МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

##### «МИНСКИЙ ГОСУДАРСТВЕННЫЙ КОЛЛЕДЖ ЦИФРОВЫХ ТЕХНОЛОГИЙ»

##### ***Группа 82 ТП***

## **Отчет**

## **по лабораторной работе № 24**

# "ИСПОЛЬЗОВАНИЕ РАЗЛИЧНЫХ СВОЙСТВ НАСЛЕДОВАНИЯ ДЛЯ РАСШИРЕНИЯ И УТОЧНЕНИЯ КЛАССОВ"

учебный предмет

«Основы алгоритмизации и программирования»

**Исполнитель: Шатуха А.К.**

**Руководитель: Д.А.Федкевич**

Минск 2025

Применить на практике знания:

* по созданию производных классов, образованных от одного базового;
* по реализации конструкторов и методов производных классов;
* по вызову методов базовых классов.

Задания  
Для выполнения задания внесите изменения в предыдущую Лабораторную работу №22 и 23. Варианты заданий остаются прежними.

Листинг или ссылка на проект:

#include <iostream>

#include <string>

using namespace std;

class ORDER {

protected:

string orderNumber;

double amount;

public:

ORDER() : orderNumber(""), amount(0.0) {

cout << "Вызван конструктор без параметров для ORDER" << endl;

}

ORDER(const string& number, double amt) : orderNumber(number), amount(amt) {

cout << "Вызван конструктор с параметрами для ORDER" << endl;

}

ORDER(const ORDER& other) : orderNumber(other.orderNumber), amount(other.amount) {

cout << "Вызван конструктор копирования для ORDER" << endl;

}

virtual ~ORDER() {

cout << "Вызван деструктор для ORDER" << endl;

}

virtual void input() {

cout << "Введите номер: ";

getline(cin, orderNumber);

cout << "Введите сумму: ";

cin >> amount;

cin.ignore();

}

virtual void display() const {

cout << "Номер заказа: " << orderNumber << endl;

cout << "Сумма заказа: " << amount << endl;

}

};

class ORDER1 : public ORDER {

private:

string senderBank;

public:

ORDER1() : ORDER(), senderBank("") {

cout << "Вызван конструктор без параматеров" << endl;

}

ORDER1(const string& number, double amt, const string& bank)

: ORDER(number, amt), senderBank(bank) {

cout << "Вызван конструктор с параметрами" << endl;

}

ORDER1(const ORDER1& other) : ORDER(other), senderBank(other.senderBank) {

cout << "Вызван конструктор копирования" << endl;

}

~ORDER1() override {

cout << "Вызван деструктор для ORDER1" << endl;

}

void input() override {

ORDER::input();

cout << "Введите банк-отправитель: ";

getline(cin, senderBank);

}

void display() const override {

ORDER::display();

cout << "Банк-отправитель: " << senderBank << endl;

}

};

int main() {

setlocale(LC\_ALL, "ru");

cout << "\nСоздание объекта ORDER1 с конструктором без параметров:\n";

ORDER1 order1;

order1.input();

order1.display();

cout << "\nСоздание объекта ORDER1 с конструктором с параметрами:\n";

ORDER1 order2("ORD123", 500.75, "Sberbank");

order2.display();

cout << "\nСоздание копии объекта ORDER1:\n";

ORDER1 order3(order2);

order3.display();

return 0;

}

Скриншот:

