**Лабораторна робота 4**

**Бурлаченко Єгор**

**Успадкування функції**

**Код:**

#include <iostream>

using namespace std;

class Money {

protected:

int hryvnia;

int kopecks;

public:

Money(int hryvnia, int kopecks) : hryvnia(hryvnia), kopecks(kopecks) {}

void display() const {

cout << hryvnia << " грн " << kopecks << " коп" << endl;

}

Money operator+(const Money& other) {

int total\_kopecks = hryvnia \* 100 + kopecks + other.hryvnia \* 100 + other.kopecks;

int new\_hryvnia = total\_kopecks / 100;

int new\_kopecks = total\_kopecks % 100;

return Money(new\_hryvnia, new\_kopecks);

}

Money operator-(const Money& other) {

int total\_kopecks = hryvnia \* 100 + kopecks - (other.hryvnia \* 100 + other.kopecks);

int new\_hryvnia = total\_kopecks / 100;

int new\_kopecks = total\_kopecks % 100;

return Money(new\_hryvnia, new\_kopecks);

}

bool operator==(const Money& other) const {

return hryvnia == other.hryvnia && kopecks == other.kopecks;

}

bool operator!=(const Money& other) const {

return !(\*this == other);

}

};

class DerivedMoney : public Money {

public:

DerivedMoney(int hryvnia, int kopecks) : Money(hryvnia, kopecks) {}

// Перевантаження оператора додавання для класу-нащадка

DerivedMoney operator+(const DerivedMoney& other) {

int total\_kopecks = hryvnia \* 100 + kopecks + other.hryvnia \* 100 + other.kopecks;

int new\_hryvnia = total\_kopecks / 100;

int new\_kopecks = total\_kopecks % 100;

return DerivedMoney(new\_hryvnia, new\_kopecks);

}

// Перевантаження оператора віднімання для класу-нащадка

DerivedMoney operator-(const DerivedMoney& other) {

int total\_kopecks = hryvnia \* 100 + kopecks - (other.hryvnia \* 100 + other.kopecks);

int new\_hryvnia = total\_kopecks / 100;

int new\_kopecks = total\_kopecks % 100;

return DerivedMoney(new\_hryvnia, new\_kopecks);

}

};

int main() {

Money money1(10, 50);

Money money2(5, 75);

Money sum = money1 + money2;

Money difference = money1 - money2;

DerivedMoney derivedMoney1(12, 30);

DerivedMoney derivedMoney2(3, 45);

DerivedMoney derivedSum = derivedMoney1 + derivedMoney2;

DerivedMoney derivedDifference = derivedMoney1 - derivedMoney2;

money1.display();

money2.display();

sum.display();

difference.display();

derivedMoney1.display();

derivedMoney2.display();

derivedSum.display();

derivedDifference.display();

if (money1 == money2) {

cout << "Грошові суми money1 та money2 рівні." << endl;

}

else {

cout << "Грошові суми money1 та money2 не рівні." << endl;

}

if (derivedMoney1 != derivedMoney2) {

cout << "Грошові суми derivedMoney1 та derivedMoney2 не рівні." << endl;

}

else {

cout << "Грошові суми derivedMoney1 та derivedMoney2 рівні." << endl;

}

return 0;

}