**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ**

**КЫРГЫЗСКОЙ РЕСПУБЛИКИ**

**КЫРГЫЗСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ им. И.Раззакова**

**ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ**

Кафедра: **Программное обеспечение компьютерных систем**

Курс «Объектно-ориентированное программирование»

**ОТЧЕТ**

**Лабораторная работа №10**

Выполнил: студент группы ПИ-3-21

Копжашаров Азамат

Проверил: Мусабаев Э.Б.

**Бишкек 2024**

# **Задание №1**

**Постановка задачи:**

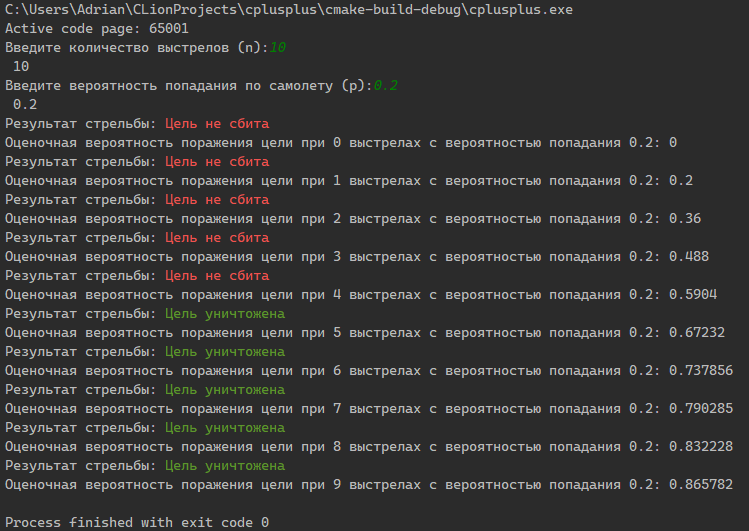
Написать программу для решения статическими методами следующей задачи. Из зенитного орудия производится стрельба по самолету. Вероятность попадания одного выстрела равна p. Стрельба производится n раз или до первого попадания. В программе необходимо создать класс с полем p и методом (n – аргумент метода) 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)).

**Исходные данные:**

**Исходный код программы на C++:**

#include <iostream>  
#include <cmath>  
#include <cstdlib>  
#include <ctime>  
using namespace std;  
  
// ANSI коды для цветов  
#define ANSI\_COLOR\_GREEN "\033[32m"  
#define ANSI\_COLOR\_RED "\033[31m"  
#define ANSI\_COLOR\_RESET "\033[0m"  
  
class ZenithGun {  
private:  
 static double p;  
  
public:  
 static void setProbability(double probability) {  
 p = probability;  
 }  
  
 static int shootDownPlane(int n) {  
 srand(time(nullptr));  
  
 for (int i = 0; i < n; i++) {  
 double random = static\_cast<double>(rand()) / RAND\_MAX;  
 if (random < p) {  
 return 1;  
 }  
 }  
 return 0;  
 }  
  
 static double estimatedProbability(int n) {  
 return 1 - pow(1 - p, n);  
 }  
};  
  
double ZenithGun::p = 0.0;  
  
int main() {  
 system("chcp 65001");  
  
 int n;  
 cout << "Введите количество выстрелов (n): ";  
 cin >> n;  
  
 double p;  
 cout << "Введите вероятность попадания по самолету (p): ";  
 cin >> p;  
  
 ZenithGun::setProbability(p);  
  
 for (int i = 0; i < n; i++) {  
 int result = ZenithGun::shootDownPlane(i);  
 if (result == 1) {  
 cout << "Результат стрельбы: " << ANSI\_COLOR\_GREEN << "Цель уничтожена" << ANSI\_COLOR\_RESET << endl;  
 } else {  
 cout << "Результат стрельбы: " << ANSI\_COLOR\_RED << "Цель не сбита" << ANSI\_COLOR\_RESET << endl;  
 }  
  
 double estimated = ZenithGun::estimatedProbability(i);  
 cout << "Оценочная вероятность поражения цели при " << i << " выстрелах с вероятностью попадания " << p << ": " << estimated << endl;  
 }  
  
 return 0;  
}

**Тесты:**



# **Ответы на вопросы:**

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

Статическое поле — это поле, принадлежащее классу, которое используется для хранения данных, совместно используемых объектами класса.

2. В чем отличие статического поля от статической переменной?

Отличие статического поля от статической переменной заключается в контексте их использования. Статическое поле обычно относится к классу и является частью его определения, а статическая переменная может быть объявлена в любом месте программы и доступна только в пределах текущего файла (если не использовать ключевое слово extern для обращения к ней из других файлов). Таким образом, статическое поле существует в рамках класса, в то время как статическая переменная существует в рамках файла, где она определена.

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

Статическое поле по своим характеристикам сходно со статической переменной: оно видимо только внутри класса. В отличие от обычного поля класса, статическое поле не исчезает при удалении объекта. Если поле данного класса описано с ключевым словом static, то значение этого поля будет одинаковым для всех объектов данного класса.

4. Чем отличается определение статического и обычного полей?

Определение статических полей класса происходит не так, как для обычных полей. Обычные поля объявляются (компилятору сообщается имя и тип поля) и определяются компилятор выделяет память под хранение поля) при помощи одного оператора. Для статических полей эти два действия выполняются двумя разными операторами: объявление происходит внутри определения класса, а определение поля, как правило, располагается вне класса и зачастую представляет собой определение глобальной переменной.

5. Для чего используют статические методы класса?

Статические методы предназначены для обращения к статическим полям класса. Они могут обращаться непосредственно только к статическим полям и вызывать только другие статические методы класса. Обращение к статическим методам производится так же, как к статическим полям: либо через имя класса, либо через имя объекта, если он уже создан.

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

Статические методы не могут быть константными или виртуальными.

В константных методах можно использовать ключевое слово const для защиты от изменения значений переменных стандартных типов, таких как, например, int.

Для того, чтобы сделать функцию константной, необходимо указать ключевое слово const после прототипа функции, но до начала тела функции. Если объявление и определение функции разделены, то модификатор const необходимо указывать дважды – как при объявлении функции, так и при ее определении. Те методы, которые лишь считывают данные из поля класса, имеет смысл делать константными, поскольку у них нет необходимости изменять значения полей объектов класса.