Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»



Звіт

З лабораторної роботи № 5, варіант 8

З дисципліни « ПРОГРАМУВАННЯ, ЧАСТИНА 2 (ОБ’ЄКТНООРІЄНТОВАНЕ ПРОГРАМУВАННЯ) »

Виконав: ст. гр. КІ-15

Гербей О.М.

Перевірив: викладач

Козак Н.Б

Львів – 2020

Завдання: Розширити функціональність розроблених у 4 лабораторній роботі класів за допомогою операторів, що задані варіантом та оператора присвоювання. Конкретні функції операторів реалізувати на власний розсуд (крім оператора присвоювання). Організувати виведення та введення даних за допомогою класів-потоків сin, cout та перевантажених операторів вводу/виводу. Написати програму, яка демонструє роботу з об'єктами цього класу.

Код програми-рішення:

//CDice.h

# include <iostream>

class CDice {

public:

int dice1;

int dice2;

int dice3;

int tosses; // властивість об'єкта, що відповідає за кількіст ьзроблених підкидань костей

int porivn; // властивість об'єкта, що викорстовується для порівняння

//конструктор встановлює початкові значення

CDice(int zdice1 = 0, int zdice2 = 0, int zdice3 = 0, int tosses0 = 0,int porivn0=4);

//гетери

int getDice1();

int getDice2();

int getDice3();

int getTosses();

int getPorivn();

//сетери

void setDice1(int zdice1);

void setDice2(int zdice2);

void SetDice3(int zdice3);

void setPorivn(int porivn0);

void setTosses(int tosses0);

// метод класу, що відповідає за підкидання монет

void toss\_dices();

//метод класу, що виводить загальну к-сть очків після останнього кидання

//перевантаження операторів ++ += < > = вводу/виводу і == (не за умовою, в якості допоміжного)

CDice& operator++();

bool operator > (const CDice& other);

bool operator < (const CDice& other);

friend CDice& operator += (CDice& dices, CDice& other);

bool operator == (const CDice& other);

friend std::ostream& operator << (std::ostream& output, CDice& obj);

friend std::istream& operator>> (std::istream& input, CDice& obj);

CDice& operator=(const CDice& dices);

};

//Source1.cpp

#include "CDice.h"

#include<iostream>

#include <ctime>// для рандому

#include<stdlib.h>

using namespace std;

CDice::CDice(int zdice1, int zdice2, int zdice3, int tosses0, int porivn0) {

dice1 = zdice1;

dice2 = zdice2;

dice3 = zdice3;

porivn = porivn0;

tosses = tosses0;

}

int CDice::getDice1() {

return dice1;

}

int CDice::getDice2() {

return dice2;

}

int CDice::getDice3() {

return dice3;

}

int CDice::getTosses() {

return tosses;

}

int CDice::getPorivn() {

return porivn;

}

void CDice::setDice1(int zdice1) {

dice1 = zdice1;

}

void CDice::setDice2(int zdice2) {

dice2 = zdice2;

}

void CDice::SetDice3(int zdice3) {

dice3 = zdice3;

}

void CDice::setTosses(int tosses0) {

tosses = tosses0;

}

void CDice::setPorivn(int porivn0) {

porivn = porivn0;

}

void CDice::toss\_dices() {

srand(time(0));

int zdice1 = 1 + rand() % 6;

dice1 = zdice1;

int zdice2 = 1 + rand() % 6;

dice2 = zdice2;

int zdice3 = 1 + rand() % 6;

dice3 = zdice3;

}

//реалізація перевантаження операторів ++ += < > = вводу/виводу

//оператор ++ працює як лічильник, щоб підраховувати кількість зроблених підкидань костей

CDice& CDice:: operator++() {

this->tosses++;

return \*this;

}

CDice&CDice:: operator=(const CDice&dices) {

this->dice1 = dices.dice2;

this->dice2 = dices.dice3;

this->dice3 = dices.dice1;

this->porivn = dices.tosses;

this->tosses = dices.porivn;

return \*this;

}

bool CDice:: operator >( const CDice &other) {

return this->dice1 > porivn&& this->dice2 > porivn&& this->dice3 > porivn;

}

bool CDice:: operator <(const CDice& odice) {

return this->dice1 < porivn&&this->dice2>porivn&&this->dice3>porivn;

}

CDice& operator += (CDice& dices, CDice& other) {

dices.dice1 += dices.porivn;

dices.dice2 += dices.porivn;

dices.dice3 += dices.porivn;

dices.tosses += dices.porivn;

dices.porivn += dices.porivn;

return dices;

}

bool CDice:: operator == (const CDice& other) { // оператор використовується як допоміжний, щоб зробити коректне порівняння значень

return dice1 == porivn && dice2 == porivn && dice3 == porivn;

}

ostream& operator << (ostream& output, CDice& obj) {

output << "Значення на першій кості " << obj.dice1<< endl;

output << "Значення на другій кості " << obj.dice2 << endl;

output << "Значення на третій кості " << obj.dice3<< endl;

output << "Кількість кидань "<< obj.tosses << endl;

output << "Значення для порівняння "<<obj.porivn<< endl;

return output;

}

istream& operator>> (istream& input, CDice& obj) {

input >> obj.dice1;

input >> obj.dice2;

input >> obj.dice3;

input >> obj.porivn;

input >> obj.tosses;

return input;

}

//main.cpp

#include <ctime>

#include "CDice.h"

#include <Windows.h>

#include<stdlib.h>

#include <iostream>

using namespace std;

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

srand(time(NULL));

CDice dices;

while (true) {

cout << "- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -" << endl;

cout << "\*\*\*Оберiть дiю\*\*\*" << endl;

cout << "Кинути костi - натиснiть 1" << endl;

cout << "Показати результат роботи перевантажених операторів < > ++ виводу (для цього спочатку натисніть 1) - натисніть 2 " << endl;

cout << "Показати результат роботи перевантажених операторів += = вводу - натисніть 3"<<endl;

cout << "Завершити- натисніть 4" << endl;

cout << "- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - " << endl;

int func;

cin >> func;

if (func == 1) {

dices.toss\_dices();

++dices; //ілюструє роботу перевантаженого оператора ++

}

if (func == 2) {

cout <<"Ви здійснили "<< " "<< dices.tosses<<" "<< " кидань/ння" << endl;

cout << "Значення за замовчуванням, яке використовуємо для порівняння" << " " << dices.porivn << " " << endl;

cout << "Значення на костях" <<endl<<"Значення на першій кості: "<< dices.dice1 << endl<< "Значення на другій кості: "<<dices.dice2<<endl<< "Значення на третій кості: " <<dices.dice3<<endl<<endl; // ілюструє роботу перевантаженого оператора виводу

if (dices.dice1 > dices.porivn) { // ілюструє роботу перевантаженого оператора >

cout << "Значення на першій кості більше за 4 " << endl << endl;

}

else if (dices.dice1 < dices.porivn) {// ілюструє роботу перевантаженого оператора <

cout << "Значення на першій кості менше за 4" << endl << endl;

}

else if (dices.dice1 == dices.porivn) {// ілюструє роботу перевантаженого оператора <

cout << "Значення на першій кості дорінює 4" << endl << endl;

}

if (dices.dice2 > dices.porivn) {

cout << "Значення на другій кості більше за 4 " << endl << endl;

}

else if (dices.dice2 < dices.porivn) {

cout << "Значення на другій кості менше за 4" << endl << endl;

}

else if (dices.dice2 == dices.porivn) {// ілюструє роботу перевантаженого оператора <

cout << "Значення на другій кості дорінює 4" << endl << endl;

}

if (dices.dice3 > dices.porivn) {

cout << "Значення на третій кості більше за 4 " << endl << endl;

}

else if (dices.dice3 < dices.porivn) {

cout << "Значення на третій кості менше за 4" << endl << endl;

}

else if (dices.dice3 == dices.porivn) {// ілюструє роботу перевантаженого оператора <

cout << "Значення на третій кості дорінює 4" << endl << endl;

}

}

if (func == 3) {

cout << "Введіть значення першої кості ";

cin >> dices.dice1; //ілюструє роботу перевантаженого оператора вводу

cout << "Введіть значення другої кості ";

cin >> dices.dice2;

cout << "Введіть значення третьої кості ";

cin >> dices.dice3;

cout << "Введіть скільки спроб ви здійснили ";

cin >> dices.tosses;

cout << "Змініть значення за замовчуванням ";

cin >> dices.porivn;

dices.dice1 += dices.porivn;

dices.dice2 += dices.porivn;

dices.dice3 += dices.porivn;

dices.tosses += dices.porivn;

dices.porivn += dices.porivn;

cout << dices << endl<<endl;

cout << "Після переприсвоєння значень одних властивостей іншим " << endl << endl;

dices.dice1 = dices.dice2;

dices.dice2 = dices.dice3; // також демонструє роботу перевантаженого оператора =

dices.dice3 = dices.dice1;

dices.porivn = dices.tosses;

dices.tosses = dices.porivn;

cout << dices;

}

if (func == 4) {

break;

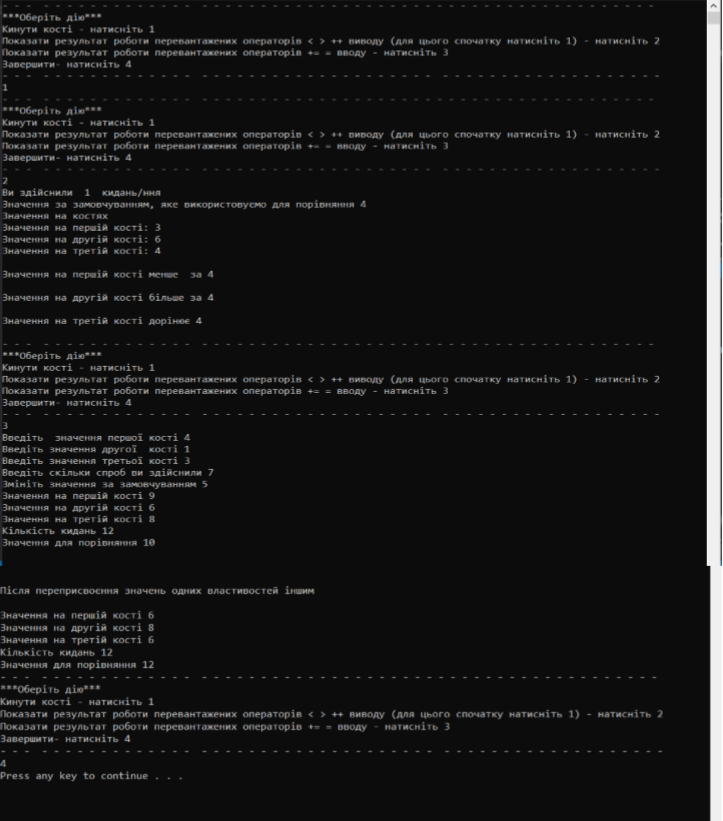
}

}

system("pause");

}

Приклад виконання програми:



Висновок: виконюючи цю лабораторну роботу, я ознайомився із перевантаженням операторів.