

**МУНИЦИПАЛЬНОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
«СРЕДНЯЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ШКОЛА №10»**

**УТВЕРЖДЕНА**

**Приказом №196 от 30.08.2022 г.**

**Директор МБОУ «СОШ №10» г. Выборг**

**/О. С. Лобанкова/**

**«30» 08 2022 г.**



**ПРИНЯТА**

**На педагогическом совете**

**№1 «30» 08 2022 г.**

**ДОПОЛНИТЕЛЬНАЯ ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА  
технической направленности  
«Олимпиадное программирование»**

**Разработана:**

Методическим объединением  
учителей естественно-научного цикла и математики

**город Выборг  
2022 год**

Программа рассчитана - 4 часа в неделю для группы учащихся 9-11 классов.

Организация курса:

Курс состоит из 64 уроков длительностью 90 минут. Урок продолжительностью 90 минут делится на 2 части по 45 минут с перерывом в 10 минут.

### **Пояснительная записка**

Учитывая слабую стартовую подготовку большинства участников в области олимпиадного программирования, планируется невысокий темп освоения нового материала, решение большого количества задач на пройденные темы.

Подавляющее большинство олимпиад по информатике являются по сути своей олимпиадами по программированию. Это и линейка этапов Всероссийской олимпиады школьников, вузовские олимпиады, отборочные этапы инженерных олимпиад, on-line-турниры и т.д. Задачи этих олимпиад сильно далеки от школьного курса, эффективные методы их решения не включены ни в один учебник профильного курса информатики. По сути, это вузовский материал, который смело может быть отнесён к профессиональной сфере в области программирования.

Решение олимпиадных задач имеет ряд особенностей, которые существенно осложняют их решение.

1. Многие олимпиадные задачи требуют хорошего знания математики, особенно теории чисел, геометрии, комбинаторики, развитого логического мышления.

2. Наличие «легенды» (сюжетной истории, порой фантастической, сюрреалистической) затрудняет понимание сути задания, вынуждает пользователя сначала отделять главное от второстепенного, строить математическую модель, что само по себе весьма нетривиальная задача.

3. Олимпиадные задачи очень формализованы: строгий формат ввода/вывода данных, ограничения на исходные величины, ограничения по времени выполнения и используемой памяти.

4. Есть ряд классических алгоритмов, без знания которых невозможно эффективно решить задачу. Чем больше их в арсенале школьника, тем выше его шансы на успешное выступление на олимпиаде. Требуется не только понять эти алгоритмы, но и научиться реализовывать их на определённом языке программирования. На это уходит масса времени.

5. Проверка решений осуществляется автоматизированной системой тестирования. Это накладывает очень жесткие ограничения на выбор языка программирования, компилятора, ввод и вывод данных из файла и в файл. Задача проверяется на большом количестве тестов, на некоторых олимпиадах зачитывается только полностью правильное решение, прошедшее абсолютно все тесты.

6. Многие олимпиады по программированию являются командными соревнованиями. Команда ограничена в ресурсах (один компьютер на троих), надо уметь разрабатывать стратегию использования этого ресурса.

Планируется, что ученики, посещающие спецкурс, будут принимать активное участие в олимпиадах, потому закладывается большое количество часов на разборы задач.

### **Планируемые результаты изучения курса «Олимпиадное программирование»**

В результате изучения курса «Олимпиадное программирование» обучающийся изучит классические алгоритмы и приемы программирования, основные типы задач и методы их решения, в том числе:

- Алгоритмы над целыми числами.
- Рекурсия.
- Сортировка.

- Переборные задачи.
- Геометрические задачи.
- Численные методы.
- Графы и деревья.
- Текстовые преобразования.

По окончании курса обучающийся будет уметь:

- анализировать текст задачи, строить математическую модель;
- формализовывать математические модели на языке программирования;
- выбирать структуры данных для представления исходных данных и вывода результата;
- реализовать основные структуры данных на языке программирования высокого уровня;
- анализировать и объяснить поведение программ с использованием сложных структур;
- использовать все наиболее важные конструкции программирования;
- применять методы структурной (функциональной) декомпозиции для разделения задачи на подзадачи, выделения функций и процедур;
- реализовать, протестировать и отладить рекурсивные функции и процедуры;
- оценивать трудоемкость алгоритмов и затраты памяти при его реализации;
- строить эффективную структуру тестов;
- разрабатывать генераторы тестов большой размерности;
- отлаживать программы в соответствии с требованиями и ограничениями;
- ориентироваться в системах автоматизированной проверки.

По окончании курса ученик должен знать понятия:

- детерминированного алгоритма;
- вычислительной трудности алгоритма;
- эффективного по времени/памяти алгоритма;
- генератора тестов.

**Категория обучающихся:** ученики общеобразовательных школ 9-11 класса в рамках внеурочной деятельности.

## Содержание учебного предмета 9-11 класс

### 1. Введение в олимпиадное программирование

Виды и типы действующих олимпиад. Различия и классификация заданий.

### 2. Основы теории сложности

Анализ эффективности алгоритма. Основные классы эффективности. Решение простых задач на эффективность.

### 3. Теория чисел в олимпиадных задачах

Целочисленная арифметика. Арифметика остатков. Кольца вычетов. Поиск натуральных делителей. Проверка на простоту. Факторизация. Решето Эратосфена. Задача Иосифа Флавия. Проблема Гольдбаха. Алгоритм Евклида. Трудоемкость алгоритма Евклида. Расширенный алгоритм Евклида. НОД и НОК. Практика решения задач на теорию чисел.

### 4. Алгоритмы поиска

Линейный поиск. Бинарный поиск. Бинарный поиск по ответу. Метод двух указателей. Практика алгоритмов поиска. Парасочетания. Линейные алгоритмы на суммы. Комбинаторика и метод двух указателей.

### **5. Элементы функционального программирования**

Динамическое программирование. Рекурсия в динамике. Последовательность Фибоначчи как основная проблема и решение рекурсивного алгоритма. Мемоизация. Средства динамического кэширования.

### **6. Строковые последовательности**

Обработка строковых последовательностей. Методы быстрого перебора строк. Непрерывные строковые подпоследовательности. Проблема поллиндромной строки. Алгоритм Манакера. Арифметические задачи со строками. Интерактивные задачи.

### **7. Линейные структуры данных**

Массивы префиксных сумм. Линейные структуры данных. Стеки. Амортизированный анализ. Скобочные последовательности. Очередь. Дек. Обратная польская запись. Практика обработки строковых последовательностей. Дек и Сириус (Игры в карты). Дек и очередь с защитой от ошибок. Заливка области.

### **8. Методы сортировки**

Сортировка. Оценка эффективности сортировочных алгоритмов. Метод выбора. Пузырьковая сортировка и сортировка "камнем". Сортировка вставками. Сортировка обменами. Сортировка слиянием. Рекурсия в сортировках. "Быстрая сортировка Хоара". Сортировка. Подведение итогов универсальных сортировочных алгоритмов. К-я порядковая статистика.

### **9. Программирование на графах**

Хэш-таблицы. Куча, сортировка кучей. Деревья. Бинарные деревья. Сортировка подсчётом. Цифровая сортировка. Сортирующие сети. Битоническая сортировка. Поиск в ширину. Алгоритм Дейкстры. Жадные алгоритмы. Двумерное динамическое программирование. Элементы теории групп в динамическом программировании. Алгоритм "к ближайших соседей".

### **10. Дополнительная практика по пройденному материалу**

Олимпиада ЛЭТИ и алгоритмическая математика. "Технокубок". Олимпиадные задачи МИРЭА.

## **Тематический план 9-11 класс**

| <b>№</b> | <b>ТЕМА</b>                               | <b>КОЛИЧЕСТВО ЧАСОВ</b> |
|----------|---|-------------------------|
| 1.       | Введение в олимпиадное программирование   | 2                       |
| 2.       | Основы теории сложности                   | 6                       |
| 3.       | Теория чисел в олимпиадных задачах        | 22                      |
| 4.       | Алгоритмы поиска                          | 14                      |
| 5.       | Элементы функционального программирования | 8                       |
| 6.       | Строковые последовательности              | 12                      |
| 7.       | Линейные структуры данных                 | 21                      |
| 8.       | Методы сортировки                         | 15                      |

|               |  |            |
|---------------|--|------------|
| 9.            | Программирование на графах                       | 22         |
| 10.           | Дополнительная практика по пройденному материалу | 6          |
| <b>ВСЕГО:</b> |  | <b>128</b> |

### **Материально-техническое обеспечение курса**

Персональный компьютер или ноутбук с выходом в сеть интернет.

Среда разработки и компиляторы/интерпретаторы языков программирования высокого уровня на усмотрение ученика: Python, C, C++, Pascal, C#.

Литература для учащегося: не предусмотрена.

Литература для учителя: не предусмотрена.