**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ КЫРГЫЗСКОЙ РЕСПУБЛИКИ**

**КЫРГЫЗСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМ. И.РАЗЗАКОВА**

**ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ**

**КАФЕДРА ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ КОМПЬЮТЕРНЫХ СИСТЕМ**

**ОТЧЕТ**

**ПО ДИСЦИПЛИНЕ: ООП**

**ПО ТЕМЕ: Статические поля и методы класса**

**Лабораторная работа №9**

**ВЫПОЛНИЛ:** студент гр. ПИ(б)-3-18

Гадайбаев М.А.

**ПРОВЕРИЛА:** Каткова С.Н.

**Бишкек 2020**

**ВОПРОСЫ К ЗАЩИТЕ ЛАБОРАТОРНОЙ РАБОТЫ № 8**

1. Что такое статическое поле класса? Как оно используется объектами класса?

Статическое поле класса – это поле, которое используется для хранения данных, совместно используемых объектами класса.

1. В чем отличие статического поля от статической переменной?

Статическая переменная – это переменная, которая доступна всем объектам класса, а статическое поле – это поле, где хранится эта переменная.

1. В чем отличие статического поля класса от автоматического (обычного) поля?

В отличие от обычного поля класса, статическое поле не исчезает при удалении объекта.

1. Чем отличается определение статического и обычного полей?

Обычные поля объявляются и определяются при помощи одного оператора, а для статических полей эти два действия выполняются двумя разными операторами: объявление поля находится внутри определения класса, а определение поля располагается вне класса, и зачастую представляет собой определение глобальной переменной.

1. Для чего используют статические методы класса?

Статические методы класса предназначены для обращения к статическим класса полям класса.

1. Что такое константные методы? Для чего их используют?

Константный метод – это метод, который гарантирует, что не будет изменять объект или вызывать неконстантные методы класса. Константные методы используют для защиты от изменения значений переменных стандартных типов.

**Задание.**

**Постановка задачи:**

Написать программу для решения статическими методами следующей задачи. Из зенитного орудия производится стрельба по самолету. Вероятность попадания одного выстрела равна p. Стрельба производится n раз или до первого попадания. В программе необходимо создать класс с полем p и методом (n – аргумент метода) для вычисления результата стрельбы (0 если самолет не сбит, 1 если цель уничтожена). Предусмотреть метод для вычисления оценочного значения вероятности поражения цели (в пределе это значение стремиться к величине 1-(1-p)n).

**Описание входных данных:**

Fire – класс для вычисления результат стрельбы снаряда, поля: **p** – статическая переменная, вероятность попадания одного выстрела; **n** – статическая переменная, количество выстрелов, методы: **get\_n()** – статический метод, для получения **n**; **fireResult()** – статический метод, для получение результата стрельбы.

shoot1, shoot2 – объекты класса Fire.

k – количество выстрелов.

**Класс Fire Рис.1.**



**Рис.1.**

**Функция main() Рис.2.**

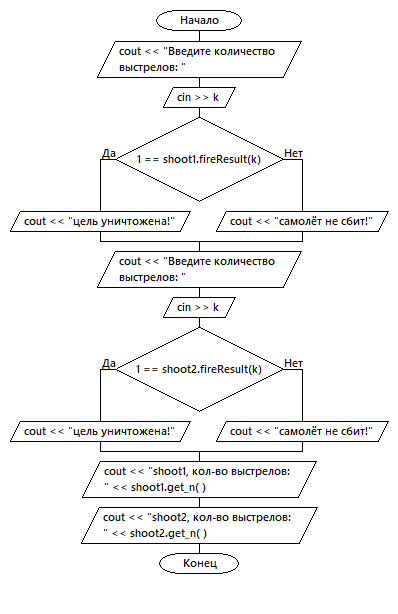
****

Рис.2.

**Метод fireResult() Рис.3.**



Рис.3.

**Код:**

#include <iostream>

using namespace std;

class Fire{

//поле класса

private:

//статичное поле

static int p;

static int n;

public:

// статичный метод

static int get\_n()

{

return n;

}

static int fireResult(int kol) { // результат стрельбы

int t;

n = kol;

t = kol / p;

cout << "Попаданий: " << t << endl;

if (t > 0){

return 1;

}

else {

return 0;

}

}

};

int Fire::n = 0; //определяем n вне класса

int Fire::p = 30; //определяем p вне класса

int main() {

setlocale(LC\_ALL,"RUS");

int choice, k;

//создания объекта

Fire shoot1, shoot2;

cout << "Введите количество выстрелов: "; cin >> k;

if (1 == shoot1.fireResult(k)) {

cout << "Результат = 1, цель уничтожена!" << endl;

}

else {

cout << "Результат = 0, самолёт не сбит!" << endl;

}

cout << endl << endl;

cout << "Введите количество выстрелов: "; cin >> k;

if (1 == shoot2.fireResult(k)) {

cout << "Результат = 1, цель уничтожена!" << endl;

}

else {

cout << "Результат = 0, самолёт не сбит!" << endl;

}

cout << endl;

cout << "Количество выстрелов у двух разных объетов одинаковая, потому что мы использовали статическое поле" << endl;

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

cout << "Объект shoot1, кол-во выстрелов: " << shoot1.get\_n() << endl;

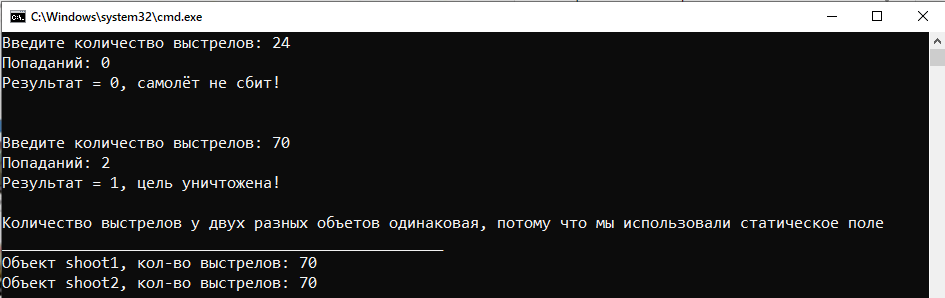
cout << "Объект shoot2, кол-во выстрелов: " << shoot2.get\_n() << endl;

system("pause>null");

return 0;

}

**Результат:**

****