**РК-3. Папин Алексей Владимирович. ИУ5Ц-41Б.**

**Main.cpp**

// RK\_13.05.22.cpp : Этот файл содержит функцию "main". Здесь начинается и заканчивается выполнение программы.

//

#include <iostream>

#include "Graduate.h"

#pragma warning(disable : 4996)

using namespace std;

int main()

{

system("chcp 1251 > null");

cout << "Выпускники:" << endl;

Graduate FN1[5];

FN1[0] = { "Иванов В.В.", 5, 4, 6, 2005 };

FN1[1] = { "Игнатьев А.И.", 4, 2, 6, 1997 };

FN1[2] = { "Свидиров И.И.", 3, 1, 7, 1998 };

FN1[3] = { "Дмитреев А.И.", 4, 4, 6, 1985 };

FN1[4] = { "Тестов Т.Г.", 3, 7, 3, 2012 };

//FN1[0] = { "TEST", 1, 4, 6, 6 };

//cout << FN1[0] << endl;

for (int i = 0; i < 5; i++) {

cout << FN1[i] << endl;

}

// Изменение имена

cout << "Был (Изменение имена): " << endl;

cout << FN1[0] << endl;

FN1[0].setName("Интернетов А.А.");

cout << "Стало (Изменение имена): " << endl;

cout << FN1[0] << endl;

cout << "Был (РЕЙТИНГ): " << endl;

cout << FN1[0] << endl;

// Увеличение рейтинга

FN1[0] += 3;

cout << "Стало: " << endl << FN1[0] << endl;

}

**Graduate.h**

#include <string.h>

#include <iostream>

using namespace std;

class Date {

private:

int day, month, year;

public:

// №1 КОНСТРУКТОРЫ

// Без параметров

Date() { day = 0; month = 0; year = 0; }

// С параметрами

Date(int DAY, int MONTH, int YEAR) {

day = DAY; month = MONTH; year = YEAR;

}

// №1 И ДЕСТРУКТОРЫ

~Date() {

cout << "ДЕСТРУКТОР! ~Date" << endl;

//delete &this->day, &this->month, &this->year;

}

// №2 Конструктор копирования

Date(const Date& OBJ) { day = OBJ.day; month = OBJ.month; year = OBJ.year; }

// №4 Перегрузка оператора присваивания

Date& operator=(const Date& OBJ) {

if (this != &OBJ) {

day = OBJ.day; month = OBJ.month; year = OBJ.year;

}

return \*this;

}

// №5 Перегрукза оператора вывода в консоль

friend ostream& operator<<(ostream& out\_stream, Date& OBJ) {

out\_stream << "День: " << OBJ.day << "Месяц: " << OBJ.month << "Год: " << OBJ.year << endl;

return out\_stream;

}

int get\_day() { return day; }

int get\_month() { return month; }

int get\_year() { return year; }

void set\_day(int DAY) { day = DAY; }

void set\_month(int MONTH) { month = MONTH; }

void set\_year(int YEAR) { year = YEAR; }

};

class Graduate {

private:

char\* name;

int rating;

Date\* birthday;

public:

// №1 КОНСТРУКТОРЫ

// №1 КОНСТРУКТОРЫ Без параметров

Graduate() {

name = new char[1];

strcpy\_s(name, 1, "");

rating = 0;

}

// №1 КОНСТРУКТОРЫ С параметрами

Graduate(char\* NAME, int RAT) {

delete[] name;

name = new char[strlen(NAME) + 1];

strcpy\_s(name, strlen(NAME) + 1, NAME);

rating = RAT;

}

// №1 КОНСТРУКТОРЫ С параметрами

Graduate(const char\* NAME, int RAT) {

//delete[] name;

name = new char[strlen(NAME) + 1];

strcpy\_s(name, strlen(NAME) + 1, NAME);

rating = RAT;

}

// №1 КОНСТРУКТОРЫ С параметрами

Graduate(char\* NAME, int RAT, int DAY, int MONTH, int YEAR) {

delete[] name;

name = new char[strlen(NAME) + 1];

strcpy\_s(name, strlen(NAME) + 1, NAME);

rating = RAT;

birthday = new Date[1];

birthday->set\_day(DAY);

birthday->set\_month(MONTH);

birthday->set\_year(YEAR);

//cout << birthday;

}

// №1 КОНСТРУКТОРЫ С параметрами

Graduate(const char\* NAME, int RAT, int DAY, int MONTH, int YEAR) {

//delete[] name;

name = new char[strlen(NAME) + 1];

strcpy\_s(name, strlen(NAME) + 1, NAME);

rating = RAT;

//Date tmp(DAY, MONTH, YEAR);

//cout << tmp;=

birthday = new Date[1];

birthday->set\_day(DAY);

birthday->set\_month(MONTH);

birthday->set\_year(YEAR);

//cout << birthday;

}

// №1 И ДЕСТРУКТОРЫ

~Graduate() {

//cout << "ДЕСТРУКТОР! ~Graduate" << endl;

delete[] name;

delete[] birthday;

}

// №2 Конструктор копирования

Graduate(const Graduate& OBJ) {

name = new char[strlen(OBJ.name) + 1];

strcpy\_s(name, strlen(OBJ.name) + 1, OBJ.name);

rating = OBJ.rating;

\*birthday = \*OBJ.birthday;

}

// №3 Метод setName изменения поля name объекта

void setName(char\* newName) {

delete[] name;

name = new char[strlen(newName) + 1];

strcpy\_s(name, strlen(newName) + 1, newName);

}

// №3 Метод setName изменения поля name объекта

void setName(const char\* newName) {

delete[] name;

name = new char[strlen(newName) + 1];

strcpy\_s(name, strlen(newName) + 1, newName);

}

// №4 Перегрузка оператора присваивания

Graduate& operator=(const Graduate& OBJ) {

if (this != &OBJ) {

name = new char[strlen(OBJ.name) + 1];

strcpy\_s(name, strlen(OBJ.name) + 1, OBJ.name);

rating = OBJ.rating;

birthday = OBJ.birthday;

}

return \*this;

}

// №5 Перегрукза оператора вывода в консоль

friend ostream& operator<<(ostream& out\_stream, Graduate& OBJ) {

out\_stream << "Имя: " << OBJ.name << "; " << "Рейтинг: " << OBJ.rating << endl

<< "День рождения: " << endl

<< "День: " << OBJ.birthday->get\_day() << " " << "Месяц: " << OBJ.birthday->get\_month() << " "

<< "Год: " << OBJ.birthday->get\_year() << endl;

return out\_stream;

}

// №6 Перегрузка оператора += (увеличение рейтинга на заданную констату)

friend Graduate& operator+=(Graduate& OBJ, int value) {

OBJ.rating = OBJ.rating + value;

return OBJ;

}

};

**Вывод на консольное окно:**

Выпускники:

Имя: Иванов В.В.; Рейтинг: 5

День рождения:

День: 4 Месяц: 6 Год: 2005

Имя: Игнатьев А.И.; Рейтинг: 4

День рождения:

День: 2 Месяц: 6 Год: 1997

Имя: Свидиров И.И.; Рейтинг: 3

День рождения:

День: 1 Месяц: 7 Год: 1998

Имя: Дмитреев А.И.; Рейтинг: 4

День рождения:

День: 4 Месяц: 6 Год: 1985

Имя: Тестов Т.Г.; Рейтинг: 3

День рождения:

День: 7 Месяц: 3 Год: 2012

Был (Изменение имена):

Имя: Иванов В.В.; Рейтинг: 5

День рождения:

День: 4 Месяц: 6 Год: 2005

Стало (Изменение имена):

Имя: Интернетов А.А.; Рейтинг: 5

День рождения:

День: 4 Месяц: 6 Год: 2005

Был (РЕЙТИНГ):

Имя: Интернетов А.А.; Рейтинг: 5

День рождения:

День: 4 Месяц: 6 Год: 2005

Стало:

Имя: Интернетов А.А.; Рейтинг: 8

День рождения:

День: 4 Месяц: 6 Год: 2005