем и разделов с учетом межпредметных и внутрипредметных связей, логики учебного процесса, возрастных особенностей обучающихся.

Данная рабочая программа ориентирована на учащихся 10-11 классов.

Рабочая программа выполняет две основные функции.

Информационно-методическая функция позволяет всем участникам образовательного процесса получить представление о целях, содержании, общей стратегии обучения, воспитания и развития обучающихся средствами данного учебного предмета.

Организационно-планирующая функция предусматривает выделение этапов обучения, структурирование учебного материала, определение его количественных и качественных характеристик на каждом из этапов, в том числе для содержательного наполнения промежуточной аттестации обучающихся.

Элективный предмет на старшей ступени школы становится неотъемлемой частью обновления среднего образования, средством улучшения его качества. Задачей элективного предмета является создание необходимой базы для понимания вузовских курсов и развитие навыков самостоятельной учебной деятельности.

### **Структура документа**

Рабочая программа включает разделы: пояснительную записку; основное содержание с распределением учебных часов по разделам курса, последовательность изучения тем и разделов; учебно-тематический план; требования к уровню подготовки обучающихся; литература и средства обучения; календарно-тематическое планирование.

### **Цели**

**Изучение учебного курса направлено на достижение следующих целей:**

- овладеть базовыми понятиями теории алгоритмов и как следствие - развитие алгоритмического мышления, способностей к формализации, элементов системного мышления;
- приобретение навыков разработки эффективные алгоритмов и реализация их в виде программы, написанных на языке программирования Питон;
- формирование представления о профессии программиста;
- освоение и систематизация знаний, относящихся к математическим объектам информатики;

- формирование общеучебных умений и способов интеллектуальной деятельности для решения познавательных задач и саморазвития на основе методов информатики;
- воспитание ответственного отношения к соблюдению этических и правовых норм информационной деятельности.

### **Задачи**

- расширить представление о понятиях алгоритм, вычислимая функция, язык программирования;
- научить составлять и читать блок-схемы;
- сформировать навыки выполнения технологической цепочки разработки программ средствами языка программирования Питон;
- изучить основные конструкции языка программирования Питон, позволяющие работать с простыми скалярными и составными (массивами, файлами, множествами, строками) типами данных;
- научить работать с графическими средствами языка программирования Питон;
- научить применять подпрограммы при написании программ на языке программирования Питон;
- научить отлаживать и тестировать программы, делать выводы о работе этих программ.

### **Основные формы и виды организации учебного процесса**

Основной формой организации учебного процесса является классно-урочная система. В качестве дополнительных форм организации образовательного процесса в школе используется система консультационной поддержки, индивидуальных занятий, лекционные, семинарские занятия, самостоятельная работа учащихся с использованием современных информационных технологий. Организация сопровождения учащихся направлена на:

1. создание оптимальных условий обучения;
2. исключение психотравмирующих факторов;
3. сохранение психосоматического состояния здоровья учащихся;
4. развитие положительной мотивации к освоению гимназической программы;
5. развитие индивидуальности и одаренности каждого подростка.

### **Типы урока**

1. Урок усвоения новых знаний или нового материала.
2. Повторительно-обобщающий урок.
3. Урок закрепления изученного материала.
4. Урок контроля.
5. Урок обобщающего контроля

### **Формы организации работы учащихся:**

1. Индивидуальная.
2. Коллективная:
  - 2.1. Фронтальная;
  - 2.2. Парная;
  - 2.3. Групповая.

### **Особенности организации учебного процесса. Используемые технологии**

Организация учебно-воспитательного процесса должна соответствовать принципам развивающего обучения (нарастание самостоятельности, поисковой деятельности обучающихся. Выполнение заданий, ведущих от воспроизводящей деятельности к творческой, а также, личностно-ориентированному и дифференцированному подходам).

В учебно-воспитательном процессе используются современные образовательные технологии (ИКТ, проблемное обучение, учебное исследование, проблемно-поисковые технологии, тестовая технология, технология опорных сигналов, технология адаптивной системы обучения, технология работы в парах).

Применяется интеграция традиционной, алгоритмической, модульной, игровой, компьютерной технологий и развивающего обучения.

### **Формы учебных занятий**

1. Мини – лекции;
2. Диалоги и беседы;
3. Практические работы;
4. Проектные работы

### **Виды деятельности учащихся**

1. Устные сообщения;
2. Обсуждения;
3. Работа с источниками;
4. Рефлексия

### **Общая характеристика учебного курса**

Профильная дифференциация обучения на старшей ступени школы становится неотъемлемой частью обновления среднего образования, средством улучшения его качества.

Задачей элективных предметов является создание необходимой базы для понимания вузовских курсов и развитие навыков самостоятельной учебной деятельности.

Обучение информатике отвечает потребностям различных направлений специализации в старших классах и именно поэтому уже достаточно широко используется в школьной практике. Но наряду с очевидным положительным опытом появляются и отрицательные тенденции в формировании содержания обучения.

Последние годы характеризовались уменьшением количества часов на изучение алгоритмизации и программирования в старшей школе, что было объективно связано с бурным развитием информационных технологий. Чрезмерное увлечение «пользовательской компонентой» вытеснило изучение этих вопросов не только из программы предмета «Информатика и ИКТ», но даже из ряда учебников. При явном улучшении оснащения школ компьютерной техникой уровень общеобразовательной подготовки в данной области выпускников заметно снизился. Исключение из учебной программы вопросов, связанных с алгоритмизацией и программированием из-за плохого восприятия учащимися крайне необоснованно. Данный учебный курс направлен на решение сложившейся проблемы.

#### **Место учебного курса в учебном плане**

В соответствии с учебным планом МОУ СШ № 81 на изучение учебного курса «Алгоритмизация и программирование» в 10-11 классах отводится 1 час в неделю, 34 часа годовых.

#### **Общеучебные умения, навыки и способы деятельности**

Программа предусматривает формирование у учащихся общеучебных умений и навыков, универсальных способов деятельности и ключевых компетенции. В этом направлении приоритетами для курса «Алгоритмизация и программирование» на этапе среднего (полного) общего образования являются: определение адекватных способов решения учебной задачи на основе заданных алгоритмов; комбинирование известных алгоритмов деятельности в ситуациях, не предполагающих стандартное применение одного из них; использование для решения познавательных и коммуникативных задач различных источников информации, включая энциклопедии, словари, Интернет-ресурсы и базы данных; владение умениями совместной деятельности (согласование и координация деятельности с другими ее участниками; объективное оценивание своего вклада в решение общих задач коллектива; учет особенностей различного ролевого поведения).

При реализации данной программы планируется использовать следующие приемы и методы обучения:

- Словесные методы обучения (рассказ, объяснение, лекция, беседа, работа с учебником на печатной основе или электронным);
- Наглядные методы (наблюдение, иллюстрация, демонстрация наглядных пособий, презентаций);
- Практические методы (устные и письменные упражнения, практические компьютерные работы);
- Активные методы (метод проблемных ситуаций, метод проектов).

В рамках реализации данной программы предусматривается использование следующих образовательных технологий: проблемное обучение, тестовая технология, технология опорных сигналов, технология работы в парах.

Важным компонентом учебного процесса является контроль, или проверка результатов обучения. Входной оперативный контроль осуществляется в начале каждого урока. Это типовой для каждого комбинированного урока этап актуализации и проверки усвоения изученного материала. Одной из наиболее актуальных форм организации проверочного, тематического и итогового контроля является тестирование. Используются также и такие формы тематического контроля, как практическая контрольная работа. Также используются такие формы контроля как собеседование, экспресс-опрос, зачет по опросному листу и др. Кроме того, каждый учащийся в течение учебного года, в процессе выполнения компьютерного практикума пополняет содержимое своего портфолио – личной файловой папки, где содержится вся коллекция работ, демонстрирующая его усилия, прогресс или достижения в определенной области.

#### **Основное содержание**

##### **Раздел 1. Алгоритмизация**

Алгоритмы. Виды, свойства, типы алгоритмов. Этапы создания алгоритма. Представление и запись алгоритмов. Исполнители алгоритма. Язык блок-схем. Линейный алгоритм. Разветвляющийся алгоритм. Циклический алгоритм. Вспомогательный алгоритм.

*Учащиеся должны знать / понимать:*

- понятие алгоритма;
- понятие исполнителя;
- свойства алгоритма;
- язык блок-схем;
- виды алгоритмов.

*Учащиеся должны уметь:*

- записать алгоритм разными способами;
- определить исполнителя алгоритма.

## **Контрольная работа № 1.**

### **Раздел 2. Основы языка**

#### **Тема 2.1. Структура программы на языке Python. Типы данных и операторы**

История и классификация языков программирования. Программы. Язык программирования Python и его характерные особенности. Структура программы на языке Python. Простейшая программа. Тело программы. Среда программирования PythonABC. Элементы языка Python. Создание и исполнение программ в среде программирования PythonABC. Операторы ввода-вывода. Использование памяти. Переменные. Типы данных в языке Python. Простые типы данных. Целые и вещественные типы. Значения. Операторы присваивания. Операции, допустимые с переменными и значениями целого и вещественного типа.

*Учащиеся должны знать/понимать:*

- понятие программы;
- общую структуру программы;
- типы данных;
- целые, вещественные типы данных и операции над ними;
- оператор присваивания;
- операторы ввода-вывода.

*Учащиеся должны уметь:*

- пользоваться интерфейсом среды программирования Python ABC;
- использовать команды редактора;
- организовывать ввод и вывод данных;
- записывать арифметические выражения.

**Практическая работа:** работа с системой ABC Python. Составление программ. Задачи и упражнения разделов «Алгоритмы», «Способы описания алгоритмов», «Простые типы данных. Описание переменных», «Числовые типы данных. Выражения. Оператор присваивания. Процедуры ввода, вывода».

#### **Тема 2.2. Логический тип данных. Условный оператор. Составной оператор**

Логический тип данных. Логические выражения. Сложные условные выражения (логические операции **and**, **or**, **not**). Условный оператор. Составной оператор.

Этапы решения задачи на ЭВМ.

*Учащиеся должны знать / понимать:*

назначение условного оператора;  
способ записи условного оператора;  
логический тип данных;  
логические операторы **OR**, **AND**, **NOT**;  
операторные скобки **BEGIN...END**.

*Учащиеся должны уметь:*

использовать условный оператор;  
создавать сложные условия с помощью логических операторов.

**Практическая работа:** решение задач по теме «Условный оператор». Упражнения и задачи разделов «Логический тип данных. Конструкции ветвления».

#### **Тема 2.3. Оператор выбора. Символьный тип данных**

Оператор выбора. Символьный тип данных. Организация таблицы ASCII-кодов. Операции над данными символьного типа.

*Учащиеся должны знать / понимать:*

назначение оператора выбора варианта;

организацию таблицы ASCII-кодов;  
операции, допустимые над данными символьного типа.

*Учащиеся должны уметь:*

правильно применять оператор выбора варианта;  
описывать данные символьного типа;  
определять код символа и символ по заданному коду;  
использовать основные функции для символьного типа данных.

**Практическая работа:** решение задач по теме «Оператор варианта». Упражнения и задачи раздела «Символьный тип данных. Оператор варианта».

## **Контрольная работа №2.**

### **Раздел 3. Циклы и графика**

#### ***Тема 3.1. Операторы циклов с условием***

Понятие цикла. Тело цикла. Условия выполнения тела цикла. Циклы с условием и их виды. Оператор цикла с предусловием, блок-схема оператора. Оператор цикла с постусловием. Отличия циклов с предусловием от циклов с постусловием. Вложенные циклы. Бесконечные циклы.

*Учащиеся должны знать / понимать:*

циклы с условием и их виды;  
различие между циклами с предусловием и постусловием;  
правила записи циклов с предусловием и постусловием;  
примеры использования циклов различных типов.

*Учащиеся должны уметь:*

определять вид цикла, наиболее удобный для решения поставленной задачи;  
использовать цикл с предусловием;  
использовать цикл с постусловием.

**Практическая работа:** решение задач по теме «Оператор цикла с предусловием и с постусловием». Упражнения и задачи раздела «Операторы циклов».

#### ***Тема 3.2. Оператор цикла с параметром***

Оператор цикла с параметром. Правила записи параметра цикла. Вложенные циклы.

*Учащиеся должны знать / понимать:*

назначение и особенности использования цикла с параметром;  
формат записи цикла с параметром;  
тип данных параметра цикла.

*Учащиеся должны уметь:*

определять целесообразность применения цикла с параметром для решения поставленной задачи;  
использовать цикл с параметром.

**Практическая работа:** решение задач по теме «Оператор цикла с параметром». Упражнения и задачи раздела «Операторы циклов».

#### ***Тема 3.3. Построение графических изображений средствами языка Python***

Подключение модулей. Работа с модулем GraphABC. Графический режим работы монитора. Переключение монитора в графический режим. Координаты. Подпрограммы вывода графических изображений. Пиксель. Вывод пикселя. Создание простейших графических изображений. Создание анимации.

*Учащиеся должны знать / понимать:*

основные принципы построения графических изображений;  
способ инициализации графического режима работы;  
принцип анимации;  
процедуры и функции модуля GraphABC.

*Учащиеся должны уметь:*

инициализировать графический режим работы;  
использовать для построения графических изображений процедуры и функции модуля GraphABC.

**Практическая работа:** построение графических изображений средствами языка Python.  
Упражнения и задачи раздела «Графика».

### **Контрольная работа № 3.**

#### **Раздел 4. Сложные типы данных**

##### **Тема 4.1. Одномерные массивы**

Сложные типы данных. Массивы. Одномерные массивы. Ячейки массива. Индексы. Описание одномерных массивов. Ввод, вывод массивов. Работа с массивами: поиск элемента в массиве, поиск минимума и максимума, нахождение суммы элементов массива и т. д.

*Учащиеся должны знать / понимать:*

сложные типы данных;

назначение массивов;

способ описания одномерного массива;

операции, выполняемые с массивами;

способ доступа к элементам массива;

алгоритмы поиска минимального и максимального элементов массива.

*Учащиеся должны уметь:*

описывать массивы;

вводить элементы массива;

выводить элементы массива;

находить минимальный и максимальный элементы массива.

**Практическая работа:** решение задач по теме «Некоторые составные типы». Упражнения и задачи раздела «Регулярные типы (массивы)».

##### **Тема 4.2. Многомерные массивы (матрицы)**

Многомерные массивы. Описание многомерных массивов. Ввод, вывод массивов. Работа с массивами: поиск элемента в массиве, поиск минимума и максимума, нахождение суммы элементов массива и т. д.

*Учащиеся должны знать / понимать:*

назначение многомерных массивов;

способ описания многомерных массивов;

способ доступа к элементам многомерного массива.

*Учащиеся должны уметь:*

использовать многомерные массивы;

приводить примеры использования многомерных массивов.

**Практическая работа:** решение задач по теме «Некоторые составные типы». Упражнения и задачи раздела «Регулярные типы (массивы)».

##### **Тема 4.3. Строковый тип данных**

Строки. Строковый тип данных. Описание строк. Ввод, вывод строковых переменных.

Операции со строками: конкатенация, сравнение. Функции и процедуры для работы со строками: определение длины строки, удаление подстроки, вставка подстроки.

*Учащиеся должны знать / понимать:*

назначение строкового типа данных;

способы описания строк;

операторы для работы со строками;

процедуры и функции для работы со строками;

операции со строками.

*Учащиеся должны уметь:*

описывать строки;

соединять строки;

находить длину строки;

вырезать часть строки;

находить подстроку в строке;

находить количество слов в строке.

**Практическая работа:** решение задач по теме «Строковый тип». Упражнения и задачи раздела «Строковый тип данных».

#### **Контрольная работа № 4.**

##### **Тема 4.4. Множественный тип данных. Рекурсия.**

Множества. Множественный тип данных. Описание множеств. Операции, допустимые над множествами: объединение, пересечение, разность, включение. Оператор определения принадлежности элемента множеству. Рекурсивная процедура.

*Учащиеся должны знать / понимать:*

понятие множества;  
способы описания множества;  
операторы работы с множествами;  
понятие рекурсивной процедуры.

*Учащиеся должны уметь:*

описывать множества;  
задавать константы-множества;  
определять принадлежность элемента множеству;  
вводить элементы множества;  
выводить элементы множества.

**Практическая работа:** решение задач по теме «Множественный тип». Упражнения и задачи раздела «Комбинированный тип данных (запись)».

#### **Контрольная работа № 5.**

##### **Раздел 5. Шаги к профессионализму**

##### **Тема 5.1. Подпрограммы**

Подпрограммы. Цели и преимущества использования подпрограмм. Виды подпрограмм. Процедуры. Описание процедур. Функции. Описание функций. Параметры подпрограмм. Формальные и фактические параметры. Область действия переменных. Параметры-переменные, параметры-значения. Передача параметров по значению. Передача параметров по ссылке.

*Учащиеся должны знать / понимать:*

понятие подпрограммы;  
способы описания процедуры;  
способы описания функции;  
отличия функций от процедур;  
принципы структурного программирования;  
понятие локальных переменных подпрограмм;  
понятие формальных и фактических параметров подпрограмм;  
способ передачи параметров.

*Учащиеся должны уметь:*

создавать и использовать процедуры;  
создавать и использовать функции;  
использовать механизм параметров для передачи значений в подпрограмму.

**Практическая работа:** решение задач по теме «Подпрограммы». Упражнения и задачи раздела «Подпрограммы».

##### **Тема 5.2. Работа с файлами**

Файловый тип данных. Файловые переменные. Типизированные файлы. Текстовые файлы. Основные операции с файлами.

*Учащиеся должны знать / понимать:*

- понятие файловой переменной;
- способы описания файловых переменных;
- отличия типизированных файлов от текстовых;
- последовательность работы с файлом на диске.

*Учащиеся должны уметь:*

- инициализировать файловую переменную;
- открывать файл для чтения или записи;
- производить чтение или запись данных в файл.

**Практическая работа:** решение задач по теме «Файловые типы». Упражнения и задачи раздела «Файловые типы».

**Тема 5.3. Комбинированный тип данных. Оператор присоединения**

Комбинированный тип данных. Записи. Поля записей. Описание записей. Способы обращения к полям записи. Оператор присоединения.

*Учащиеся должны знать / понимать:*

понятие записи;

назначение записи;

способ описания записи;

обращение к полю записи;

оператор присоединения.

*Учащиеся должны уметь:*

разрабатывать программы, использующие записи;

использовать оператор присоединения.

**Практическая работа:** решение задач по теме «Комбинированный тип (запись)». Упражнения и задачи раздела «Комбинированный тип данных (запись)».

**Тема 5.4. Динамические массивы. Динамическое программирование.**

Динамические структуры данных. Списки. Стек, очередь, дек. Деревья. Графы. Динамическое программирование.

*Учащиеся должны знать / понимать:*

понятие динамического массива;

примеры динамических массивов;

назначение и применение динамических структур данных.

*Учащиеся должны уметь:*

решать сложные задачи путем сведения к более простым подзадачам.

**Индивидуальный проект**

# 1. УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН

Наименование разделов и тем	Количество часов
	Всего
<b>Основы алгоритмизации</b>	<b>19</b>
<b>Основы программирования</b>	<b>15</b>
<b>ВСЕГО</b>	<b>34</b>

## Поурочное планирование Всего часов -34

№ п/п	Раздел	Тема урока
1	Основы алгоритмизации	Понятие алгоритма. Характеристики алгоритма. Исполнитель алгоритма. Свойства алгоритма.
2		Формы записи алгоритма. Словесный способ записи алгоритмов. Графический способ записи алгоритмов.
3		Отличия программного способа записи алгоритмов от других. Какие у машинных языков достоинства и недостатки.
4		Практическая работа № 1. «Составление алгоритмов решения задач»
5		Понятия алгоритмического языка и языков программирования. Имена. Операции. Данные: константы, переменные, массивы. Выражения: арифметические, логические и строковые. Операторы: неисполняемые, исполняемые. Основные символы языка Паскаль.
6		Величина. Исходные (входные), выходные, промежуточные данные. Атрибуты величины: имя (идентификатор), тип, значение. Постоянная, переменная величина. Описание переменной. Типы переменных. Целые типы. Вещественные типы. Логический тип. Символьный тип. Ввод-вывод информации. Оператор присваивания. Операции целочисленного деления и нахождения остатка от деления.
7		Практическая работа № 2. Решение задач.
8		Встроенные математические функции языка Паскаль. Правила записи математических выражений.
9		Обобщение по темам: «Встроенные математические функции языка Паскаль. Правила записи математических выражений».
10		Базовые алгоритмические структуры: следование, ветвление: полная и неполная форма ветвления, форма ветвления «выбор», «иначе».
11		Самостоятельная работа. Разбор и решение задач с использованием алгоритмической структуры «ветвление».
12		Практическая работа № 3. «Решение задач с использованием ветвления».
13		Контрольная работа.
14		Базовые алгоритмические структуры: цикл
15		Практическая работа №4 по теме «Циклы»

№ п/п	Раздел	Тема урока
16		Итерационный цикл (цикл пока). Определение итерационного цикла. Запись цикла с использованием алгоритмической структуры «цикл пока». Запись цикла на языке программирования Паскаль. Разбор решения задач
17		Стандартные функции. Запись математических выражений на языке Паскаль
18		Запись логических выражений. Использование операций отношений. Разбор задач на вычисление значения логических выражений.
19		Самостоятельная работа по теме «Логические выражения».
20	Основы программирования	Среда Паскаль
21		Структура программы на языке Паскаль
22		Этапы решения задач с помощью компьютера.
23		Отладка и тестирование программ.
24		Одномерный массив. Задание массива.
25		Самостоятельная работа по теме: «Одномерный массив».
26		Строковый тип данных в языке Паскаль.
27		Самостоятельная работа по теме: «Символьные переменные. Операции со строками».
28		Структура двумерного массива и его описание.
29		Самостоятельная работа по теме: «Структура двумерного массива и его описание».
30		Функции. Процедуры. Фактические и формальные переменные.
31		Локальные и глобальные идентификаторы.
32		Графика Паскаль.
33		Использование процедур в графике
34		Урок-обобщение по курсу

## ТРЕБОВАНИЯ К УРОВНЮ ПОДГОТОВКИ ОБУЧАЮЩИХСЯ

В процесс обучения заложены следующие необходимые умения и навыки:

- Знать основные алгоритмические конструкции: следование, ветвление, цикл при решении задач;
- Знать основные принципы построения алгоритма для решения задач;
- Знать этапы решения задач на компьютере;
- Знать назначение языков программирования;
- Уметь составлять алгоритмы решения задач;
- Уметь составлять программы и реализовывать их на компьютере с использованием языка программирования Python.

## **ЛИТЕРАТУРА И СРЕДСТВА ОБУЧЕНИЯ**

Преподавание данного учебного курса ориентировано на использование **учебного и программно-методического комплекса**, в который входят:

- Л.З. Шауцукова «Информатика. Алгоритмизация и программирование»,
- Э.В.Лебедева «Практикум по решению задач в курсе информатики»,

С.В. Мациевский, С.А. Ишанов, С.В. Клевцур «Информатика. Учебное пособие».

### **Оборудование и приборы**

#### **Аппаратные средства**

- Компьютер
- Проектор
- Принтер
- Модем
- Устройства вывода звуковой информации — наушники для индивидуальной работы со звуковой информацией
- Устройства для ручного ввода текстовой информации и манипулирования экранными объектами — клавиатура и мышь.
- Локальная и глобальная сети.

#### **Программное обеспечение курса:**

- Среда программирования Паскаль ABC;
- Браузер Internet Explorer, Mozilla Firefox;
- Система онлайн-словарей и переводчиков;
- Менеджер загрузки файлов FlashGet.