

*ПРИНЯТО*

на заседании Педагогического совета  
Протокол № 8  
от 10.08.2020 г.

*УТВЕРЖДЕНО*

приказом директора  
АНО ОШ ЦПМ  
От 16.08.2020 г. № 52/9-ОД20

## **РАБОЧАЯ ПРОГРАММА**

по предмету «**Информатика**»  
для обучающихся 10 – 11 классов  
(профильный уровень)  
на 2020 – 2021 учебный год

Составитель:

М.С. Густокашин

Москва, 2020 год

**Оглавление**

Планируемые результаты освоения учебного предмета .....	3
Содержание разделов и тем курса .....	7
Раздел 1. Язык программирования Python .....	7
Раздел 2. Алгоритмы и структуры данных для олимпиад по информатике начального уровня .....	8
Раздел 3. Язык программирования C++ .....	10
Тематическое планирование курсов в рамках учебного предмета.....	11

## Планируемые результаты освоения учебного предмета

### Обучающийся научится:

– кодировать и декодировать тексты по заданной кодовой таблице; строить неравномерные коды, допускающие однозначное декодирование сообщений, используя условие Фано; понимать задачи построения кода, обеспечивающего по возможности меньшую среднюю длину сообщения при известной частоте символов, и кода, допускающего диагностику ошибок;

– строить логические выражения с помощью операций дизъюнкции, конъюнкции, отрицания, импликации, эквиваленции; выполнять эквивалентные преобразования этих выражений, используя законы алгебры логики (в частности, свойства дизъюнкции, конъюнкции, правила де Моргана, связь импликации с дизъюнкцией);

– строить таблицу истинности заданного логического выражения; строить логическое выражение в дизъюнктивной нормальной форме по заданной таблице истинности; определять истинность высказывания, составленного из элементарных высказываний с помощью логических операций, если известна истинность входящих в него элементарных высказываний; исследовать область истинности высказывания, содержащего переменные; решать логические уравнения;

– строить дерево игры по заданному алгоритму; строить и обосновывать выигрышную стратегию игры;

– записывать натуральные числа в системе счисления с данным основанием; использовать при решении задач свойства позиционной записи числа, в частности признак делимости числа на основание системы счисления;

– записывать действительные числа в экспоненциальной форме; применять знания о представлении чисел в памяти компьютера;

– описывать графы с помощью матриц смежности с указанием длин ребер (весовых матриц); решать алгоритмические задачи, связанные с анализом графов, в частности задачу построения оптимального пути между вершинами ориентированного ациклического графа и определения количества различных путей между вершинами;

– формализовать понятие «алгоритм» с помощью одной из универсальных моделей вычислений (машина Тьюринга, машина Поста и др.); понимать содержание тезиса Черча–Тьюринга;

– понимать и использовать основные понятия, связанные со сложностью вычислений (время работы и размер используемой памяти при заданных исходных данных; асимптотическая сложность алгоритма в зависимости от размера исходных данных); определять сложность изучаемых в курсе базовых алгоритмов;

– анализировать предложенный алгоритм, например определять, какие результаты возможны при заданном множестве исходных значений и при каких исходных значениях возможно получение указанных результатов;

– создавать, анализировать и реализовывать в виде программ базовые алгоритмы, связанные с анализом элементарных функций (в том числе приближенных вычислений), записью чисел в позиционной системе счисления, делимостью целых чисел; линейной обработкой последовательностей и массивов чисел (в том числе алгоритмы сортировки), анализом строк, а также рекурсивные алгоритмы;

- применять метод сохранения промежуточных результатов (метод динамического программирования) для создания полиномиальных (не переборных) алгоритмов решения различных задач; примеры: поиск минимального пути в ориентированном ациклическом графе, подсчет количества путей;
- создавать собственные алгоритмы для решения прикладных задач на основе изученных алгоритмов и методов;
- применять при решении задач структуры данных: списки, словари, деревья, очереди; применять при составлении алгоритмов базовые операции со структурами данных;
- использовать основные понятия, конструкции и структуры данных последовательного программирования, а также правила записи этих конструкций и структур в выбранном для изучения языке программирования;
- использовать в программах данные различных типов; применять стандартные и собственные подпрограммы для обработки символьных строк; выполнять обработку данных, хранящихся в виде массивов различной размерности; выбирать тип цикла в зависимости от решаемой подзадачи; составлять циклы с использованием заранее определенного инварианта цикла; выполнять базовые операции с текстовыми и двоичными файлами; выделять подзадачи, решение которых необходимо для решения поставленной задачи в полном объеме; реализовывать решения подзадач в виде подпрограмм, связывать подпрограммы в единую программу; использовать модульный принцип построения программ; использовать библиотеки стандартных подпрограмм;
- применять алгоритмы поиска и сортировки при решении типовых задач;
- выполнять объектно-ориентированный анализ задачи: выделять объекты, описывать на формальном языке их свойства и методы; реализовывать объектно-ориентированный подход для решения задач средней сложности на выбранном языке программирования;
- выполнять отладку и тестирование программ в выбранной среде программирования; использовать при разработке программ стандартные библиотеки языка программирования и внешние библиотеки программ; создавать многокомпонентные программные продукты в среде программирования;
- устанавливать и деинсталлировать программные средства, необходимые для решения учебных задач по выбранной специализации;
- пользоваться навыками формализации задачи; создавать описания программ, инструкции по их использованию и отчеты по выполненным проектным работам;
- разрабатывать и использовать компьютерно-математические модели; анализировать соответствие модели реальному объекту или процессу; проводить эксперименты и статистическую обработку данных с помощью компьютера; интерпретировать результаты, получаемые в ходе моделирования реальных процессов; оценивать числовые параметры моделируемых объектов и процессов;
- понимать основные принципы устройства и функционирования современных стационарных и мобильных компьютеров; выбирать конфигурацию компьютера в соответствии с решаемыми задачами;
- понимать назначение, а также основные принципы устройства и работы современных операционных систем; знать виды и назначение системного программного обеспечения;

- владеть принципами организации иерархических файловых систем и именования файлов; использовать шаблоны для описания группы файлов;
- использовать на практике общие правила проведения исследовательского проекта (постановка задачи, выбор методов исследования, подготовка исходных данных, проведение исследования, формулировка выводов, подготовка отчета); планировать и выполнять небольшие исследовательские проекты;
- использовать динамические (электронные) таблицы, в том числе формулы с использованием абсолютной, относительной и смешанной адресации, выделение диапазона таблицы и упорядочивание (сортировку) его элементов; построение графиков и диаграмм;
- владеть основными сведениями о табличных (реляционных) базах данных, их структуре, средствах создания и работы, в том числе выполнять отбор строк таблицы, удовлетворяющих определенному условию; описывать базы данных и средства доступа к ним; наполнять разработанную базу данных;
- использовать компьютерные сети для обмена данными при решении прикладных задач;
- организовывать на базовом уровне сетевое взаимодействие (настраивать работу протоколов сети TCP/IP и определять маску сети);
- понимать структуру доменных имен; принципы IP-адресации узлов сети;
- представлять общие принципы разработки и функционирования интернет-приложений (сайты, блоги и др.);
- применять на практике принципы обеспечения информационной безопасности, способы и средства обеспечения надежного функционирования средств ИКТ; соблюдать при работе в сети нормы информационной этики и права (в том числе авторские права);
- проектировать собственное автоматизированное место; следовать основам безопасной и экономичной работы с компьютерами и мобильными устройствами; соблюдать санитарно-гигиенические требования при работе за персональным компьютером в соответствии с нормами действующих СанПиН.

**Обучающийся получит возможность научиться:**

- применять коды, исправляющие ошибки, возникшие при передаче информации; определять пропускную способность и помехозащищенность канала связи, искажение информации при передаче по каналам связи, а также использовать алгоритмы сжатия данных (алгоритм LZW и др.);
- использовать графы, деревья, списки при описании объектов и процессов окружающего мира; использовать префиксные деревья и другие виды деревьев при решении алгоритмических задач, в том числе при анализе кодов;
- использовать знания о методе «разделяй и властвуй»;
- приводить примеры различных алгоритмов решения одной задачи, которые имеют различную сложность; использовать понятие переборного алгоритма;
- использовать понятие универсального алгоритма и приводить примеры алгоритмически неразрешимых проблем;
- использовать второй язык программирования; сравнивать преимущества и недостатки двух языков программирования;
- создавать программы для учебных или проектных задач средней сложности;
- использовать информационно-коммуникационные технологии при моделировании и анализе процессов и явлений в соответствии с выбранным профилем;

- осознанно подходить к выбору ИКТ-средств и программного обеспечения для решения задач, возникающих в ходе учебы и вне ее, для своих учебных и иных целей;
- проводить (в несложных случаях) верификацию (проверку надежности и согласованности) исходных данных и валидацию (проверку достоверности) результатов натуральных и компьютерных экспериментов;
- использовать пакеты программ и сервисы обработки и представления данных, в том числе – статистической обработки;
- использовать методы машинного обучения при анализе данных; использовать представление о проблеме хранения и обработки больших данных;
- создавать многотабличные базы данных; работе с базами данных и справочными системами с помощью веб-интерфейса.

## Содержание разделов и тем курса

### Раздел 1. Язык программирования Python

#### Тема 1.1. Арифметические операции

Ввод и вывод целых чисел, переменные, операции сложения, вычитания, деления с остатком (в т.ч. для отрицательных чисел), взятие остатка от деления, возведение в степень, именованное переменных.

#### Тема 1.2. Условный оператор

Переменные логического типа, логические выражения, логические связки, условный оператор if, оператор “иначе” else, оператор “иначе если” elif, вложенные условные операторы.

#### Тема 1.3. Цикл while

Цикл “пока” while, понятие пошаговой отладки, задачи о подсчете суммы последовательности и поиске минимума и максимума в последовательности, операторы управления циклом break и continue, else для циклов while, вложенные циклы.

#### Тема 1.4. Вещественные числа и строки

Хранение вещественных чисел в памяти, операции над вещественными числами, погрешности и точность вычислений, преобразования вещественных чисел. Хранение строк в памяти, операции над строками, срезы и стандартные методы обработки строк.

#### Тема 1.5. Функции и рекурсия

Область применения функций, механизм запуска функций, создание собственных функций, передача параметров и получение результата работы функции, локальные и глобальные переменные, выделение памяти в стеке, рекурсия, примеры решения задач на обработку последовательностей с помощью рекурсивных функций.

#### Тема 1.6. Цикл for

Кортежи, итерируемые объекты (кортежи, строки), функция range с одним, двумя и тремя параметрами, вложенные циклы for, инструкции управления циклом for: break и continue, примеры задач, решаемых с помощью циклов for.

#### Тема 1.7. Списки

Понятие списка, отличие от кортежа, изменяемые объекты и ссылки на них, чтение списка и добавление элементов, передача и получение списков в качестве параметров в функции, вложенные списки, рекурсивная обработка вложенных списков, примеры решений задач на списки.

#### Тема 1.8. Использование встроенной сортировки

Метод sort для списков, функция sorted для итерируемых объектов, сортировка составных объектов, параметр reversed, параметр key и примеры его использования, лямбда-функции для сортировок.

#### Тема 1.9. Множества

Множества из чисел, строк и кортежей, основы хеширования объектов, хеш-таблица как структура данных для представления множества, работа с отдельными элементами множеств (добавление, поиск, удаление), работа с множествами (объединение, пересечение, симметрическая разность), создание множеств из других типов данных.

#### Тема 1.10. Словари

Понятие ключ-значение, создание словарей, поиск в словаре, добавление и удаление элементов, проход по всем значениям словаря, создание обратного словаря для поиска по ключу, типы ключей.

### **Тема 1.11. Функциональное программирование**

Зачем нужно функциональное программирование, основные встроенные функции для работы с последовательностями: `map`, `enumerate`, `zip` и др., комбинаторные функции, логические функции для последовательностей, методика написания программ в функциональном стиле.

### **Тема 1.12. Объектно-ориентированное программирование**

Основные понятия ООП: наследование, инкапсуляция и полиморфизм, классы и объекты, методы и поля класса, статические переменные, создание собственных классов, переопределение стандартных методов, иерархия классов, создание классов-наследников, функции `isinstance` и `type`.

## **Раздел 2. Алгоритмы и структуры данных для олимпиад по информатике начального уровня**

### **Тема 2.1. Олимпиадные задачи на моделирование**

Составные части олимпиадной задачи: легенда, форматы ввода и вывода, ограничения и примеры, интуитивное понятие сложности алгоритма, оценка времени работы, построение математической модели по условию задачи.

### **Тема 2.2. Линейный поиск**

Задача поиска минимума и максимума в последовательности, минимум и максимум с дополнительными требованиями, поиск индексов минимального и максимального элементов, поиск элемента по значению, подсчет количества вхождений, второй максимум и минимум, барьерный метод.

### **Тема 2.3. Частичные суммы, обработка двумерных массивов**

Предварительный подсчет суммы на префиксах для быстрого подсчета суммы элементов на отрезке, подсчет произведения на отрезке, перебор соседних элементов в двумерных массивах с помощью массивов `dx`, `dy` (в зависимости от топологии), барьеры вокруг двумерных массивов для избавления от краевых эффектов, заполнение двумерных массивов разными способами: змейкой, спиралью и т.п.

### **Тема 2.4. Квадратичные сортировки**

Сортировка пузырьком, вставка элемента в список, сортировка вставкой, сдвиг элементов, сортировка выбором, оценка сложности сортировок, сложность сортировок по сравнениям и присваиваниям, простые задачи на сортировку и жадный алгоритм.

### **Тема 2.5. Арифметика и арифметические алгоритмы**

Поиск делителей числа, оценка количества делителей, проверка числа на простоту, разложение на простые делители, решето Эратосфена, наибольший общий делитель, наименьшее общее кратное, алгоритм Евклида, быстрое возведение в степень, основные операции над матрицами.

### **Тема 2.6. Простые структуры данных**

Стек, проверка правильности скобочного выражения, проверка корректности xml-документа, подсчет значения арифметического выражения с использованием обратной польской записи, поиск ближайшего меньшего соседа за линейное время для всех элементов массива, использование очереди, использование дека, поиск минимума в скользящем окне за линейное время.



**Тема 2.7. Бинарный поиск**

Реализация бинарного поиска, бинарный поиск в массивах, подсчет числа элементов, бинарный поиск по ответу, бинарный поиск для вещественных чисел, тернарный поиск, бинарный поиск по производной функции.

**Тема 2.8. Эффективные алгоритмы сортировок**

Сортировка подсчетом, поразрядная сортировка, побайтовая сортировка, слияние отсортированных массивов, сортировка слиянием, сортировка во внешней памяти, эффективный подсчет числа перестановок элементов при сортировке пузырьком.

**Тема 2.9. Быстрая сортировка и порядковые статистики**

Понятие опорного элемента, перестановка массива относительно опорного элемента, быстрая сортировка и ее сложность в среднем и худшем случае, оптимизации быстрой сортировки, нахождение порядковой статистики за линейное время, медиана массива.

**Тема 2.10. Методы, использующие сортировку**

Метод двух указателей, сортировка событий, события на прямой и на окружности, отрезки и интервалы событий, подсчет числа пересекающихся событий, двупроходные алгоритмы для получения сертификата, сжатие координат.

**Тема 2.11. Генерация комбинаторных объектов**

Рекурсивная генерация комбинаторных объектов: перестановок, подмножеств, подмножеств заданного размера, разбиение множества на все возможные подмножества, решение NP-полных задач полным перебором и базовые методы оптимизации перебора.

**Тема 2.12. Одномерное динамическое программирование**

Метод динамического программирования, реализация динамического программирования с помощью рекурсии, числа Фибоначчи, обобщенные числа Фибоначчи, наибольшая возрастающая последовательность, задачи подсчета количества решений и поиска оптимального решения с помощью динамического программирования.

**Тема 2.13. Двумерное динамическое программирование**

Использование двух параметров в задачах на динамическое программирование, задача о наибольшей общей подпоследовательности, задачи подсчета числа решений и поиска оптимального решения с использованием двух параметров, динамическое программирование на подотрезках, рекурсия с мемоизацией.

**Тема 2.14. Задача о рюкзаке**

Формулировка задачи о рюкзаке, NP-полнота задачи, методы оптимизации задачи о рюкзаке для случая целых весов предметов, использование кольцевого буфера для экономии памяти, задача о рюкзаке с ценностью предметов, восстановление сертификата в задаче о рюкзаке.

**Тема 2.15. Задачи на лабиринтах**

Определение лабиринта как двумерного массива с проходимыми и непроходимыми клетками, существование пути между клетками, барьерный метод, кратчайший путь между клетками, кратчайший путь с несколькими концами, кратчайший путь с несколькими концами, кратчайший путь с несколькими началами и концами.

**Тема 2.16. Представление графов, обход в ширину**

Способы хранения графов: матрицы смежности, списки ребер, списки смежности, преобразования типов представления графа, обход в ширину на графе и его реализации с помощью списков вершин, находящихся на заданном расстоянии и с помощью очереди.

**Тема 2.17. Обход в глубину и его применения**

Рекурсивная реализация обхода в глубину, проверка графа на связность, дерево обхода в глубину, типы ребер: прямые, обратные и поперечные, поиск компонента

связности, раскраска графа в два цвета, проверка наличия циклов в ориентированном и неориентированном графе, восстановление циклов.

## **Раздел 3. Язык программирования C++**

### **Тема 3.1. Арифметика, условный оператор, циклы**

Ввод и вывод на языке C++, особенности целочисленной арифметики на C++, создание переменных, особенности условных операторов в C++, оператор switch, циклы while, for и do-while.

### **Тема 3.2. C++: структуры данных и функции**

Вектор, итераторы произвольного доступа, стандартные функции для работы с вектором, реаллокация памяти и порча итераторов вектора при расширении, undefined behavior, кортежи, пары, сравнение кортежей и пар.

### **Тема 3.3. C++: символы и строки**

Типы данных char и string в C++, коды символов, таблица ascii, чтение строки до перевода строки и по словам, стандартные методы и функции для строк, строка как массив символов, посимвольный ввод, чтение числовых и символьных данных одновременно.

### **Тема 3.4. C++: указатели и итераторы**

Понятие указателя и ссылки, стандартные функции STL для работы с векторами и массивами, отличие итератора от указателя, односвязные и двусвязные списки, внутреннее устройство структур данных stack, deque и queue в STL.

### **Тема 3.5. C++: set, multiset, map**

Понятие о бинарных деревьях поиска: построение, поиск, вставка и удаление элементов, использование бинарного дерева поиска для реализации множества, реализация бинарного дерева поиска в STL, удаление элементов по значению и итератору, мультимножество, словари.

### Тематическое планирование курсов в рамках учебного предмета

Раздел / тема	Количество ак. ч.
<b>Раздел 1. Язык программирования Python</b>	<b>72</b>
Тема 1.1. Арифметические операции	6
Тема 1.2. Условный оператор	6
Тема 1.3. Цикл while	6
Тема 1.4. Вещественные числа и строки	6
Тема 1.5. Функции и рекурсия	6
Тема 1.6. Цикл for	6
Тема 1.7. Списки	6
Тема 1.8. Использование встроенной сортировки	6
Тема 1.9. Множества	6
Тема 1.10. Словари	6
Тема 1.11. Функциональное программирование	6
Тема 1.12. Объектно-ориентированное программирование	6
<b>Раздел 2. Алгоритмы и структуры данных для олимпиад по информатике начального уровня</b>	<b>102</b>
Тема 2.1. Олимпиадные задачи на моделирование	6
Тема 2.2. Линейный поиск	6
Тема 2.3. Частичные суммы, обработка двумерных массивов	6
Тема 2.4. Квадратичные сортировки	6
Тема 2.5. Арифметика и арифметические алгоритмы	6
Тема 2.6. Простые структуры данных	6
Тема 2.7. Бинарный поиск	6
Тема 2.8. Эффективные алгоритмы сортировок	6
Тема 2.9. Быстрая сортировка и порядковые статистики	6
Тема 2.10. Методы, использующие сортировку	6
Тема 2.11. Генерация комбинаторных объектов	6
Тема 2.12. Одномерное динамическое программирование	6

Тема 2.13. Двумерное динамическое программирование	6
Тема 2.14. Задача о рюкзаке	6
Тема 2.15. Задачи на лабиринтах	6
Тема 2.16. Представление графов, обход в ширину	6
Тема 2.17. Обход в глубину и его применения	6
<b>Раздел 3. Язык программирования C++</b>	<b>30</b>
Тема 3.1. Арифметика, условный оператор, циклы	6
Тема 3.2. C++: структуры данных и функции	6
Тема 3.3. C++: символы и строки	6
Тема 3.4. C++: указатели и итераторы	6
Тема 3.5. C++: set, multiset, map	6