

УПРАВЛЕНИЕ ОБРАЗОВАНИЯ АДМИНИСТРАЦИИ ГОРОДА
НОВОЧЕРКАССКА
МУНИЦИПАЛЬНОЕ АВТОНОМНОЕ УЧРЕЖДЕНИЕ
ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
«ЦЕНТР ВОСПИТАНИЯ И ДОСУГА «ЭСТЕТИКА»

СОГЛАСОВАНО

на заседании педагогического совета
Протокол
от 26.08.2025 г. №1

УТВЕРЖДАЮ

Директор
МАУ ДО «ЦВД Эстетика»
Приказ № 336-о/д от 26.08.2025 г.
 Цивилёва И.В.



ДОПОЛНИТЕЛЬНАЯ ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА
«ОСНОВЫ ПРОГРАММИРОВАНИЯ НА C++»

Уровень программы: продвинутый
Вид программы: модульная
Уровень программы: разноуровневая
Возраст детей: от 14 до 18 лет
Срокреализации: 1 год – 144 часа
Разработчик: педагог дополнительного
образования Мороз Владислав Олегович

г. Новочеркасск 2025 г.

ОГЛАВЛЕНИЕ

I. КОМПЛЕКС ОСНОВНЫХ ХАРАКТЕРИСТИК ОБРАЗОВАНИЯ.....	3
1.1. Пояснительная записка.....	5
1.2. Календарно-тематическое планирование (144 часа)	7
II. КОМПЛЕКС ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИХ УСЛОВИЙ	17
3.1. Материально-техническое обеспечение	17
3.2. Информационно-методическое обеспечение.....	17
3.3. Система оценки результатов обучения.....	17
3.4. Критерии оценки.....	18
3.5. Наполняемость групп	18
III. ДИАГНОСТИЧЕСКИЙ ИНСТРУМЕНТАРИЙ	19
4.1. Стартовая диагностика	19
4.2. Текущая диагностика.....	19
4.3. Промежуточная диагностика	19
4.4. Чек-листы для практических работ.....	19
4.5. Карты наблюдения за практической работой	20
4.6. Критерии оценки мини-проектов	20
4.7. Карта оценки итогового проекта	20
КАЛЕНДАРНЫЙ ПЛАН ВОСПИТАТЕЛЬНОЙ РАБОТЫ.....	25
VI. СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ.....	27

I. КОМПЛЕКС ОСНОВНЫХ ХАРАКТЕРИСТИК ОБРАЗОВАНИЯ

Нормативно-правовая база

1. Конституция Российской Федерации (принята всенародным голосованием 12.12.1993 с изменениями, одобренными в ходе общероссийского голосования 01.07.2020).

2. Федеральный закон Российской Федерации от 24.07.1998 № 124-ФЗ «Об основных гарантиях прав ребенка в Российской Федерации».

3. Федеральный закон от 29.12.2012 №273-ФЗ «Об образовании в Российской Федерации».

4. Федеральный закон от 13.07.2020 № 189-ФЗ «О государственном (муниципальном) социальном заказе на оказание государственных (муниципальных) услуг в социальной сфере».

5. Федеральный проект «Успех каждого ребенка», утвержденный протоколом заседания проектного комитета по национальному проекту «Образование» от 07.12.2018 № 3.

6. Федеральный закон от 19 декабря 2023 г. №618-ФЗ «О внесении изменений в Федеральный закон «Об образовании в Российской Федерации».

7. Федеральный государственный образовательный стандарт основного общего образования, утвержденный приказом Минпросвещения России от 31 мая 2021 г. № 287 «Об утверждении федерального государственного образовательного стандарта основного общего образования» (далее - ФГОС ООО).

8. Федеральный государственный образовательный стандарт среднего общего образования, утвержденный приказом Минобрнауки России от 17 мая 2012 г. № 413 «Об утверждении федерального государственного образовательного стандарта среднего общего образования» в редакции от 27 декабря 2023 г. (далее с ФГОССОО).

9. Федеральная образовательная программа основного общего образования, утвержденная приказом Минпросвещения России от 18 мая 2023 г. № 370 «Об утверждении федеральной образовательной программы основного общего образования» (далее - ФОП ООО).

10. Федеральная образовательная программа среднего общего образования, утвержденная приказом Минпросвещения России от 18 мая 2023 г. № 371 «Об утверждении федеральной образовательной программы начального общего образования» (далее - ФОПСОО).

11. Приказ Минпросвещения России от 1 февраля 2024 г. № 62 «О внесении изменений в некоторые приказы Министерства просвещения Российской Федерации, касающиеся федеральных образовательных программ основного общего и среднего общего образования».

12. Приоритетный проект «Доступное дополнительное образование для детей», утвержденный протоколом заседания президиума Совета при Президенте Российской Федерации от 30.11.2016 № 11.

13. Распоряжение Правительства Российской Федерации от 29.05.2015 № 996-р «Стратегия развития воспитания в Российской Федерации на период до 2025 года».

14. Распоряжение Правительства Российской Федерации от 31.03.2022 № 678-р «Концепция развития дополнительного образования детей до 2030 года».

15. Приказ Министерства просвещения Российской Федерации от 03.09.2019 № 467 «Об утверждении Целевой модели развития региональных систем дополнительного образования детей».

16. Приказ Министерства образования и науки Российской Федерации и Министерства просвещения Российской Федерации от 05.08.2020 № 882/391 «Об организации и осуществлении образовательной деятельности при сетевой форме реализации образовательных программ».

17. Приказ Министерства труда и социального развития Российской Федерации от 22.09.2021 № 652н «Об утверждении профессионального стандарта «Педагог дополнительного образования детей и взрослых».

18. Приказ Министерства просвещения Российской Федерации от 27.07.2022 № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам».

19. Письмо Министерства образования и науки Российской Федерации от 28.08.2015 № АК-2563/05 «О методических рекомендациях» направлении информации» (вместе с «Методическими рекомендациями по организации образовательной деятельности с использованием сетевых форм реализации образовательных программ»).

20. Письмо Министерства образования и науки Российской Федерации от 18.11.2015 № 09-3242 «О направлении информации» (вместе с «Методическими рекомендациями по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы)»).

21. Письмо Министерства образования и науки Российской Федерации от 29.03.2016 № ВК-641/09 «О направлении методических рекомендаций» (вместе с Методическими рекомендациями по реализации адаптированных дополнительных общеобразовательных программ, способствующих социально-психологической реабилитации, профессиональному самоопределению детей с ограниченными возможностями здоровья, включая детей-инвалидов, с учетом их особых образовательных потребностей).

22. Письмо Министерства просвещения Российской Федерации от 30.12.2022 № АБ-3924/06 «Создание современного инклюзивного образовательного пространства для детей с ограниченными возможностями здоровья и детей-инвалидов на базе образовательных организаций, реализующих дополнительные общеобразовательные программы в субъектах Российской Федерации».

23. Письмо Министерства просвещения Российской Федерации от 29.09.2023

№АБ-3935/06«Ометодическихрекомендациях»(вместе с «Методическими рекомендациями по формированию механизмов обновления содержания, методов и технологий обучения в системе дополнительного образования детей, направленных на повышение качества дополнительного образования детей, в том числе включение компонентов, обеспечивающих формирование функциональной грамотности и компетентностей, связанных с эмоциональным, физическим, интеллектуальным, духовным развитием человека, значимых для вхождения Российской Федерации в число десяти ведущих стран мира по качеству общего образования, для реализации приоритетных направлений научно- технологического и культурного развития страны»).

24. Постановление Главного государственного санитарного врача Российской Федерации от 28.09.2020 № 28 «Об утверждении санитарных правил СП2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи».

25. Постановление Главного государственного санитарного врача Российской Федерации от 28.01.2021 № 2 «Об утверждении санитарных правил и норм СанПиН 1.2.368521 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания».

26. ПостановлениеПравительстваРостовскойобластииот08.12.2020 № 289 «О мероприятиях по формированию современных управленческих решений и организационно-экономических механизмов в системе дополнительного образования детей в Ростовской области в рамках федерального проекта «Успех каждого ребёнка» национального проекта «Образование».

27. Приказ министерства общего и профессионального образования Ростовской области от 01.08.2023 № 718 «О проведении независимой оценки качества дополнительных общеразвивающих программ в Ростовской области».

28. Приказ министерства общего и профессионального образования Ростовскойобластииот03.08.2023№724«Об утверждении Требований к условиям и порядку оказания государственной услуги в социальной сфере «Реализация дополнительных общеразвивающих программ» в Ростовской области.

1.1. Пояснительная записка

Направленность: Техническая.

Актуальность: С++ – один из фундаментальных языков программирования, лежащий в основе операционных систем, игровых движков, высоконагруженных приложений и системного программного обеспечения. Изучение С++ дает глубокое понимание работы компьютера, управления памятью и объектно-ориентированного программирования (ООП), что является неотъемлемой частью подготовки будущих инженеров-программистов, разработчиков игр и специалистов в области ComputerScience.

Новизна: Программа сочетает классический подход к изучению синтаксиса C++ с современными практиками разработки и акцентом на решении практических задач и создании собственных проектов.

Педагогическая идея: От простого к сложному – от написания первой программы до разработки полноценного консольного приложения с использованием основных парадигм программирования.

Цель: Формирование у обучающихся систематизированных знаний и практических умений в области программирования на языке C++.

Задачи:

Обучающие:

- Сформировать представление о базовых конструкциях языка C++ (переменные, типы данных, операторы, условия, циклы).
- Обучить работе с функциями, массивами, структурами и указателями.
- Раскрыть основные принципы объектно-ориентированного программирования (инкапсуляция, наследование, полиморфизм).
- Научить основам работы с файлами и стандартной библиотекой шаблонов (STL).

Развивающие:

- Развивать алгоритмическое и логическое мышление.
- Развивать навыки поиска и исправления ошибок (отладки).
- Развивать умение самостоятельно осваивать новый материал и читать техническую документацию.

Воспитательные:

- Воспитывать внимательность, аккуратность и структурированность при написании кода.
- Формировать культуру работы в профессиональной среде (соблюдение код-стайла, комментирование кода).

Возраст обучающихся: 14-18 лет (9-11 классы). Программа требует наличия базовых знаний математики и логики.

Срок реализации: 144 академических часа (например, 2 раза в неделю по 2 часа в течение 36 учебных недель).

Формы и режим занятий: Групповая (до 12-15 человек). Теоретические занятия (лекции, демонстрации) и практические (лабораторные работы, проектная деятельность).

1.2. Календарно-тематическое планирование (144 часа)

Таблица 1

№ модуля	Наименование модуля и тем	Всего часов	Теория	Практика	Форма контроля
Модуль 1	Введение в С++ и базовые конструкции	30	10	20	Тест, практическая работа
1.1.	Введение в среду разработки (IDE). Структура программы. Первая программа "Hello, World!". Компиляция и запуск.	4	1	3	
1.2.	Переменные и фундаментальные типы данных (int, float, double, char, bool).	4	1	3	
1.3.	Операторы: арифметические, присваивания, инкремента/декремента.	4	1	3	
1.4.	Ввод и вывод данных (cout, cin).	2	1	1	
1.5.	Условные операторы: if, else, elseif. Оператор выбора switch.	6	2	4	
1.6.	Циклы: while, do-while, for. Операторы break и continue.	6	2	4	
1.7.	Контрольное занятие по Модулю 1.	4	2	2	
Модуль 2	Функции, массивы и указатели	36	12	24	Тест, мини-проект
2.1.	Функции: объявление, определение, вызов. Передача параметров по значению.	6	2	4	

№ модуля	Наименование модуля и тем	Всего часов	Теория	Практика	Форма контроля
2.2.	Пространства имен (namespace).	2	1	1	
2.3.	Массивы: одномерные и многомерные. Работа с индексами.	6	2	4	
2.4.	Указатели и ссылки. Адресная арифметика.	8	3	5	
2.5.	Строки в стиле C (массивы char). Функции для работы со строками (<cstring>).	6	2	4	
2.6.	Контрольное занятие по Модулю 2: Разработка консольной игры "Угадай число".	8	2	6	
Модуль 3	Структуры и основы ООП	32	12	20	Защита проекта
3.1.	Структуры (struct). Создание собственных типов данных.	4	1	3	
3.2.	Введение в ООП. Классы и объекты. Инкапсуляция.	6	2	4	
3.3.	Спецификаторы доступа (public, private). Методы классов.	6	2	4	
3.4.	Конструкторы и деструкторы.	6	2	4	
3.5.	Перегрузка функций и операторов.	6	3	3	
3.6.	Контрольное занятие по Модулю 3: Разработка класса "Студент" или "Книга".	4	2	2	
Модуль 4	Динамическая память и наследование	24	8	16	Тест, практическая работа
4.1.	Динамическое выделение	6	2	4	

№ модуля	Наименование модуля и тем	Всего часов	Теория	Практика	Форма контроля
	памяти (new, delete).				
4.2.	Наследование. Создание иерархий классов.	6	2	4	
4.3.	Полиморфизм. Виртуальные функции.	6	2	4	
4.4.	Контрольное занятие по Модулю 4.	6	2	4	
Модуль 5	Стандартная библиотека шаблонов (STL) и файлы	16	6	10	Защита проекта
5.1.	Контейнеры STL: vector, string, map.	6	2	4	
5.2.	Работа с файлами: чтение и запись (<fstream>).	6	2	4	
5.3.	Контрольное занятие по Модулю 5: Разработка программы для учета данных (например, список дел).	4	2	2	
Модуль 6	Итоговый проект	6	2	4	Защита итогового проекта
6.1.	Постановка задачи и проектирование архитектуры.	2	1	1	
6.2.	Разработка и отладка проекта.	2	0	2	
6.3.	Презентация и защита итоговых проектов.	2	1	1	
Итого:		144	50	94	

Содержание программы

Модуль 1: Знакомство со средой разработки (например, VisualStudioCode, Code::Blocks или CLion). Изучение базового синтаксиса, позволяющего писать

простые линейные и разветвленные алгоритмы. Основное внимание – отработке навыков компиляции и отладки.

Модуль 2:Изучение принципов структурного программирования. Разбиение программы на логические блоки (функции). Работа с структурами данных – массивами и указателями, что является ключевой особенностью и сложностью C++.

Модуль 3:Переход к объектно-ориентированной парадигме. Создание собственных типов данных (классов), инкапсуляция данных и методов для работы с ними.

Модуль 4:Углубленное изучение ООП: механизмы наследования и полиморфизма для создания гибких и расширяемых программ. Освоение работы с динамической памятью.

Модуль 5:Знакомство с мощнейшим инструментом C++ – STL, который значительно упрощает работу с коллекциями данных и строками. Освоение базовых операций ввода-вывода с файлами.

Модуль 6:Самостоятельная или командная (2-3 человека) работа над проектом, объединяющим все изученные темы (например, "Консольный файловый менеджер", "Простая текстовая игра", "База данных сотрудников").

Модуль 1: Введение в C++ и базовые конструкции (30 часов)

Занятие 1.1. Введение в среду разработки (IDE). Первая программа (4 часа)

Цель: Познакомить со средой разработки, создать и запустить первую программу.

Теория (1 ч):

- Что такое компилятор и IDE (на примере VisualStudioCode/Code::Blocks).
 - Структура простейшей программы на C++:

```
#include<iostream>,usingnamespacestd;,intmain() { }.
```
- Понятие функции `main()` как точки входа.

Практика (3 ч):

- Установка и настройка IDE (если требуется).
- Создание проекта "HelloWorld".
- Написание, компиляция и запуск первой программы.
- **Задание:** Модифицировать программу для вывода разных фраз (имя, приветствие).

Занятие 1.2. Переменные и фундаментальные типы данных (4 часа)

Цель: Научить объявлять переменные и работать с базовыми типами данных.

Теория (1 ч):

- Понятие переменной как "ящика" для данных.
- Основные типы: `int`(целые), `float/double`(дробные), `char`(символы), `bool`(логический).
- Правила именования переменных.

Практика (3 ч):

Задачи: Объявить переменные разных типов, присвоить им значения, вывести на экран.

Пример: Программа для хранения и вывода возраста, роста, оценки и инициалов.

Занятие 1.3. Операторы (4 часа)

Цель: Освоить основные операторы языка.

Теория (1 ч):

- Арифметические операторы: `+`, `-`, `*`, `/`, `%`.
- Операторы присваивания: `=`, `+=`, `-=`.
- Инкремент/декремент: `++`, `--`.

Практика (3 ч):

Задачи: Написать программу-калькулятор для двух чисел.

Пример: Вычисление площади прямоугольника, среднего балла.

Занятие 1.4. Ввод и вывод данных (2 часа)

Цель: Научить программу взаимодействовать с пользователем.

Теория (1 ч):

- Поток вывод `std::cout` форматирование.
- Поток ввода `std::cin`.

Практика (1 ч):

Задание: Создать программу, которая запрашивает имя и год рождения, а затем вычисляет и выводит возраст.

Занятие 1.5. Условные операторы (6 часов)

Цель: Научить программу принимать решения.

Теория (2 ч):

Операторы сравнения: `==`, `!=`, `>`, `<`, `>=`, `<=`.

- Логические операторы: `&&`, `|`, `!`.
- Конструкции `if`, `elseif`, `else`.
- Оператор выбора `switch`.

Практика (4 ч):

Задачи:

1. Проверка числа на четность.
2. Калькулятор с выбором операции (`+`, `-`, `*`, `/`).
3. Определение времени года по номеру месяца (`switch`).

Занятие 1.6. Циклы (6 часов)

Цель: Освоить механизмы многократного выполнения кода.

Теория (2 ч):

- Цикл `while`: выполнение пока условие истинно.
- Цикл `do-while`: выполнение хотя бы один раз.
- Цикл `for`: итерация с счетчиком.
- Операторы `break` (выход) и `continue` (переход к следующей итерации).

Практика (4 ч):

Задачи:

1. Вывести числа от 1 до N.
2. Найти сумму всех четных чисел в диапазоне.
3. Реализовать программу "Загадай число", где компьютер угадывает число пользователя.

Занятие 1.7. Контрольное занятие по Модулю 1 (4 часа)

Теория (2 ч): Письменный тест на знание синтаксиса, типов данных, операторов.

Практика (2 ч): Самостоятельная работа: Написать программу, которая решает квадратное уравнение (с проверкой дискриминанта) или реализует простой шифратор текста.

Модуль 2: Функции, массивы и указатели (36 часов)

Занятие 2.1. Функции (6 часов)

Цель: Научить разбивать программу на логические блоки.

Теория (2 ч):

- Понятие функции. Прототип, определение, вызов.
- Возвращаемое значение (`return`).
- Передача параметров по значению.

Практика (4 ч):

Задачи:

1. Создать функции для основных математических операций.
2. Написать функцию, проверяющую, является ли число простым.
3. Функция для поиска максимального из трех чисел.

Занятие 2.2. Пространства имен (2 часа)

Цель: Понять и уметь использовать пространства имен.

Теория (1 ч): Зачем нужны `namespace`, `std::`, директива `using`.

Практика (1 ч): Создание своего пространства имен для функций из предыдущего занятия.

Занятие 2.3. Массивы (6 часов)

Цель: Освоить работу с наборами данных.

Теория (2 ч):

- Объявление и инициализация одномерных массивов.
- Доступ к элементам по индексу.
- Многомерные массивы (матрицы).

Практика (4 ч):

Задачи:

1. Найти минимальный/максимальный элемент в массиве.
2. Отсортировать массив "пузырьком".
3. Вычислить сумму элементов матрицы.

Занятие 2.4. Указатели и ссылки (8 часов)

Цель: Понять одну из ключевых особенностей C++ – работу с памятью.

Теория (3 ч):

- Что такое адрес памяти. Оператор взятия адреса &.
- Указатели: объявление, разыменование*.
- Ссылки как "алиасы" для переменных.
- Адресная арифметика (для продвинутых групп).

Практика (5 ч):

Задачи:

1. Написать функцию swap, меняющую значения двух переменных с помощью указателей.
2. Использовать указатели для обхода массива.
3. Создать массив указателей на функции.

Занятие 2.5. Строки в стиле C (6 часов)

Цель: Научить работать со строками как с массивами символов.

Теория (2 ч):

- Массив char[] и нуль-терминатор \0.
- Библиотека <cstring>: strlen(), strcpy(), strcat(), strcmp().

Практика (4 ч):

Задачи:

1. Реализовать аналог функций strlen, strcpy самостоятельно.
2. Написать программу, проверяющую, является ли строка палиндромом.
3. Простой шифратор Цезаря для строки.

Занятие 2.6. Контрольное занятие: Мини-проект "Угадай число" (8 часов)

Теория (2 ч): Проектирование архитектуры программы: основные функции (генерация числа, проверка, интерфейс).

Практика (6 ч): Разработка игры:

- Компьютер загадывает число.
- Пользователь вводит догадки.
- Программа выдает подсказки ("больше", "меньше").
- Учет количества попыток.
- *(Для сильных групп): Режим наоборот (компьютер отгадывает).*

Модуль 3: Структуры и основы ООП (32 часа)

Занятие 3.1. Структуры (4 часа)

Цель: Научиться создавать собственные сложные типы данных.

Теория (1 ч): Ключевое слово `struct`, объявление полей, создание переменных-структур.

Практика (3 ч):

Задание: Создать структуры `Student` (поля: имя, возраст, средний балл) и `Book` (название, автор, год). Написать функции для их вывода на экран.

Занятие 3.2. Введение в ООП. Классы и объекты (6 часов)

Цель: Понять базовые принципы Объектно-Ориентированного Программирования.

Теория (2 ч):

- Класс как "чертеж" объекта. Объект как экземпляр класса.
- Понятие инкапсуляции – объединение данных и методов в одном классе.

Практика (4 ч): Преобразовать структуру `Student` в класс, добавить методы для установки и получения значений полей.

Занятие 3.3. Спецификаторы доступа и методы (6 часов)

Цель: Научиться управлять доступом к данным класса.

Теория (2 ч): `public` (открытый интерфейс), `private` (внутренняя реализация). Методы класса.

Практика (4 ч): Модифицировать класс `Student`: сделать поля `private`, создать `public` -методы (геттеры/сеттеры), добавить метод `printInfo()`.

Занятие 3.4. Конструкторы и деструкторы (6 часов)

Цель: Освоить механизмы инициализации и очистки объектов.

Теория (2 ч):

- Конструктор (по умолчанию, с параметрами, копирования). Список инициализации.
- Деструктор.

Практика (4 ч): Добавить в класс `Student` конструкторы и деструктор с выводом отладочных сообщений.

Занятие 3.5. Перегрузка функций и операторов (6 часов)

Цель: Понять принцип полиморфизма на этапе компиляции.

Теория (3 ч):

- Перегрузка функций (функции с одинаковыми именами, но разными параметрами).
- Перегрузка операторов (например, `==` для сравнения двух объектов `Student`).
- **Практика (3 ч):** Перегрузить операторы `==`, `<<` (для вывода в `cout`) для класса `Student`.

Занятие 3.6. Контрольное занятие: Защита проекта (4 часа)

- **Теория (2 ч):** Подготовка презентации своего класса.
- **Практика (2 ч):** **Защита проекта:** Каждый обучающийся представляет разработанный класс (например, "Банковский счет", "Прямоугольник"), объясняет его архитектуру и демонстрирует работу.

Модуль 4: Динамическая память и наследование (24 часа)

(Продолжение по аналогии: для каждого занятия расписываются цели, теория и практические задания)

Занятие 4.1. Динамическая память: Теория – операторы `new/delete`. Практика – создание динамических массивов, массивов объектов.

Занятие 4.2. Наследование: Теория – создание иерархий (`classStudent:publicPerson`). Практика – построение простой иерархии классов (например, транспортные средства).

Занятие 4.3. Полиморфизм: Теория – виртуальные функции, абстрактные классы. Практика – создание массива указателей на базовый класс, который хранит объекты разных производных классов.

Модуль 5: Стандартная библиотека шаблонов (STL) и файлы (16 часов)

Занятие 5.1. Контейнеры STL: Теория – `vector, string, map`. Практика – переписать одну из старых программ с использованием `vector` вместо массива.

Занятие 5.2. Работа с файлами: Теория – потоки `ifstream, ofstream`. Практика – сохранение и загрузка данных класса `Student` в текстовый файл.

Модуль 6: Итоговый проект (6 часов)

Занятие 6.1-6.3: Самостоятельная или командная работа над проектом по выбору ("Консольный органайзер", "Простая RPG-игра", "Система тестирования"). Защита перед группой с демонстрацией кода и функционала. Этот подробный план дает педагогу четкий пошаговый маршрут для проведения занятий, а вам как методисту – полное представление о содержании каждого учебного дня.

4. Планируемые результаты

Обучающийся будет знать: синтаксис и основные конструкции C++, принципы ООП, основы работы с памятью, основные компоненты STL.

Обучающийся будет уметь: писать, компилировать и отлаживать консольные приложения на C++, использовать функции, классы, указатели, работать с файлами и стандартными контейнерами.

Обучающийся будет владеть: навыками алгоритмизации, основами объектно-ориентированного проектирования, культурой написания читаемого кода.

5. Система оценки результатов

Текущий контроль: Выполнение лабораторных работ, устные опросы, тесты.

Промежуточный контроль: Защита мини-проектов после каждого модуля.

Итоговый контроль: Защита итогового проекта, который оценивается по критериям: функциональность, качество кода, сложность, оформление и презентация.

6. Методическое обеспечение

- Раздаточные материалы (памятки по синтаксису, алгоритмы).
- Примеры кода для каждого занятия.
- Сборник практических заданий и тем для проектов.
- Список рекомендованной литературы и онлайн-ресурсов (cplusplus.com, cppreference.com).

II. КОМПЛЕКС ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИХ УСЛОВИЙ

Условия реализации программы

3.1. Материально-техническое обеспечение

Оборудованный компьютерный класс:

- Стационарные компьютеры/ноутбуки (не менее 1 на 2 обучающихся)
- Операционная система: Windows 10/11 или Linux
- Процессор: не менее 2 ядер, 2 ГГц
- Оперативная память: не менее 4 ГБ
- Свободное дисковое пространство: не менее 10 ГБ на рабочем месте

Программное обеспечение:

- Среда разработки: VisualStudioCode с расширениями для C++ или Code::Blocks/CLion
- Компилятор: MinGW-w64/GCC или Microsoft Visual C++ Compiler
- Система контроля версий: Git (для старших групп)
- Офисный пакет для документации

Дополнительное оборудование:

- Проектор или интерактивная доска
- Принтер для распечатки методических материалов
- Бесперебойное интернет-соединение

Резервный источник электропитания

3.2. Информационно-методическое обеспечение

Учебно-методический комплекс:

- Рабочая программа с календарно-тематическим планированием
- Конспекты занятий и презентации
- Сборник практических заданий и лабораторных работ
- Банк контрольных вопросов и тестовых заданий
- Методические рекомендации по выполнению проектов

Цифровые образовательные ресурсы:

- Электронная библиотека с учебной литературой
- Видеоуроки по ключевым темам программы
- Онлайн-компиляторы для самостоятельной работы
- Образцы готовых проектов и шаблоны кода

Формы аттестации и оценочные материалы

3.3. Система оценки результатов обучения

Текущий контроль:

- Проверка практических заданий на каждом занятии
- Устный опрос по теоретическому материалу
- Экспресс-тесты для проверки усвоения ключевых понятий

Промежуточная аттестация:

- Контрольные работы по окончании каждого модуля
- Защита мини-проектов и лабораторных работ
- Тестирование по основным разделам программы

Итоговая аттестация:

- Защита итогового проекта
- Комплексный экзамен (теоретическая и практическая части)

3.4. Критерии оценки

Практические работы:

- Корректность выполнения задания - 40%
- Оптимальность и качество кода - 30%
- Соблюдение стандартов оформления - 20%
- Своевременность выполнения - 10%

Проектные работы:

- Функциональность и сложность проекта - 40%
- Архитектура и качество кода - 30%
- Оформление документации - 20%
- Качество презентации и защиты - 10%

Организационные условия

3.5. Наполняемость групп

- Рекомендуемая наполняемость: 10-12 человек
- Минимальное количество для запуска группы: 8 человек
- Максимальное количество: 15 человек

Мониторинг и оценка качества

- Регулярный сбор обратной связи от обучающихся и родителей
- Анализ успеваемости и динамики обучения
- Корректировка программы на основе результатов мониторинга
- Проведение итогового анкетирования удовлетворенности качеством образования

III. ДИАГНОСТИЧЕСКИЙ ИНСТРУМЕНТАРИЙ

Диагностика теоретических знаний

4.1. Стартовая диагностика

Форма: Входное анкетирование

Цель: Выявление исходного уровня подготовки

Содержание:

- Базовые понятия алгоритмизации
- Опыт работы с компьютерами
- Мотивация к обучению

Методика: Анонимное анкетирование, анализ результатов

4.2. Текущая диагностика

Формы:

- Экспресс-опросы в начале занятия (5-7 минут)
- Карточки-задания по терминологии
- Электронные тесты по пройденным темам

Пример тестового задания:

*"Что выведет программа?"

```
сpp
```

```
int x =5;
```

```
cout<<"+x + x++;
```

```
""
```

Критерии: Правильность ответов, скорость выполнения

4.3. Промежуточная диагностика

Форма: Контрольные работы по модулям

Содержание:

- Теоретические вопросы
- Задачи на анализ кода
- Практические задания

Пример задания: "Найти и объяснить ошибки в приведенном коде"

Диагностика практических умений

4.4. Чек-листы для практических работ

Структура:

- Корректность синтаксиса
- Логическая правильность алгоритма
- Оптимальность решения
- Соблюдение стандартов оформления кода

Чек-лист для задачи "Калькулятор":

Таблица 2

Критерий	Баллы	Примечания
Программа компилируется без ошибок	0-2	
Корректно работают все арифметические операции	0-3	
Обработка ошибочного ввода	0-2	
Читаемость и оформление кода	0-2	
Использование функций	0-1	
Итого	0-10	

4.5. Карты наблюдения за практической работой

Параметры оценки:

- Умение пользоваться IDE
- Навыки отладки программ
- Скорость решения типовых задач
- Умение находить и исправлять ошибки

Диагностика проектной деятельности

4.6. Критерии оценки мини-проектов

Функциональность (40%):

- Соответствие техническому заданию
- Полнота реализации функций
- Отсутствие критических ошибок

Качество кода (30%):

- Соблюдение codestyle
- Оптимальность алгоритмов
- Комментирование сложных участков

Сложность (20%):

- Использование изученных технологий
- Оригинальность решения

Оформление (10%):

- Наличие документации
- Читаемость вывода

4.7. Карта оценки итогового проекта

Таблица 3

Критерий	Вес	Оценка	Комментарии
Техническая реализация	50%		
- Корректность работы	15%		
- Качество и читаемость кода	15%		
- Сложность алгоритмов	10%		
- Использование ООП	10%		
Содержательная часть	30%		
- Соответствие заданию	10%		
- Полнота реализации	10%		
- Оригинальность идеи	10%		
Оформление и защита	20%		
- Качество документации	10%		
- Презентация проекта	10%		
ИТОГО	100%		

IV. ДИАГНОСТИКА ЛИЧНОСТНЫХ РЕЗУЛЬТАТОВ

5.1. Карта наблюдения за развитием softskills

Параметры оценки:

- Алгоритмическое мышление
- Умение работать с ошибками
- Навыки самопрезентации
- Командная работа (для групповых проектов)

Методика: Наблюдение в процессе занятий, анализ участия в обсуждениях

5.2. Листы самооценки

Вопросы для обучающихся:

- "Насколько уверенно я чувствую себя в работе с..."
- "Какие темы требуют дополнительного изучения?"
- "Что получилось лучше всего в этом проекте?"

Периодичность: Заполнение после каждого модуля

5.3. Итоговый практический экзамен

Форма: Разработка программы по заданному ТЗ

Время выполнения: 2 академических часа

Критерии:

- Соответствие результата техническому заданию
- Качество и оптимальность кода
- Соблюдение стандартов оформления
- Время выполнения

5.4. Итоговое тестирование

Структура:

- Блок А: Базовые конструкции (20 вопросов)
- Блок В: ООП и сложные темы (15 вопросов)
- Блок С: Анализ кода (5 заданий)

Система оценки: За каждый правильный ответ - 1 балл

Инструменты фиксации результатов

5.5. Электронное портфолио обучающегося

Содержание:

- Результаты всех тестов и контрольных работ
- Выполненные практические задания
- Исходные коды проектов
- Листы самооценки
- Рефлексивные эссе по итогам модулей

5.6. Сводная ведомость успеваемости

Параметры:

- Динамика прогресса по модулям
- Рейтинг среди сверстников (анонимно)
- Рекомендации для дальнейшего развития

Интерпретация результатов

5.7. Уровни освоения программы

Базовый (60-74 балла): Знание основных конструкций, умение писать простые программы

Продвинутый (75-89 баллов): Уверенное использование ООП, решение сложных задач

Углубленный (90-100 баллов): Творческое применение знаний, способность к самостоятельной проектной деятельности

5.8. Шкала перевода баллов

- 90-100 баллов - "отлично"
- 75-89 баллов - "хорошо"
- 60-74 балла - "удовлетворительно"
- менее 60 баллов - "неудовлетворительно"

V. ВОСПИТАТЕЛЬНАЯ ПРОГРАММА К ДОПОЛНИТЕЛЬНОЙ ОБЩЕРАЗВИВАЮЩЕЙ ПРОГРАММЕ «C++»

1. ЦЕЛЬ И ЗАДАЧИ ВОСПИТАТЕЛЬНОЙ РАБОТЫ

Цель: Воспитание ответственной, творческой и социально-активной личности, обладающей цифровой грамотностью, профессиональной этикой и готовой к непрерывному саморазвитию в сфере IT-технологий.

Задачи:

Гражданско-патриотическое воспитание: Формирование осознания роли отечественных IT-разработок в технологическом суверенитете страны.

Духовно-нравственное воспитание: Воспитание этичного отношения к интеллектуальной собственности, формирования культуры кибербезопасности.

Профессионально-трудовое воспитание: Развитие ответственности за качество кода, формирование культуры проектной работы.

Эстетическое воспитание: Развитие чувства гармонии и стиля в создании пользовательских интерфейсов.

Экологическое воспитание: Формирование понимания принципов энергоэффективного программирования.

2. НАПРАВЛЕНИЯ ВОСПИТАТЕЛЬНОЙ РАБОТЫ

Таблица 5

Направление	Мероприятия	Формы работы
Профессиональная этика программиста	Дискуссии о авторском праве в IT, разбор кейсов неэтичного использования технологий	Дебаты, case-study, встречи с IT-специалистами
Цифровая грамотность и безопасность	Уроки по кибергигиене, создание памяток по безопасному поведению в сети	Практикумы, проектная деятельность
Развитие softskills	Тренинги по тайм-менеджменту, командной работе, публичным выступлениям	Ролевые игры, групповые проекты
Профориентация	Экскурсии в IT-компании, встречи с выпускниками-программистами	Мастер-классы, карьерные гостиные

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ

Личностные: Ответственное отношение к результатам интеллектуального труда

Метапредметные: Сформированные навыки проектной работы и командного взаимодействия

Профессиональные: Понимание социальной значимости профессии программиста

КАЛЕНДАРНЫЙ ПЛАН ВОСПИТАТЕЛЬНОЙ РАБОТЫ

I четверть (сентябрь-октябрь)

Сентябрь

«**IT-старт**» - Торжественное посвящение в программисты

Форма: Квест с элементами тимбилдинга

Задачи: Сплочение коллектива, формирование корпоративной культуры

«**Этика в коде**» - Дискуссия о моральной ответственности программиста

Форма: Дебаты с обсуждением реальных кейсов

Задачи: Формирование профессиональной этики

Октябрь

«**Кибершит**» - Неделя цифровой безопасности

Форма: Практикум по созданию безопасных паролей

Задачи: Воспитание культуры кибербезопасности

II четверть (ноябрь-декабрь)

Ноябрь

«**IT-марафон**» - Хакатон социально значимых проектов

Форма: Командное соревнование

Задачи: Развитие социальной ответственности

Декабрь

«**Новогодний байт**» - IT-фестиваль творческих проектов

Форма: Выставка digital-арта

Задачи: Развитие креативного мышления

III четверть (январь-март)

Январь

«**Карьера в IT**» - Профориентационный интенсив

Форма: Встречи с IT-специалистами

Задачи: Осознанный выбор профессионального пути

Февраль

«**Технологии добра**» - Социальный IT-проект

Форма: Разработка программ для людей с ОВЗ

Задачи: Воспитание толерантности и социальной ответственности

Март

«**Digital-портфолио**» - Неделя IT-творчества

Форма: Конкурс программистских талантов

Задачи: Развитие навыков самопрезентации

IV четверть (апрель-май)

Апрель

«Космический код» - Тематический проект ко Дню космонавтики

Форма: Разработка космических симуляторов

Задачи: Патриотическое воспитание

Май

«Выпускной IT-балл» - Итоговое мероприятие года

Форма: Церемония награждения и презентация проектов

Задачи: Подведение итогов, планирование на следующий год

СИСТЕМА МОНИТОРИНГА

Диагностика: Анкетирование, портфолио достижений

Критерии оценки: Активность в мероприятиях, качество проектов

Формы фиксации: Электронное портфолио, рейтинговая система

ВЗАИМОДЕЙСТВИЕ С РОДИТЕЛЯМИ

IT-родительский клуб - Совместные мастер-классы

Дни открытых дверей - Презентации детских проектов

Онлайн-консультации по вопросам профессионального ориентирования

VI. СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Страуструп Б. "Программирование: принципы и практика использования C++". - М.: Вильямс, 2019. - 1248 с.
2. Липпман С.Б., Лажоие Ж. "Язык программирования C++. Базовый курс". - М.: Вильямс, 2022. - 1120 с.
3. Прата С. "Язык программирования C++. Лекции и упражнения". - М.: Вильямс, 2020. - 1184 с.

Для обучающихся:

4. Дейтел Х.М., Дейтел П.Д. "Как программировать на C++". - М.: Бином, 2021. - 1456 с.
5. Васильев А.Н. "Программирование на C++ для начинающих". - М.: Эксмо, 2023. - 352 с.
6. Шилдт Г. "C++: базовый курс". - М.: Вильямс, 2022. - 624 с.

Углубленное изучение:

7. Мейерс С. "Эффективное использование C++". - М.: ДМК Пресс, 2020. - 320 с.
8. Александреску А. "Современное проектирование на C++". - М.: Вильямс, 2021. - 400 с.
9. Саттер Г. "Решение сложных задач на C++". - М.: Вильямс, 2019. - 400 с.

Практикумы и задачки:

10. Керниган Б. "Язык программирования C++. Практикум". - М.: Диалектика, 2022. - 480 с.
11. Лафоре Р. "Объектно-ориентированное программирование в C++". - СПб.: Питер, 2023. - 928 с.

ОНЛАЙН-РЕСУРСЫ И ЭЛЕКТРОННЫЕ МАТЕРИАЛЫ

Официальная документация:

12. C++ Reference - <https://en.cppreference.com/w/>
13. ISO C++ Standard - <https://isocpp.org/>

Образовательные платформы:

14. Stepik - "Программирование на C++" (онлайн-курс)
15. Coursera - "C++ Programming: Basic Concepts"
16. UdeMy - "Современный C++ для начинающих"

Русскоязычные ресурсы:

17. CppStudio - <https://cppstudio.com/>
18. CyberForum - <https://www.cyberforum.ru/cpp-beginners/>
19. Habr - <https://habr.com/ru/hubs/cpp/articles/>