

**МУНИЦИПАЛЬНОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ
ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
«ВОЛОСОВСКИЙ ЦЕНТР ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ»**

ПРИНЯТО:
решением педагогического совета
Протокол от 31.08.2020 г. №1

Утверждено:
Приказом директора МБУ ДО
«Волосовский ЦИТ»
от 01.09.2020 г. №38

**Дополнительная общеразвивающая программа
технической направленности
«Программирование на языке C++»**

Возраст обучающихся: 13-18 лет
Срок реализации: 1 год
Автор: Иванов Александр Алексеевич,
педагог дополнительного образования
МБУ ДО «Волосовский ЦИТ»

г. Волосово
2020 год

Оглавление

Пояснительная записка	3-4
Учебный план	5
Учебно-тематическое планирование	5-6
Содержание изучаемого курса	7-10
Организационно-педагогические условия реализации образовательной программы	10-11
Средства обучения	11
Планируемые результаты освоения образовательной программы	11-12
Система оценки результатов освоения общеразвивающей программы	13
Список литературы	14
Приложения	15-18

1. Пояснительная записка

Дополнительная общеразвивающая программа технической направленности «Программирование на языке C++» разработана на основе:

- Федерального закона от 29 декабря 2012 года № 273-ФЗ «Об образовании в Российской Федерации»;
- Концепции развития дополнительного образования детей (утверждена Распоряжением Правительства Российской Федерации от 04.09. 2014 года № 1726-р);
- Приказа Министерства Просвещения Российской Федерации от 9 ноября 2018 года №196 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;
- Санитарно-эпидемиологических требований к устройству, содержанию и организации режима работы образовательных организаций дополнительного образования детей (СанПиН 2.4.4.3172-14);
- Стратегии развития воспитания в Российской Федерации на период до 2025 года (№ 996-р от 29.05.15);
- Устава МБУ ДО «Волосовский ЦИТ».

Многие языки программирования требуют однозначного описания последовательности операторов, то есть алгоритма. Между тем школьники, умеющие составить алгоритм и написать по нему программу, обычно значительно легче овладевают и пользовательскими навыками, так как понимают механизм управления компьютером. Они лучше успевают и по другим предметам, поскольку культура их мышления выше, а ассортимент методов выполнения различных работ богаче. В связи с этим представляется достаточно важным привить обучающимся навыки алгоритмического мышления как можно более с раннего возраста.

Актуальность. Содержание курса «Программирование на языке C++» способствует устойчивому и эффективному развитию системного и алгоритмического мышления обучающихся.

Новизна. Есть задача, проблема. Ученику требуется найти решение путем разработки соответствующей программы. Если решение известно, решались аналогичные задачи, то задействуется ассоциативная составляющая интеллекта, работа сводится к набору программы и ее отладке. Если решение неизвестно, то за постановкой задачи следует гипотеза и разработка первоначального варианта программы. Затем она подвергается исследованию, экспериментальной проверке с помощью системы тестов – сравнению ожидаемых результатов и полученных.

Ученику мысленно следует предсказать, предвидеть результаты работы. Наступает фаза или экспериментального опровержения, или экспериментального подтверждения. Т.е. деятельность при разработке программ характеризуется контролируемостью, обоснованностью и целенаправленностью. Оценка своих действий – неперенный атрибут программирования. На каждом шаге ученик имеет возможность осознать, насколько правильно принятое решение, насколько верен ход рассуждений, все ли факты учтены при принятии решения и т.д. Деятельность при программировании можно назвать направленной на получение желаемого результата. Она не просто

активна, она сверхактивна, и мы видим возможность реализации концепции развивающего обучения в полном объеме.

Цель программы – освоение и систематизация знаний, относящихся к построению и описанию объектов и процессов, позволяющих осуществить их компьютерное моделирование в соответствии с требованиями спортивного программирования.

Задачи курса:

Обучающие:

- познакомить обучающихся с основными алгоритмическими конструкциями и правилами их записи, с основными способами организации данных;
- научить обучающихся составлять и записывать алгоритмы с использованием соответствующих алгоритмических конструкций;
- научить распознавать необходимость применения той или иной алгоритмической конструкции при решении задач;
- научить организовывать данные для эффективной алгоритмической обработки;
- научить разрабатывать алгоритмы и реализовывать их на языке программирования;
- научить осуществлять отладку и тестирование программы.

Развивающие:

- формировать новый тип мышления – операционный, который направлен на выбор оптимальных решений;
- предоставление возможности узнать новое в области компьютерного программирования;
- формирование представления о роли компьютерного программирования в развитии общества, изменении содержания и характера деятельности человека.

Воспитательные:

- повышение общекультурного уровня обучающихся;
- выделение и раскрытие роли информационных технологий и компьютеров в развитии современного общества;
- привитие навыков сознательного и рационального использования компьютера в своей учебной, а затем и профессиональной деятельности;
- формирование эмоционально-ценностного отношения к миру, к себе;
- воспитание трудолюбия, инициативности и настойчивости в преодолении трудностей.

Учебная программа разработана для разновозрастных групп от 13 до 18 лет (7 – 11 класс).

Общий объём времени обучения, включая теоретические, практические занятия и контрольную работу составляет 68 академических часов.

2. Учебный план		
Учебный модуль	Количество часов	Формы проведения промежуточной аттестации
Раздел 1. Основы C++	24	Практическая работа, наблюдение, контрольная работа
Раздел 2. Переменные и основные типы данных в C++	14	Практическая работа, наблюдение, контрольная работа
Раздел 3. Операторы в C++	12	Практическая работа, наблюдение, контрольная работа
Раздел 4. Циклы в C++	18	
Всего:	68	

3. Учебно-тематическое планирование		
Наименование раздела, тема		Количество часов
Раздел 1. Основы C++		
1	Повторение изученного материала. Техника безопасности.	2
2	Локальная область видимости.	2
3	Ключевые слова и идентификатор.	2
4	Операторы.	2
5	Базовое форматирование кода.	2
6	Предварительное объявление и прототип функции.	2
7	Многофайловые программы.	2
8	Заголовочные файлы. Директивы препроцессора.	2
9	Header guards. Конфликт имён и std namespace.	2
10	Разработка ваших первых программ.	2
11	Отладка программ: степпинг и точки останова. Отладка программ: стек вызовов и отслеживание переменных.	2
12	Итоговый тест	2
Раздел 2. Переменные и основные типы данных в C++		
13	Размер типов данных и оператор sizeof Тип данных void.	2
14	Целочисленные типы данных	2

	Фиксированный размер целочисленных типов.	
15	Типы данных с плавающей точкой.	2
16	Логический тип данных.	2
17	Символьный тип данных.	2
18	Литералы и магические числа. const, constexpr и символьные константы.	2
19	Итоговый тест.	2
	Раздел 3. Операторы в C++	
20	Приоритет операций и правила ассоциативности.	2
21	Арифметические операторы.	2
22	Инкремент, декремент и побочные эффекты. sizeof, запятая и условный тернарный оператор.	2
23	Операторы сравнения.	2
24	Логические операторы: И, ИЛИ, НЕ.	2
25	Конвертация чисел из двоичной системы в десятичную и наоборот.	2
	Раздел 4. Циклы в C++	
26	Операторы управления потоком выполнения программ	2
27	Операторы условного ветвления if/else	2
28	Оператор switch	2
29	Оператор goto	2
30	Цикл while	2
31	Цикл do while	2
32	Цикл for	2
33	Операторы break и continue	2
34	Итоговый тест.	2

4. Содержание изучаемого курса

Раздел 1. Основы C++ (24 часа)

Тема 1. Повторение изученного материала. Техника безопасности.

Тема 2. Локальная область видимости

Создание переменной, область видимости переменной, локальная область видимости.

Тема 3. Ключевые слова и идентификатор

Список всех ключевых слов в C++ (включая C++17). Два специальных идентификатора: `override` и `final`. Идентификаторы. Польза комментариев в коде

Тема 4. Операторы

Литералы. Унарные операторы. Бинарные операторы. Тернарные операторы

Тема 5. Базовое форматирование кода

Основные рекомендации. Разрыв/перевод строки. Исключения читабельность и логичность кода.

Тема 6. Предварительное объявление и прототип функции

Предыстория. Прототипы функций и предварительное объявление. Предварительно объявили, но не определили. Объявление vs. Определение

Тема 7. Многофайловые программы

Добавление файлов к проекту в Visual Studio 2017. Добавление файлов к проекту в Code::Blocks. Добавление файлов к проекту в GCC/G++. Пример многофайловой программы.

Тема 8. Заголовочные файлы. Директивы препроцессора

Заголовочные файлы из стандартных библиотек C++. Пишем свои собственные заголовочные файлы. Угловые скобки (`<>`) vs. Двойные кавычки (`<<>>`). Почему `iostream` пишется без окончания `.h`? Можно ли записывать определения в заголовочных файлах? Директива `#include`. Директива `#define`. Макросы-объекты с `substitution_text`. Макросы-объекты без `substitution_text`. Условная компиляция. Область видимости директивы `#define`. Советы.

Тема 9. Header guards. Конфликт имён и `std namespace`

Проблема дублирования определений. Header guards. `#pragma once`. Оператор разрешения области видимости. `std namespace`. Пример конфликта имён.

Тема 10. Разработка ваших первых программ

Определите проблему. Определите свой инструментарий, цели и план бэкапа. Деление проблемы на части. Определение последовательности событий. Определение данных ввода/вывода на каждом этапе. Детали. Соединение и перемещение данных ввода/вывода в программе.

Тема 11. Отладка программ: степпинг и точки останова. Отладка программ: стек вызовов и отслеживание переменных

Типы ошибок. Отладчик. Степпинг. Команда «Шаг с заходом». Команда «Шаг с обходом». Команда «Шаг с выходом». Команда «Выполнить до текущей позиции». Команда «Продолжить». Точки останова. Отслеживание переменных. Окно просмотра. Стек вызовов.

Тема 12. Итоговый тест

Раздел 2. Переменные и основные типы данных в C++ (14 часов)

Тема 13 Размер типов данных и оператор sizeof Тип данных void

Адреса и переменные. Фундаментальные типы данных в C++. Определение переменных. Инициализация переменных. uniform инициализация. Присваивание переменных. Определение нескольких переменных. Где определять переменные? Использование Типа данных void.

Тема 14. Целочисленные типы данных Фиксированный размер целочисленных типов

Целочисленные типы данных. Определение целочисленных переменных. Диапазоны значений и знак целочисленных типов данных. Что используется по умолчанию: signed или unsigned? Переполнение. Примеры переполнения. Деление целочисленных переменных.

Тема 15. Типы данных с плавающей точкой

Типы данных с плавающей точкой. Экспоненциальная запись. Конвертация чисел в экспоненциальную запись. Точность и диапазон типов с плавающей точкой. Ошибки округления. nan и inf.

Тема 16. Логический тип данных

Переменные логического типа данных. Использование логического типа данных в ветвлениях if. Возвращаемые значения логического типа данных.

Тема 17. Символьный тип данных

Тип данных char. Вывод символов. Оператор static_cast. Ввод символов. Размер, диапазон и знак типа char. Управляющие символы. Что использовать: '\n' или std::endl? Другие символьные типы: wchar_t, char16_t и char32_t. В чём разница между одинарными и двойными кавычками при использовании с символами?

Тема 18. Литералы и магические числа. const, constexpr и символьные константы

Литеральные константы. Литералы в восьмеричной и шестнадцатеричной системах счисления. Бинарные литералы и разделитель цифр в C++14. Магические числа. Что с ними не так? Константы. Время компиляции и время выполнения. Спецификатор constexpr. Имена констант. Символьные константы. Использование символьных констант в программе.

Тема 19. Итоговый тест

Раздел 4. Операторы в C++ (12 часов)

Тема 20 Приоритет операций и правила ассоциативности

Правила ассоциативности. Таблица приоритета и ассоциативности операций

Тема 21. Арифметические операторы

Унарные арифметические операторы. Бинарные арифметические операторы. Деление целых чисел и чисел типа с плавающей точкой. Использование static_cast в операциях деления. Деление с остатком. Отрицательные числа в операциях деления до C++11. Арифметические операторы присваивания. Где оператор возведения в степень

Тема 22. Инкремент, декремент и побочные эффекты. sizeof, запятая и условный тернарный оператор

Инкремент, декремент, префиксы и побочные эффекты. Оператор sizeof. Оператор запятая. Условный тернарный оператор

Тема 23. Операторы сравнения

6 операторов сравнения. Сравнение чисел с плавающей точкой. Поиск алгоритмов для нужной ситуации.

Тема 24. Логические операторы: И, ИЛИ, НЕ

Логический оператор НЕ. Логический оператор ИЛИ. Логический оператор И. Короткий цикл вычислений. Использование операторов И/ИЛИ. Законы Де Моргана.

Тема 25. Конвертация чисел из двоичной системы в десятичную и наоборот

Представление чисел в двоичной системе. Конвертация чисел из двоичной системы в десятичную. Способ №1: Конвертация чисел из десятичной системы в двоичную. Способ №2: Конвертация чисел из десятичной системы в двоичную. Сложение двоичных чисел. Числа signed и метод «two's complement». Почему важен тип данных?

Тема 26. Итоговый тест

Раздел 4. Циклы в C++

Тема 27. Операторы управления потоком выполнения программ

Знакомство с операторами.

Тема 28. Операторы условного ветвления if/else

Знакомство с операторам if/else, условия использования.

Тема 29. Оператор switch

Знакомство с операторам switch, условия использования, эффективное использование.

Тема 30. Оператор goto

Знакомство с операторам goto, условия использования, эффективное использование.

Тема 31. Цикл while

Знакомство с циклом while, условия использования, эффективное использование.

Тема 32. Цикл do while

Знакомство с циклом do while, условия использования, эффективное использование.

Тема 33. Цикл for

Знакомство с циклом for, условия использования, эффективное использование.

Тема 34. Операторы break и continue

Знакомство с оператором break и continue, условия использования, эффективное использование.

Тема 35. Итоговый тест.

5. Организационно-педагогические условия реализации образовательной программы

Формы обучения – очная.

Наполняемость групп– до 10 человек в МБУ ДО «Волосовский ЦИТ».

Режим занятий:– 1 раз в неделю, два академических часа (40 минут), с перерывом в 10 минут.

Общее количество часов по программе: 68 часов.

Формы организации образовательной деятельности обучающихся: групповая, подгрупповая, индивидуальная, индивидуально-групповые, дистанционные, используемые технологии обучения (лекционные, блочно-модульные, дистанционные).

Организация аудиторных, внеаудиторных (самостоятельных) занятий, определение формы аудиторных занятий - учебное занятие, игра, фестиваль, дискуссия, семинар, проектная работа, исследовательская работа.

Форма контроля: практические работы; самостоятельные работы; опросы; контрольные работы.

Особенности организации образовательного процесса – осуществляется в соответствии с учебным планом в сформированных разновозрастных группах, постоянного состава.

Занятия проводятся полным составом объединения, но в зависимости от задания предполагает работу в паре или группе, а также индивидуальные занятия при подготовке к конкурсу.

6. Средства обучения

Материально-техническое обеспечение:

- Компьютерный класс (компьютерная техника последнего поколения (системный блок, монитор, клавиатура, мышка), звуковые колонки, наушники, принтер, сканер, видеокамера, микрофон, мультимедиа проектор, экран, наличие доступа в интернет, кабинет, оборудованный согласно правилам пожарной безопасности), столы и кресла с регулируемой высотой.

Средства обучения:

- комплект учебно-наглядных пособий;
- доступ в интернет;
- видеоуроки и презентации по тематическому планированию;
- комплект учебно-методической документации по дисциплине;
- перечень рекомендуемых учебных изданий, Интернет-ресурсов, дополнительной литературы.

7. Планируемые результаты освоения образовательной программы

В результате освоения курса обучающийся должен:

знать:

- этапы решения задачи на компьютере;
- типы данных;
- базовые конструкции изучаемого языка C++;
- принципы структурного и модульного программирования средствами языка C++;
- концепции объектно-ориентированного программирования средствами языка C++;

уметь:

- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- работать с файловой системой операционных систем;
- осуществлять отладку и тестирование программы;
- использовать ресурсы языка для программирования графического интерфейса пользователя;
- использовать приобретенные знания и умения в практической деятельности и повседневной жизни для:
- эффективной организации индивидуального информационного пространства;

- автоматизации коммуникационной деятельности;
- эффективного применения информационных ресурсов в учебной деятельности.

8. Система оценки результатов освоения общеразвивающей программы

Контроль и оценка результатов освоения курса осуществляется преподавателем в процессе проведения практических занятий, тестирования, а также выполнения обучающимися индивидуальных заданий в процессе текущего контроля и промежуточной аттестации.

Текущий контроль

Текущий контроль успеваемости обучающихся в центре осуществляется педагогом дополнительного образования по каждой изученной теме (разделу). Текущий контроль может проводиться в следующих формах: опрос, диктант, тестирование, реферат, контрольная работа, контрольное соревнование, конкурс творческих работ, защита творческих проектов, зачет, нетрадиционные формы контроля (игры, викторины, кроссворды), игра, конкурс.

Промежуточная аттестация

Основными формами проведения промежуточной аттестации обучающихся являются: тестирование, опрос, диктант, реферат, собеседование, наблюдение, контрольная работа, защита творческого проекта, контрольное соревнование, викторина, зачет, выставка, творческий отчет. Педагог выбирает форму промежуточной аттестации самостоятельно с учетом содержания реализуемой дополнительной общеразвивающей программы и документов, регламентирующих промежуточную аттестацию.

Итоговая аттестация

Основными формами проведения итоговой аттестации обучающихся являются: тестирование, диктант, защита творческого проекта, экзамен, творческий отчет.

Оценка достижения планируемых результатов

Критерии оценки результатов текущего контроля, промежуточной и итоговой аттестации:

1) Критерии оценки теоретической подготовки обучающихся:

- соответствие теоретических знаний программным требованиям;
- осмысленность и свобода владения специальной терминологией.

2) Критерии оценки практической подготовки обучающихся:

- соответствие уровня практических умений и навыков программным требованиям;
- свобода владения специальным инструментом, оборудованием и оснащением;
- качество выполнения практического задания.

Итоговая работа проводится педагогом в форме контрольной работы. Система оценивания – безотметочная (зачет/незачет). Используется только словесная оценка достижений обучающихся.

9. Список литературы

1. Чернышов М.К. Учебно-методическое пособие: Основы языка программирования С++ с применением технологии объектно-ориентированного программирования. Воронежский ГУ, 2007 год.
 2. Мамонова Т.Е. Информатика. Программирование на С++. Томский политехнический университет, 2011г.
 3. Зырянов К.И., Кисленко Н.П. Программирование на С++. Новосибирск, 2017.
 6. Основы программирования /С. М. Окулов. – 2- е издю, испр. – М.:БИНОМ. Лаборатория знаний, 2005. – 440 с.: ил. ISBN 5-94774-217-9
 7. Программирование в алгоритмах /С. М. Окулов. – 2- е издю, испр. – М.:БИНОМ. Лаборатория знаний, 2004. – 341 с.: ил. ISBN 5-94774-010 -9.
 8. Задачи по программированию / С.М. Окулов, Т.В. Ашихмина, Н.А. Бушмелева и др.; Под. Ред. С.М. Окулова. – М.: БИНОМ. Лаборатория знаний, 2006. – 820 с.; ил.
 9. Фундаментальные алгоритмы на С++. Анализ/Структуры данных/Сортировка/Поиск: Пер. с англ./Роберт Седжвик. – СПб.: ООО «ДиаСофтЮП», 2002. – 688 с.
 - 10.Фундаментальные алгоритмы на С++. Алгоритмы на графах: Пер. с англ./Роберт Седжвик. – СПб.: ООО «ДиаСофтЮП», 2002. – 496 с.
- Интернет ресурсы:
- Уроки С++: <https://ravesli.com/uroki-cpp/>;
- Основы программирования на языках Си и С++ для начинающих <http://cppstudio.com>;
- С++ с нуля: <https://code-live.ru/tag/cpp-manual/>

ПРИЛОЖЕНИЕ 1

Календарный учебный график к дополнительной общеразвивающей программе «Программирование на языке С++» на 2020-2021 учебный год

Письмо Министерства образования и науки Российской Федерации от 18.11.2015 № 09-3242 «О направлении информации» «Методические рекомендации по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы)»

Общеразвивающая программа	«Программирование на языке С++» технической направленности	
Сроки освоения	1 год	
Начало учебного года	С <u>1</u> сентября 2020 года	
Учебные периоды	<u>1-ый период (1-ое полугодие)</u> с <u>1</u> сентября по <u>30</u> декабря 2020 года <u>2-ой период (2-ое полугодие)</u> С <u>11</u> января по <u>29</u> мая 2021 года	
Продолжительность учебного года	34 недели	
Количество учебных недель по полугодиям	первое учебное полугодие	второе учебное полугодие
	32	36
Продолжительность учебной недели. Комплектование групп	Продолжительность учебной недели – 5 дней. Занятия проводятся по группам. Групповые – <u>10</u> человек (а)	
Режим занятий	Четверг 15.50- 16.30 ; 16.40- 17.20 Продолжительность занятий – 40 минут (академический час), с перерывом в 10 минут	
Учебная нагрузка в неделю	1 год обучения (2 час в неделю) 1 раз в неделю по 2 часа	
Праздничные дни	Согласно календарю праздничных дней, утвержденному Министерством труда и социального развития РФ праздничные дни В первом полугодии: <u>1- 10 января 2021; 23 февраля – 2021; 08 марта – 2021; 1-3 и 9-10 мая 2021 года;</u> <u>во втором полугодии: 04 ноября 2020 года.</u>	
Промежуточная аттестация	С 20 апреля по 20 мая (согласно Положению об аттестации обучающихся)	
Окончание учебного года	29 мая 2021 года	
Каникулы в учреждении	<u>Зимние</u> – продолжительность каникул определяется количеством праздничных дней, согласно календарю праздничных дней, утвержденному Министерством труда и социального развития РФ. <u>Весенние</u> – 23.03-29.03 <u>Летние</u> – с 01 июня по 31 августа 2021 года. В летнее время – реализация досуговых образовательных программ. Учебно-тренировочные сборы по графику Осенние – 29.10-4.11.	
Учеба в период каникул	В период <u>школьных каникул</u> занятия проводятся в соответствии с учебным планом.	

Перечень заданий для подготовки к итоговой аттестации

1. Калькулятор для ипотеки
2. Обмен-возврат
3. Валидатор кредитной карты
4. Гороскоп
5. Гостевая книга / журнал
6. Шифр Винежера / Вернама / Цезаря
7. Монитор трафика
8. Сканер портов
9. Галерея изображений
10. Менеджер рецептов
11. Площадь и периметр фигуры
12. Цветочный магазин с возможностью заказа

Вопросы для самопроверки

1. Какое самое длинное имя можно использовать в C++ в Вашей системе?
2. Определите функции `f(char)`, `g( char& )`, `h( const char & )`. Вызовите их с параметрами `'a'`, `49.3`, `c` (`c` - типа `char`), `uc` (типа `unsigned char`), `sc` (типа `signed char`). Какие вызовы допустимы? Объясните механизм вызовов.
3. Что происходит в Вашей системе, если в программе выполняется деление на нуль? Что происходит при переполнении?
4. Как использовать модули, написанные на C и C++, в одной программе?
5. В чем разница между оператором `new` и функцией `malloc()`?
6. Попробуйте определить `inline`-функцию с циклом. Как реагирует компилятор на такую конструкцию?
7. Можно ли перегрузить функции следующего вида:
`void func( int );`
`void func( int & );`
`void func( int * ); ?`
8. Что такое анонимное объединение? Приведите пример использования.

Вопросы для самопроверки

1. Объясните назначение конструкторов и деструкторов.
2. Могут ли члены-данные иметь атрибут `const`? Как инициализировать эти члены?
3. Можно ли создавать объекты массивов? Какие конструкторы вызываются при их создании? Как вызываются деструкторы?
4. Как создать динамический массив объектов? Как освободить память при уничтожении динамического массива объектов?

5. Пусть класс имеет вид:

```
class Vector
{
    int * data;
    int size;
public:
    Vector(int)
    { data = new T [s];
      size = s;
    }
    ~Vector() { delete data; }
};
```

Что будет происходить при выполнении следующих действий?

```
void main()
{
    Vector a(50);
    Vector * pa = &a;
    Vector b = a;
}
```

Как избавиться от ошибок в программе?