Лабораторна №5

**Тема:** Віртуальні функції та поліморфізм.

**Мета:** Практично ознайомитись з поняттям поліморфізму, його застосуванням та вивчити механізм його реалізації за допомогою віртуальних функцій

Створити клас ПАРА ЧИСЕЛ. Визначити віртуальну функцію порівняння пар (пара p1 більша за пару p2, якщо перше число p1 більше за перше число p2 або перші числа рівні і друге число p1 більше за друге число p2). Створити похідні класи ДРОБОВЕ ЧИСЛО З ФІКСОВАНОЮ КРАПКОЮ (з полями: ціла частина, дробова частина), ДРОБОВЕ ЧИСЛО З ПЛАВАЮЧОЮ КРАПКОЮ (з полями: мантиса, порядок) зі своїми функціями порівняння (на рівність/нерівність, більшість/меншість). Для перевірки використати масив вказівників на об’єкти базового класу, яким присвоїти адреси об’єктів похідних класів.

#include <iostream>

using namespace std;

class Para

{

public:

Para() {

};

~Para() {

};

virtual bool bilse(int a1, int a2) = 0;

virtual bool mense(int a1, int a2) = 0;

virtual bool rivnist(int a1, int a2) = 0;

virtual int GetFirst() = 0;

virtual int GetSeccond() = 0;

};

class DrobStala:public Para {

int cila;

int drobova;

public:

DrobStala(int s1, int s2) :Para() {

cila = s1;

drobova = s2;

}

int GetFirst() override {

return cila;

}

int GetSeccond()override {

return drobova;

}

bool rivnist(int a1, int a2) override{

if (a1==cila && a2==drobova)

{

return true;

}

else

{

return false;

}

}

bool bilse(int a1, int a2) override {

if (cila > a1)

{

return true;

}

else

{

if (cila == a1)

{

if (drobova > a2)

{

return true;

}

else

{

return false;

}

}

else

{

return false;

}

}

}

bool mense(int a1, int a2) override {

if (cila < a1)

{

return true;

}

else

{

if (cila == a1)

{

if (drobova < a2)

{

return true;

}

else

{

return false;

}

}

else

{

return false;

}

}

}

}\*mas[4];

class DrobPlav:public Para

{

private:

int mantusa;

int poradok;

public:

int GetFirst()override {

return mantusa;

}

int GetSeccond()override{

return poradok;

}

DrobPlav(int s1,int s2):Para() {

mantusa = s1;

poradok = s2;

}

bool rivnist(int a1, int a2) override {

if (a1 == mantusa && a2 == poradok)

{

return true;

}

else

{

return false;

}

}

bool bilse(int a1, int a2) override {

if (poradok>a2)

{

return true;

}

else

{

if (poradok == a2)

{

if (mantusa > a1)

{

return true;

}

else

{

return false;

}

}

else

{

return false;

}

}

}

bool mense(int a1, int a2) override {

if (poradok < a2)

{

return true;

}

else

{

if (poradok == a2)

{

if (mantusa < a1)

{

return true;

}

else

{

return false;

}

}

else

{

return false;

}

}

}

};

int main()

{

Para:Para \*mass[4];

mass[0] = new DrobStala(1, 2);

mass[1] = new DrobStala(1, 1);

mass[2] = new DrobPlav(1, 1);

mass[3] = new DrobPlav(1, 2);

setlocale(LC\_CTYPE, "UKR");

cout << "Робота з числами з сталою крапкою \n";

if (mass[0]->bilse(mass[1]->GetFirst(), mass[1]->GetSeccond()))

{

cout << "Перше число більше\n";

}

else

{

cout << "Друге число більше\n";

}

if (mass[0]->rivnist(mass[1]->GetFirst(), mass[1]->GetSeccond()))

{

cout << "Числа рівні\n";

}

else

{

cout << "Числа не рівні\n";

}

cout << "Робота з числами з плаваючою крапкою\n";

if (mass[2]->bilse(mass[3]->GetFirst(), mass[3]->GetSeccond()))

{

cout << "Перше число більше\n";

}

else

{

cout << "Друге число більше\n";

}

if (mass[2]->rivnist(mass[3]->GetFirst(), mass[3]->GetSeccond()))

{

cout << "Числа рівні\n";

}

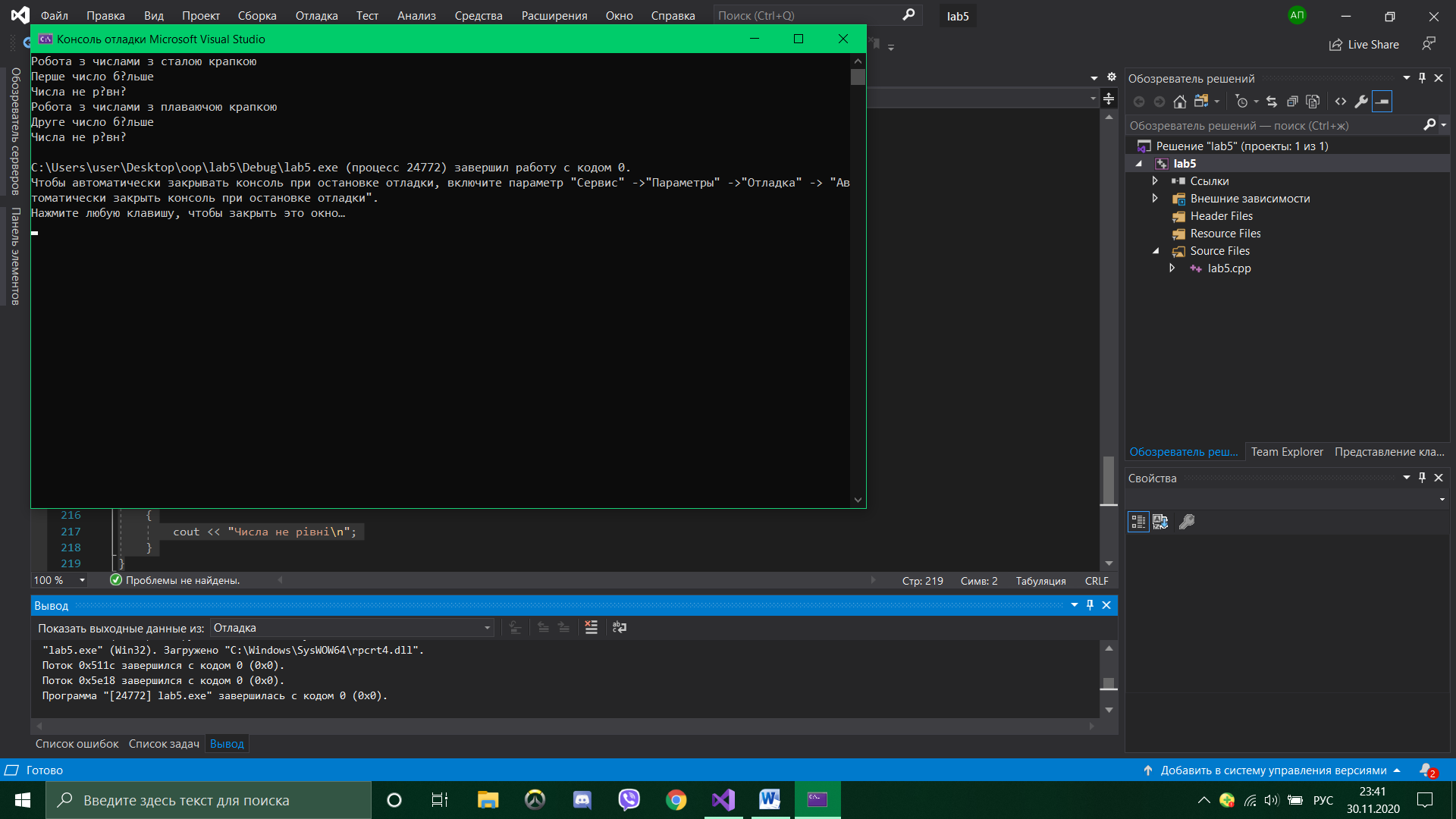
else

{

cout << "Числа не рівні\n";

}

}



**Висновок:** я практично ознайомився з поняттям поліморфізму, його застосуванням та вивчити механізм його реалізації за допомогою віртуальних функцій