***Министерство образования Республики Беларусь***

***Учреждение образования***

***«Брестский государственный технический университет»***

***Кафедра ИИТ***

**Лабораторная работа №1**

**По дисциплине ППвИС за III семестр**

**Тема: «Среда программирования С++. Разработка и документирование консольных приложений. Классы»**

**Выполнил:**

Студент группы ИИ-15 (1)

2-го курса

Волк И. А.

**Проверил:**

Михняев А.Л.

Брест 2018

ЦЕЛЬ РАБОТЫ

1. Знакомство с интегрированной средой (на примере Microsoft Visual Studio). Изучение ГОСТ 19.701 и правил описания графических схем алгоритмов.

2. Знакомство с разработкой консольных приложений в технологии структурной разработки ПС и перенос задач в среду программирования С++ (например, варианты задач 1 и 2) – тестирование навыков по использованию базовых средств языка С.

3. Изучение индивидуальных заданий по разработке классов, изучение правил описания классов в терминах языка UML, описание классов.

ХОД РАБОТЫ

Получить задания на разработку классов (например, арифметического и не арифметического классов - задачи 4, а, б)

**ЗАДАЧА 1** (индивидуальное задание).

*Описание задания:*

Объект автоматизации – предметная область типа “Неарифметический класс”, например, СТУДЕНТ, ГРУППА, ТОВАР, ПРАЙС, КНИГА, РЕЙС, МАРШРУТ и т.п.

Для заданной предметной области следует разработать класс. Он должен в итоговом варианте содержать следующие члены.

1. 2-3 свойства-атрибута (один массивного типа).

2. Три конструктора (по одному каждого типа).

3. Методы, достаточные для установки и получения свойств объекта класса (например, TO\_SET(), TO\_SHOW() и т.п.).

4. Три перегруженных оператора (операторы типа “=”, а также двуместный и одноместный операторы).

*Выбранная предметная область:* компьютерная игра.

*Описание тестового примера.*

В качестве тестовых входных данных будем использовать следующие:

1. Name = “CS:GO”

Release year = 2012

Current price = 8.5

2. Name = “Dota 2”

Release year = 2003

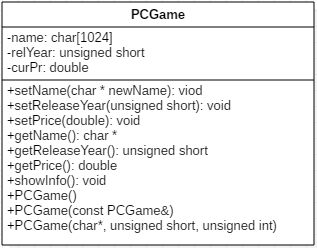
Current price = 0

*Описание классов:*

1. Имя: PCGame

Назначение: хранит информацию об игре, а именно: имя, дату выпуска и текущую цену.

UML:



*Код программы:*

#include "stdafx.h"

#include <iostream>

#include <cstring>

#include <Windows.h>

using namespace std;

#define year unsigned short

#define price double

class PCGame

{

private:

char name[1024]; // name of the game

year relYear; // release year

price curPr; // current price

public:

PCGame(char \* initName, year releaseYear, price currentPrice);

PCGame(PCGame& object);

PCGame();

void setName(char\*newName);

void setReleaseYear(year newReleaseYear);

void setPrice(price newPrice);

char \* getName();

year getReleaseYear();

price getPrice();

void showInfo();

PCGame operator=(PCGame& object);

PCGame operator+(PCGame& object);

void operator++();

};

int main(int argc, char\* argv[])

{

PCGame g1("Dota 2", 2013, 0);

g1.showInfo();

PCGame g2("CS:GO", 2012, 10);

g2.showInfo();

++g2;

g2.showInfo();

PCGame g3 = g1 + g2;

g3.showInfo();

system("pause");

return 0;

}

//Constructors

PCGame::PCGame(char \* initName, year yearOfCreation, price currentPrice)

{

this->setName(initName);

this->setReleaseYear(yearOfCreation);

this->setPrice(currentPrice);

}

PCGame::PCGame(PCGame& object)

{

this->setName(object.getName());

this->setReleaseYear(object.getReleaseYear());

this->setPrice(object.getPrice());

}

PCGame::PCGame()

{

this->setName("without\_name");

this->setReleaseYear(0);

this->setPrice(0);

}

// SET

void PCGame::setName(char\*newName)

{

strcpy(this->name, newName);

}

void PCGame::setReleaseYear(year newYearOfCreation)

{

this->relYear = newYearOfCreation;

}

void PCGame::setPrice(price newPrice)

{

this->curPr = newPrice;

}

//GET

char \* PCGame::getName()

{

char nameCopy[1024];

strcpy(nameCopy, this->name);

return nameCopy;

}

year PCGame::getReleaseYear()

{

return this->relYear;

}

price PCGame::getPrice()

{

return this->curPr;

}

void PCGame::showInfo()

{

cout << "Name: " << this->name << endl;

cout << "Release year: " << this->relYear << endl;

cout << "Current price: " << this->curPr << endl;

}

//Operators

PCGame PCGame::operator=(PCGame& object)

{

this->setName(object.getName());

this->setReleaseYear(object.getReleaseYear());

this->setPrice(object.getPrice());

return PCGame(object);

}

PCGame PCGame::operator+(PCGame& object)

{

PCGame pack;

char newName[1024] = "Pack: ";

strcat(newName, this->getName());

strcat(newName, " + ");

strcat(newName, object.getName());

pack.setName(newName);

year newYOfCr;

if (this->getReleaseYear()<object.getReleaseYear())

newYOfCr = this->getReleaseYear();

else

newYOfCr = object.getReleaseYear();

pack.setReleaseYear(newYOfCr);

pack.setPrice(this->getPrice() + object.getPrice());

return pack;

}

void PCGame::operator++()

{

this->curPr++;

}

*Результат выполнения программы:*



ЗАДАЧА 2 (индивидуальное задание).

Объект автоматизации – предметная область типа “Арифметический класс”, например, класс для обработки и хранения трех значений целого типа; класс для обработки и хранения одного значения вещественного и двух значений целого типа; класс для обработки и хранения значения комплексного типа; класс для обработки и хранения значений вектора и т.п.

Для заданной предметной области следует разработать класс аналогично ЗАДАЧЕ 1.

*Выбранная предметная область:* правильный треугольник.

*Описание тестового примера.*

В качестве тестовых входных данных будем использовать:

First cathetus = 3.0

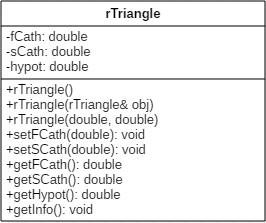
Second cathetus = -4.0

*Описание классов:*

1. Имя: rTriangle

Назначение: хранит информацию о правильном треугольнике

UML:



*Код программы:*

#include "stdafx.h"

#include <iostream>

#include <Windows.h>

#include <cmath>

using namespace std;

class rTriangle // right triangle

{

private:

double fCath; // first cathetus

double sCath; // second cathetus

double hypot; // hypotenuse

public:

rTriangle();

rTriangle(rTriangle& obj);

rTriangle(double fCath, double sCath);

void setFCath(double val);

void setSCath(double val);

double getFCath();

double getSCath();

double getHypot();

void getInfo();

const rTriangle operator=(rTriangle& obj);

const rTriangle operator\*(rTriangle& obj);

void operator~();

};

int main(int argc, char\* argv[])

{

rTriangle t1(3.0, -4.0);

t1.getInfo();

~t1;

t1.getInfo();

system("pause");

return 0;

}

// Constructors

rTriangle::rTriangle()

{

this->setFCath(3.0);

this->setSCath(4.0);

this->hypot = 5.0;

}

rTriangle::rTriangle(rTriangle& obj)

{

this->setFCath(obj.getFCath());

this->setSCath(obj.getSCath());

this->hypot = obj.getHypot();

}

rTriangle::rTriangle(double fCath, double sCath)

{

this->setFCath(fCath);

this->setSCath(sCath);

this->hypot = sqrt(pow(fCath, 2.0)+pow(sCath, 2.0));

}

// GET

void rTriangle::getInfo()

{

cout << "First cathetus: " << this->fCath << endl;

cout << "Second cathetus: " << this->sCath << endl;

cout << "Hypotenuse: " << this->hypot << endl;

}

double rTriangle::getFCath()

{

return this->fCath;

}

double rTriangle::getSCath()

{

return this->sCath;

}

double rTriangle::getHypot()

{

return this->hypot;

}

//SET

void rTriangle::setFCath(double val)

{

if(val < 0) val = -val;

this->fCath = val;

this->hypot = sqrt(pow(this->fCath, 2.0)+pow(this->sCath, 2.0));

}

void rTriangle::setSCath(double val)

{

if(val < 0) val = -val;

this->sCath = val;

this->hypot = sqrt(pow(this->fCath, 2.0)+pow(this->sCath, 2.0));

}

// Overloaded operators

const rTriangle rTriangle::operator=(rTriangle& obj)

{

this->fCath = obj.getFCath();

this->sCath = obj.getSCath();

this->hypot = obj.getHypot();

return rTriangle((\*this));

}

const rTriangle rTriangle::operator\*(rTriangle& obj)

{

rTriangle res;

res.fCath = this->fCath \* obj.getFCath();

res.sCath = this->sCath \* obj.getSCath();

res.hypot = sqrt(pow(res.fCath, 2.0)+pow(res.sCath, 2.0));

return res;

}

void rTriangle::operator~()

{

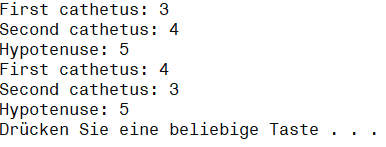
double fCath\_copy = this->fCath;

this->fCath = this->sCath;

this->sCath = fCath\_copy;

}

*Результат выполнения программы:*



Вывод: изучил правила описания классов в терминах UML, научился работать с классами.