

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
Кафедра вычислительной техники

**ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №1
ПО ДИСЦИПЛИНЕ «ПРОГРАММИРОВАНИЕ»**

Факультет: Заочное отделение Преподаватель: Дубков Илья

Сергеевич

Группа: ДТ-460а

Студент: Дроздов Иван Сергеевич

Вариант: 0

Новосибирск, 2025 г.

Цель работы: Изучить структуру класса, механизм создания и использования, описание членов-данных класса и методов доступа к ним, возможность инициализации объектов класса с помощью конструкторов и уничтожение их с помощью деструкторов.

Задание: Реализовать класс в соответствии с вариантом. Класс должен обеспечивать набор методов для работы с данными. Создать перегруженные конструкторы: конструктор с параметрами, конструктор копирования, конструктор по умолчанию.

Реализовать указанные классы с динамическим выделением памяти для хранения некоторых полей. Создать деструктор для освобождения памяти. Составить демонстрационную программу.

Исходные коды модулей проекта:

```
//Date.h
#ifndef DATE_H
#define DATE_H

class Date {
private:
    int day;
    int month;
    int year;
    char* weekDay;

public:
    Date(); // Конструктор по умолчанию
    Date(int d, int m, int y, const char* wd); // Конструктор с параметрами
    Date(const Date& other); // Копирующий конструктор
    ~Date(); // Деструктор

    void display() const; // Метод для вывода даты
};

#endif

Date.cpp
#include "Date.h"
#include <cstdio>
#include <cstring>

// Конструктор по умолчанию
Date::Date() : day(1), month(1), year(2000) {
    printf("Конструктор по умолчанию\n");
    weekDay = new char[10];
    strcpy(weekDay, "Monday");
}
```

```

// Конструктор с параметрами
Date::Date(int d, int m, int y, const char* wd) : day(d), month(m), year(y) {
    printf("Конструктор с параметрами\n");
    weekDay = new char[strlen(wd) + 1];
    strcpy(weekDay, wd);
}

// Копирующий конструктор
Date::Date(const Date& other) : day(other.day), month(other.month), year(other.year) {
    printf("Копирующий конструктор\n");
    weekDay = new char[strlen(other.weekDay) + 1];
    strcpy(weekDay, other.weekDay);
}

// Деструктор
Date::~Date() {
    delete[] weekDay;
    printf("Освобождение памяти\n");
}

// Метод для вывода даты
void Date::display() const {
    printf("Дата: %d.%d.%d, День недели: %s\n", day, month, year, weekDay);
}

```

L1.cpp
`#include "Date.h"`

```

int main() {
    Date defaultDate;
    defaultDate.display();

    Date customDate(28, 12, 2024, "Friday");
    customDate.display();

    Date copiedDate(customDate);
    copiedDate.display();

    return 0;
}

```

Выходы:

В рамках первой лабораторной работы по «Программированию» было проведено практическое ознакомление с фундаментальными принципами объектно-ориентированного программирования (ООП) в C++. Основное внимание уделялось изучению класса как базовой конструкции: его объявлению, наполнению (поля, методы) и манипулированию объектами.

На практике это выразилось в самостоятельной разработке класса, который включал в себя:

- Различные типы конструкторов (по умолчанию, параметризованный и копирующий) и деструктор.

- Методы для работы с данными.
- Строковые поля, реализованные через указатели `char*` для глубокого понимания работы с памятью без привлечения STL.

В ходе работы были детально рассмотрены:

- Механизм вызова и назначение конструкторов и деструкторов, а также последовательность их выполнения.
- Ключевые различия между `class` и `struct` и применение модификаторов доступа `public` и `private`.
- Способы объявления и использования методов класса.

Итогом выполнения задания стала стабильно работающая программа на C++, которая наглядно демонстрирует процессы создания и уничтожения объектов, а также корректное управление динамической памятью с помощью операторов `new` и `delete`.