

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ УПРАВЛЕНИЯ»  
(ГУУ)

Предуниверсарий ГУУ

Одобрено на  
заседании  
Президиума  
Учебно-  
методического сове  
та ГУУ  
от 08.06.2020 г.  
протокол № 7

Принято Ученым  
советом  
университета  
Протокол № 13 от  
«08» 07 2020 г.

Согласовано  
директор  
Предуниверсария  
ГУУ  
М.Ю. Григорьева  
«08» 07 2020 г.

Утверждаю  
проректор ГУУ

А.В. Троицкий

«08» 07 2020 г.



РАБОЧАЯ ПРОГРАММА

Наименование учебного предмета  
Класс(ы)  
Уровень образования  
Образовательная программа  
Срок реализации программы

Программирование  
10-11  
Среднее общее образование  
Основная образовательная программа  
(элективный курс)  
2 года

**Рабочая программа элективного курса «Программирование»** для учащихся 10 - 11 классов информационно-технологического профиля ориентирована на использование учебного пособия для общеобразовательных организаций К.Ю. Полякова «Программирование. Python. С++» в четырех частях. Включает 68 часов аудиторных занятий и (при возможности) самостоятельную работу учащихся, из них 35 часов в 10 классе и 33 часа в 11 классе

Курс может быть использован для профильной подготовки учащихся в классах физико-математического, информационно-технологического профилей и рассчитан на 1 час в неделю сроком реализации 2 года.

Электронный материал авторских разработок доктора технических наук К.Ю. Полякова для сопровождения курса представлен в полном объеме на сайте автора <http://kpolyakov.spb.ru/school/c.htm> и включает презентации, решение задач и сборник программ в среде Dev-C++.

Одно из достоинств современного языка С++ — обширная стандартная библиотека: набор готовых подпрограмм для решения часто встречающихся задач (например, для обработки символьных строк).

Целесообразность изучения данного курса определяется тем, что язык С++ — основной язык для создания игр и операционных систем. Программы на С++ работают значительно быстрее, чем на других языках.

### **Планируемые результаты освоения обучающимися программы элективного курса «Программирование»**

В рамках данного курса учащиеся получают следующие знания и умения:

Уметь:

- самостоятельно определять цели деятельности и составлять планы деятельности; самостоятельно осуществлять, контролировать и корректировать деятельность; использовать все возможные ресурсы для достижения поставленных целей и реализации планов деятельности; выбирать успешные стратегии в различных ситуациях;
- продуктивно общаться и взаимодействовать в процессе совместной деятельности, учитывать позиции других участников деятельности, эффективно разрешать конфликты;
- владеть навыками познавательной, учебно-исследовательской и проектной деятельности, навыками разрешения проблем; осуществлять самостоятельный поиск методов решения практических задач;
- вести самостоятельную информационно-познавательную деятельность, включая ориентирование в различных источниках информации, критически оценивать и интерпретировать информацию, получаемую из различных источников;

- применять навыки алгоритмического мышления и понимать необходимость применения формального описания алгоритмов;
- использовать понятие сложности алгоритма, применять основные алгоритмы обработки числовой и текстовой информации, алгоритмы поиска и сортировки;
- владеть приёмами написания программы в среде Dev-C++ для решения прикладных задач с использованием основных конструкций программирования; осуществлять отладку таких программ; тестировать и оптимизировать готовые прикладные компьютерные программы;
- применять базовые типы данных и структуры данных; использовать основные управляющие конструкции;
- осуществлять формализацию задач и производить документирование программ.

**Знать:**

- алгоритмические конструкции, лежащие в основе программирования в среде Dev-C++ с открытым исходным кодом, включающей компилятор GCC;
- синтаксис операторов и их применение в решении задач;
- основные понятия объектно-ориентированного программирования (объект, свойство, метод, наследование).

## **Формы подведения итогов**

Текущий контроль уровня усвоения материала осуществляется по результатам выполнения учащимися практических заданий. В конце разделов курса каждый учащийся выполняет индивидуальный проект в качестве зачетной работы.

## **Содержание программы курса**

### **10 класс**

#### **Основные управляющие конструкции.**

Тема 1.1. Первые программы.

Повторение: понятие алгоритма, свойства алгоритма, исполнитель алгоритма (виды, основные характеристики), способы записи алгоритма.

Структура программы. Операторы ввода-вывода. Переменные.

Ключевые слова: компилятор, препроцессор, отдельная компиляция, вывод данных, поток вывода, поток ввода, пространство имен, символьная строка.

Тема 1.2. Диалоговые программы.

Диалоговые программы. Переменные и их типы. Сумма чисел.

Ключевые слова: ввод данных, переменная, объявление переменной, входной поток.

Тема 1.3. Компьютерная графика.

Библиотека TX Library. Управляем пикселями. Линии и фигуры. Замкнутые фигуры.

Ключевые слова: окно, координаты, оси координат, пиксель, цвет контура, цвет заливки, код цвета, прозрачный цвет.

Тема 1.4. Процедуры.

Длинная программа. Рефакторинг. Процедуры с параметрами.

Ключевые слова: подпрограмма, процедура, рефакторинг, аргументы, параметры, базовая точка.

Тема 1.5. Обработка целых чисел.

Предельные значения чисел. Различие между вещественными и целыми числами. Дискретное представление чисел. Программное повышение точности вычислений.

Хранение в памяти целых чисел: со знаком, без знака. Повторение: получение дополнительного кода числа.

Арифметические операции с целыми числами: сложение и вычитание, умножение и деление. Операция сравнения чисел. Повторение: поразрядные логические операции, сдвиги.

Ключевые слова: арифметические выражения, частное, остаток, форматный вывод, случайные числа, зерно.

Тема 1.6. Обработка вещественных чисел.

Повторение: хранение в памяти вещественных чисел. Нормализованное представление вещественных чисел в компьютере.

Арифметические операции с вещественными числами: сложение и вычитание, умножение и деление.

Ключевые слова: вещественное число, научный формат, мантисса, форматный вывод, округление.

Тема 1.7. Ветвления.

Условный оператор. Полная и неполная формы записи условного оператора. Сложные условия в условном операторе и их применение в написании программ. Множественный выбор - переключатель switch.

Ключевые слова: условный оператор, полная форма, неполная форма, составной оператор, вложенный условный оператор, логические переменные.

## Тема 1.8. Циклы.

Понятие циклического алгоритма. Циклы с предусловием. Циклы с постусловием. Циклы с переменными (счетчиком). Вложенные циклы.

Ключевые слова: цикл с предусловием, цикл с постусловием, цикл по переменной.

## Тема 1.9. Анимация

Принципы анимации. Рисуем шарик. Начальное положение. Анимация движения. Обработка нажатия клавиши. Ключевые слова: анимация, процедура, пауза, нажатие клавиши.

Учащиеся должны знать:

- понятие и свойства алгоритма, способы записи, типы алгоритмических структур;
- структуру программы, операторы ввода-вывода, переменные;
- типы переменных, стандартные функции;
- условный оператор, циклы;
- принципы анимации.

Учащиеся должны уметь:

- применять полученные знания к решению задач;
- применять операторы ввода-вывода в программе при создании диалога с пользователем в режиме компиляции;
- соблюдать структуру программы;
- применять стандартные функции к решению задач;
- владеть приемами применения типов переменных;
- составлять программы с использованием конструкций ветвления и циклов;
- применять принципы анимации при создании анимированных объектов.

Предметные результаты

- Владение универсальным языком программирования высокого уровня (по выбору), представлениями о базовых типах данных и структурах данных; умением использовать основные управляющие конструкции.
- Владение умением понимать программы, написанные на выбранном для изучения универсальном алгоритмическом языке высокого уровня; знанием основных конструкций программирования; умением анализировать алгоритмы с использованием таблиц.
- Владение навыками алгоритмического мышления и понимание необходимости формального описания алгоритмов.

- Овладение понятием сложности алгоритма, знание основных алгоритмов обработки числовой и текстовой информации.

## **Процедуры и функции — элементы структуризации программ.**

### **Массив — фундаментальная структура данных.**

#### Тема 2.1. Процедуры.

Вспомогательные алгоритмы. Процедуры. Запись процедуры на языке программирования. Процедура с параметрами.

Ключевые слова: процедура, интерфейс, реализация, параметр, локальная переменная, глобальная переменная, область видимости, передача по ссылке.

#### Тема 2.2. Рекурсия.

Рекурсия. Примеры рекурсий в математике и литературе. Рекурсивная процедура (функция). Использование рекурсий при написании программ.

Ключевые слова: рекурсивная процедура, фрактал, базовые объекты, условия останова, анимация

#### Тема 2.3. Функции.

Функция. Виды функций. Запись функций на языке программирования.

Ключевые слова: функция, параметры, вызов функции, результат функции, рекурсивная функция.

#### Тема 2.4. Символьные строки.

Символьная строка. Операции со строками: сцепление, удаление, копирование элементов. Функции поиска подстроки. Преобразование из строки в число и наоборот. Применение строковых данных в процедурах и функциях. Рекурсивный перебор символов. Сравнение и сортировка строк.

Ключевые слова: символьная строка, длина строки, сцепление строк, выход за границы строки, подстрока, удаление символов, вставка символов, поиск подстроки, замена подстроки, преобразование типов.

#### Тема 2.5. Массивы.

Массивы в C++. Обращение к элементу массива. Перебор элементов массива. Вывод и ввод массива. Заполнение массива случайными числами. Алгоритмы обработки массивов.

Ключевые слова: массив, индекс элемента, значение элемента, константа, заполнение массива, вывод массива, ввод массива.

#### Тема 2.6. Используем массивы.

Используем массивы. Игра «Стрельба по тарелкам».

Ключевые слова: массив, константа, инициализация, глобальные переменные, обработчик события.

## Тема 2.7. Матрицы.

Матрица. Основные операции с матрицами: объявление, заполнение, вывод на экран. Обработка элементов матрицы.

Ключевые слова: матрица, строка, столбец, перебор элементов, вложенный цикл, квадратная матрица, главная диагональ, побочная диагональ, перестановка строк.

## Тема 2.8. Системы управления версиями. Основные приемы работы с Git.

Системы управления версиями. Основные приемы работы с Git. Операции с файлами. Работа с удалённым архивом.

Учащиеся должны знать:

- применение вспомогательных алгоритмов: процедур и функций.
- применений рекурсивных алгоритмов к решению задач;
- понятие массива; основные операции с массивами: объявление, заполнение, вывод на экран;
- методы сортировки массивов: метод пузырька, метод выбора, быстрая сортировка;
- понятие символьной строки и операции со строками: объединение, удаление, копирование элементов строк;
- понятие матрицы и основные операции с матрицами: объявление, заполнение, вывод на экран, обработка элементов матрицы.

Учащиеся должны уметь:

- применять полученные знания к решению задач;
- осуществлять поиск и обработку элементов в массиве, определять минимальный и максимальный элементы массива, осуществлять реверс массива, сдвиг элементов массива, отбор элементов массива по условиям, сортировку в массивах, двоичный поиск в массиве;
- осуществлять операции со строками: объединение, удаление, копирование элементов, функции поиска подстроки, преобразование из строки в число и наоборот;
- применять строковые данные в процедурах и функциях;
- осуществлять рекурсивный перебор символов, сравнивать и сортировать строки;
- производить основные операции с матрицами: объявление, заполнение, вывод на экран, обработку элементов матрицы.

Предметные результаты

- Владение универсальным языком программирования высокого уровня (по выбору), представлениями о базовых типах данных и структурах данных; умением использовать основные управляющие конструкции.

- Владение умением понимать программы, написанные на выбранном для изучения универсальном алгоритмическом языке высокого уровня; знанием основных конструкций программирования; умением анализировать алгоритмы с использованием таблиц.
- Владение навыками алгоритмического мышления и понимание необходимости формального описания алгоритмов.
- Овладение понятием сложности алгоритма, знание основных алгоритмов обработки числовой и текстовой информации, алгоритмов поиска и сортировки.

## 11 класс

### Динамические структуры данных.

#### Тема 3.1. Простые алгоритмы сортировки.

Сортировка в массивах. Методы сортировки: метод пузырька, метод выбора, сортировка вставками. Массивы в подпрограммах.

Ключевые слова: сортировка, метод пузырька, сортировка вставками, перестановка элементов, сложность алгоритма.

#### Тема 3.2. Быстрые алгоритмы сортировки и поиска.

Сортировка в массивах. Методы сортировки: сортировка слиянием, быстрая сортировка. Стандартная сортировка на языке C++. Двоичный поиск.

Ключевые слова: сортировка слиянием, быстрая сортировка, двоичный поиск, лямбда-функция

#### Тема 3.3. Обработка файлов.

Понятие файла. Типы файлов. Этапы работы с файлами: открытие файла, запись в файл, удаление записей из файла, чтение из файла, закрытие файла. Обработка массивов, записанных в файле. Обработка строк, записанных в файле. Обработка смешанных данных, записанных в файле.

Ключевые слова: файл, файловый поток, открытие файла, закрытие файла, чтение из файла, запись в файл, конец файла, аргументы командной строки.

#### Тема 3.4. Целочисленные алгоритмы.

Целочисленные алгоритмы. Решето Эратосфена. Длинные числа.

Ключевые слова: решето Эратосфена, длинные числа, разряды, перенос.

#### Тема 3.5. Динамические массивы и словари.

Понятие динамического массива. Объявление и заполнение динамического массива. Использование динамических массивов в подпрограммах. Расширение массива. Принципы работы динамического массива. Тип `vector` из библиотеки STL. Итераторы. Словари. Перебор элементов словаря.



Ключевые слова: динамический массив, указатель, контейнер, вектор, словарь, ключ, значение, перебор элементов, сортировка по ключу, итератор.

### Тема 3.6. Структуры.

Понятие структуры. Объявление структур. Обращение к полю структуры. Работа с файлами. Сортировка.

Ключевые слова: структура, поле, точечная запись, сортировка, ключ.

### Тема 3.7. Стеки, очереди, деки

Понятие стека. Использование динамического массива для создания стека. Вычисление арифметических выражений. Скобочные выражения. Очередь и деки как разновидности линейного списка.

Ключевые слова: стек, очередь, дек.

### Тема 3.8. Деревья.

Дерево. Двоичное дерево. Дерево поиска. Обход двоичного дерева. Вычисление арифметических выражений. Использование связанных структур. Хранение двоичного дерева в массиве.

Ключевые слова: дерево, двоичное дерево, обход в глубину, обход в ширину, дерево поиска, ключ, хранение в массиве, модуль, проект, заголовочный файл.

### Тема 3.9. Графы

Понятие графа. «Жадные» алгоритмы. Алгоритм Прима-Крускала. Поиск кратчайших путей в графе. Алгоритм Дейкстры. Алгоритм Флойда-Уоршелла.

Ключевые слова: жадный алгоритм, остовное дерево, задача коммивояжера, Гамильтонов цикл, полный перебор, рекурсия, хвостовая рекурсия, случайный поиск, глобальные переменные, структуры.

### Тема 3.10. Динамическое программирование.

Динамическое программирование. Поиск оптимального решения. Количество решений.

Ключевые слова: динамическое программирование, перебор вариантов, мемоизация, оптимальная программа, редактирование строк, расстояние Левенштейна.

Учащиеся должны знать:

- понятие файла, типы, этапы работы;
- алгоритмы работы с файлами;
- алгоритм поиска простых чисел с помощью «решета Эратосфена»;

- понятие «длинного числа», принципы хранения и выполнения операций с «длинными» числами;
- понятие структуры (записи, множества), основные операции со структурами;
- понятие «дерево» и области применения этой структуры данных;
- понятия «граф», «узел», «ребро»;
- простые алгоритмы на графах;
- принципы динамического программирования.

Учащиеся должны уметь:

- осуществлять обработку массивов строк, смешанных данных, записанных в файле;
- использовать текстовый и типизированные файлы;
- программировать простые операции с «длинными» числами;
- использовать различные структуры, грамотно выбирать структуру для конкретной задачи;
- программировать простые алгоритмы на графах;
- программировать алгоритмы, использующие динамическое программирование.

Предметные результаты

- Владение универсальным языком программирования высокого уровня (по выбору), представлениями о базовых типах данных и структурах данных; умением использовать основные управляющие конструкции.
- Владение умением понимать программы, написанные на выбранном для изучения универсальном алгоритмическом языке высокого уровня; знанием основных конструкций программирования; умением анализировать алгоритмы с использованием таблиц.
- Владение навыками алгоритмического мышления и понимание необходимости формального описания алгоритмов.
- Овладение понятием сложности алгоритма, знание основных алгоритмов обработки числовой и текстовой информации, алгоритмов поиска и сортировки.

Динамические структуры данных. Связанный список. Создание элемента списка. Добавление узла. Проход по списку. Поиск узла в списке. Алфавитно-частотный словарь. Удаление узла. Барьеры. Двусвязный список. Операции с двусвязным списком. Циклические списки.

Стек. Реализация стека с помощью массива. Реализация стека с помощью списка. Системный стек в программах. Очередь. Реализация очереди с помощью массива. Реализация очереди с помощью списка. Дек.

Что такое деревья? Реализация двоичных деревьев в языке Си. Поиск с помощью дерева. Разбор арифметического выражения. Дерево игр.

Основные понятия. Задача Прима-Крускала. Кратчайший путь. Оптимальное размещение. Задача коммивояжера. Задача о паросочетаниях.

Тема 4.1. Классы и объекты в C++.

Классы и объекты в C++: новая задача и ее анализ. Класс CMap. Пишем свой конструктор. Рефакторинг. Рисуем карты.

Ключевые слова: класс, объект, экземпляр, модификатор доступа, конструктор, инициализация, метод, рефакторинг.

Тема 4.2. Программа с классами (практикум).

Класс CCar. Рефакторинг. Основная программа. Разбиение на модули. Основная программа. Разбиение на модули.

Ключевые слова: класс, объект, ссылка, конструктор, метод, рефакторинг.

Тема 4.3. Инкапсуляция.

Объект защищает свои данные. Изменение внутреннего устройства. Свойство «только для чтения». Свойства в C#.

Ключевые слова: инкапсуляция, поле, метод, модификатор доступа, интерфейс, свойство.

Тема 4.4. Наследование.

Моделирование жизни в океане. Иерархия классов. Базовый класс. Абстрактный класс. Защищенные поля и методы (protected). Неподвижные объекты. Подвижные объекты. Рыбы. Хищники. Вспомогательные процедуры и функции. Основная программа.

Ключевые слова: класс, иерархия классов, базовый класс, наследование, абстрактный класс, виртуальный метод, защищенные поля, утечка памяти.

Тема 4.5. Полиморфизм.

Полиморфизм в действии. Класс Океан. Деструктор.

Ключевые слова: базовый класс, класс-наследник, виртуальный метод, полиморфизм, позднее связывание, таблица виртуальных методов, деструктор.

Тема 4.6. Взаимодействие объектов.

Столкновения. Изменение базового класса. Изменение других классов. Умные указатели.

Ключевые слова: столкновение, «мертвые» объекты, итератор, преобразование типов.

Тема 4.7. Простая программа на C#.

RAD – среды для разработки программ. Язык C# и среда .NET. Проект C#. Свойства объектов. Обработчики событий.

Ключевые слова: RAD – среда, среда NET., язык C#, виртуальная машина, проект, форма, свойство, событие, обработчик события, компонент, статистический класс.

Тема 4.8. Использование компонентов.

Программы для просмотра рисунков. Ввод и вывод данных. Обработка ошибок.

Ключевые слова: компонент, свойство, событие, обработчик события, родительский объект, дочерний объект, поле ввода, метка, статический метод, обработка ошибок, исключение.

Тема 4.9. Создание новых классов.

Разработка программы для вычисления арифметических выражений: модель, новый класс. Методы класса. Вычисление арифметических выражений: представление. Добавляем новый компонент. Изменение поведения компонента.

Ключевые слова: модуль, модель, представление, класс, компонент, статический класс, методы класса, свойство.

Учащиеся должны знать:

- понятия «объект», «свойства объектов»;
- понятие «классы», «иерархия классов их классификацию»;
- понятие «базовый класс, классы-наследники, модульность»;
- принципы работы в RAD-средах;
- сообщения между объектами.

Учащиеся должны уметь:

- составлять простые программы в RAD-средах;
- оценивать вычислительную сложность изученных алгоритмов.

Понятие объектно-ориентированного программирования. Объекты. Классы.

Создание объектов в программе.

Скрытие внутреннего устройства.

Иерархия классов. Классификация. Иерархия логических элементов. Базовый класс. Классы-наследники. Модульность. Сообщения между объектами.

Программы с графическим интерфейсом. RAD-среды для разработки программ.

Основы программирования в RAD-средах. Общий подход. Создание простейшей программы. Свойства объектов. Обработчики событий.

Программа с компонентами. Ввод и вывод данных. Обработка ошибок.

Модификация компонентов.

Разработка программы вычисления арифметического выражения, записанного в символьной строке.

**Тематическое планирование с указанием количества часов на освоение  
каждой темы по 1 часа в неделю в 10 и 11 классах**

(всего 68 часов)

№ п/п	Наименование разделов и тем	Количество часов
<b>10 класс</b>		
<b>1.</b>	<b>Основные управляющие конструкции.</b>	<b>16</b>
1.1	Первые программы.	1
1.2	Диалоговые программы.	2
1.3	Компьютерная графика.	2
1.4	Процедуры.	2
1.5	Обработка целых чисел.	2
1.6	Обработка вещественных чисел.	1
1.7	Ветвления.	2
1.8	Циклы.	1
1.9	Анимация.	3
<b>2.</b>	<b>Процедуры и функции — элементы структуризации программ. Массив — фундаментальная структура данных.</b>	<b>19</b>
2.1	Процедуры.	2
2.2	Рекурсия.	2
2.3	Функции.	2
2.4	Символьные строки.	4
2.5	Массивы.	1
2.6	Используем массивы.	1
2.7	Матрицы.	4
2.8	Системы управления версиями. Основные приемы работы с Git.	3
<b>Итого</b>		<b>35</b>
<b>11 класс</b>		
<b>3.</b>	<b>Динамические структуры данных.</b>	<b>18</b>

3.1	Простые алгоритмы сортировки.	1
3.2	Быстрые алгоритмы сортировки и поиска.	1
3.3	Обработка файлов.	1
3.4	Целочисленные алгоритмы.	1
3.5	Динамические массивы и словари.	3
3.6	Структуры.	2
3.7	Стеки, очереди, деки.	1
3.8	Деревья.	3
3.9	Графы.	3
3.1 0	Динамическое программирование.	2
<b>4.</b>	<b>Объектно-ориентированное программирование.</b>	<b>15</b>
4.1	Классы и объекты в C++.	1
4.2	Программа с классами (практикум).	2
4.3	Инкапсуляция.	1
4.4	Наследование.	3
4.5	Полиморфизм.	2
4.6	Взаимодействие объектов.	2
4.7	Простая программа на C#.	1
4.8	Использование компонентов.	1
4.9	Создание новых классов.	2
<b>Итого</b>		<b>33</b>
Всего		68

## Описание учебно-методического и материально-технического обеспечения учебного процесса

Для реализации учебного курса «Программирование» необходимо наличие компьютерного класса в соответствующей комплектации:

Требования к комплектации компьютерного класса

Наиболее рациональным с точки зрения организации деятельности детей в школе является установка в компьютерном классе 13–15 компьютеров (рабочих мест) для школьников и одного компьютера (рабочего места) для педагога.

Предполагается объединение компьютеров в локальную сеть с возможностью выхода в Интернет, что позволяет использовать сетевые цифровые образовательные ресурсы.

Минимальные требования к техническим характеристикам каждого компьютера следующие:

- процессор – не ниже *Celeron* с тактовой частотой 2,8 ГГц;
- оперативная память – не менее 1 Гб;
- жидкокристаллический монитор с диагональю не менее 15 дюймов;
- жёсткий диск – не менее 80 Гб;
- клавиатура;
- мышь;
- устройство для чтения компакт-дисков (желательно);
- аудиокарта и акустическая система (наушники или колонки).

Кроме того, в кабинете информатики должны быть:

- принтер на рабочем месте учителя;
- проектор на рабочем месте учителя;
- сканер на рабочем месте учителя;
- телекоммуникационный блок,
- устройства, обеспечивающие подключение к сети;
- устройства вывода звуковой информации;
- устройства для ручного ввода текстовой информации и манипулирования экранными элементами;
- устройства создания графической информации;
- устройства для создания музыкальной информации;
- устройства для записи (ввода) визуальной и звуковой информации.

Требования к программному обеспечению компьютеров

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



- интегрированная среда для программирования на языках С и С++, работающая под управлением операционной системы Windows. Среда Dev-C++, распространяемая свободно с исходными кодами (на Delphi) по лицензии GPL Dev-C++ <http://www.bloodshed.net/dev/devcpp.html> ,

а также:

- текстовый редактор (*Блокнот* или *Gedit*) и текстовый процессор (*Word* или *OpenOffice.org Writer*);
- табличный процессор (*Excel* или *OpenOffice.org Calc*);
- средства для работы с баз данных (*Access* или *OpenOffice.org Base*);
- графический редактор *Gimp* (<http://gimp.org>);
- редактор звуковой информации *Audacity* (<http://audacity.sourceforge.net>);
- среда программирования *КуМир* (<http://www.niisi.ru/kumir/>);
- среда программирования *FreePascal* (<http://www.freepascal.org/>);
- среда программирования *Lazarus* (<http://lazarus.freepascal.org/>)

и другие программные средства.

#### **Дополнительное программное обеспечение**

- файловый менеджер (в составе операционной системы или др.);
- антивирусная программа;
- программа-архиватор;
- клавиатурный тренажер;
- звуковой редактор;
- простая геоинформационная система;
- виртуальные компьютерные лаборатории;
- программа-переводчик;
- система оптического распознавания текста;
- мультимедиа проигрыватель (входит в состав операционных систем или др.);
- почтовый клиент (входит в состав операционных систем или др.);
- браузер (входит в состав операционных систем или др.).

## Учебно-методические средства обучения

### Основная литература

1. Поляков К.Ю. Программирование. Python. C++. Часть 1: учебное пособие /К.Ю. Поляков. – М.:БИНОМ. Лаборатория знаний, 2019.- 144 с.: ил.
2. Поляков К.Ю. Программирование. Python. C++. Часть 2: учебное пособие /К.Ю. Поляков. – М.:БИНОМ. Лаборатория знаний, 2019.- 176 с.: ил.
3. Поляков К.Ю. Программирование. Python. C++. Часть 3: учебное пособие /К.Ю. Поляков. – М.:БИНОМ. Лаборатория знаний, 2019.- 208 с.: ил.
4. Поляков К.Ю. Программирование. Python. C++. Часть 4: учебное пособие /К.Ю. Поляков. – М.:БИНОМ. Лаборатория знаний, 2019.- 192 с.: ил.

### Дополнительная литература

1. Страуструп Б. Программирование: принципы и практика с использованием C++, 2-е изд.; Пер. с англ. - М.: ООО «И. Д. Вильямс», 2016.
2. Липпман С. Б., Лажоие Ж., Му Б.Э. Язык программирования C++. Базовый курс, 5-е изд. Пер. с англ. - М.: ООО «И. Д. Вильямс», 2014.
3. Шилдт Г. C++. Базовый курс, 3-е изд. Пер. с англ. - М.: ООО «И. Д. Вильямс», 2010.
4. Лафоре Р. Объектно-ориентированное программирование в C++, 4-е изд. Пер. с англ. – СПб.: Питер, 2004.
5. Мюссер Д., Дердж Ж., Сейни А. C++ и STL: справочное руководство, 2-е изд. Пер. с англ. - М.: ООО «И. Д. Вильямс», 2010.
6. Литвиненко Н. А. Технология программирования на C++. Начальный курс. – СПб.: БХВ-Петербург, 2010. – 281 с.
7. Романов Е.Л. Си++. От дилетанта до профессионала – СПб.: БХВ-Петербург, 2014. – 600 с.
8. Федеральный государственный образовательный стандарт среднего (полного) общего образования. Утвержден приказом Министерства образования и науки Российской Федерации от 17 мая 2012 г. № 413.
9. К.Ю. Поляков, Е.А. Еремин. Информатика. 10-11 классы. Программа для старшей школы. Углубленный уровень. — М.: Бином, 2016.
10. В.И. Филиппов старший преподаватель кафедры информационно-коммуникационных технологий Академии социального управления Московской области Модульное планирование курса «Информатика и ИКТ» для 10-11-ых классов. Общие рекомендации по составлению рабочей программы по предмету «Информатика и ИКТ» - <http://kpolyakov.spb.ru/school/probook/program.htm>

### Аннотация к рабочей программе

1	Полное наименование Рабочей программы	Программирование (элективный курс)
2	На каком уровне образования реализуется	СОО
3	Нормативная основа для разработки Рабочей программы	<p>Закон РФ № 273 «Об образовании в РФ» от 29.12.2012 г. Федеральный государственный образовательный стандарт среднего общего образования (далее ООП), (утверждён приказом Министерства образования и науки РФ от 17 мая 2012 года № 413, с изменениями и дополнениями 29 декабря 2014 г., 31 декабря 2015 г., 29 июня 2017 г.)</p> <p>Основная образовательная программа среднего общего образования Предуниверсария ГУУ, Учебный план.</p>
4	Срок реализации Рабочей программы	2 года
5	Планируемые результаты изучения учебного предмета	<ul style="list-style-type: none"> <li>• самостоятельно определять цели деятельности и составлять планы деятельности; самостоятельно осуществлять, контролировать и корректировать деятельность; использовать все возможные ресурсы для достижения поставленных целей и реализации планов деятельности; выбирать успешные стратегии в различных ситуациях;</li> <li>• продуктивно общаться и взаимодействовать в процессе совместной деятельности, учитывать позиции других участников деятельности, эффективно разрешать конфликты;</li> <li>• владеть навыками познавательной, учебно-исследовательской и проектной деятельности, навыками разрешения проблем; осуществлять самостоятельный поиск методов решения практических задач;</li> <li>• вести самостоятельную информационно-познавательную</li> </ul>

		<p>деятельность, включая ориентирование в различных источниках информации, критически оценивать и интерпретировать информацию, получаемую из различных источников;</p> <ul style="list-style-type: none"><li>• применять навыки алгоритмического мышления и понимать необходимость применения формального описания алгоритмов;</li><li>• использовать понятие сложности алгоритма, применять основные алгоритмы обработки числовой и текстовой информации, алгоритмы поиска и сортировки;</li><li>• владеть приёмами написания программы в среде Dev-C++ для решения прикладных задач с использованием основных конструкций программирования; осуществлять отладку таких программ; тестировать и оптимизировать готовые прикладные компьютерные программы;</li></ul>
--	--	--