Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Математическое обеспечение и применение ЭВМ»

ОТЧЕТ

по лабораторной работе №1

по курсу «Объектно-ориентированное программирование» на тему

«Класс. Создание объекта. Работа с консолью.»

Выполнили: студенты группы 21ВП2

Лакеева Софья

Малькова Анастасия

Принял: к.т.н. Афонин А.Ю.

Пенза 2023

**Цель работы:**Реализовать класс. Создать на основе класса объект, вывести и изменить cодержимое его полей, запросив новые значения от пользователя. Изучить класс System.Console.

**Лабораторное задание:**

Реализовать класс, содержащий не менее 7 полей с различными типами данных:

* cтрока;
* целое;
* вещественное.

Класс должен быть оформлен в виде отдельного файла.

Создать на основе класса объект. Класс должен содержать конструктор без параметров, с одним параметром, с двумя параметрами, с количеством параметров, равным количеству полей класса.

Вывести cодержимое его полей:

* с использование переопределенного метода ToString() для класса в целом;
* непосредственно обращаясь к полю;
* реализовав метод, выводящий на экран значение определенного поля;
* одно из числовых полей вывести в шестнадцатеричном представлении.

Поля должны быть инициализированы посредством передачи параметров конструктору. Добавить возможность изменения полей каждого из типов данных, запросив их значения от пользователя.

Работа приложения должна начинаться с вывода приветственной строки с названием лабораторной работы, группы и составом бригады.

**Листинг:**

Файл Program.cs

using System;

namespace laba1

{

public class Program

{

static void Main (string[] args)

{

Console.WriteLine("Лабораторная работа № 1. Класс. Создание объекта. Работа с консолью.\n" +

"21ВП2. Вариант 5. Малькова, Лакеева");

Console.WriteLine("\n");

Airport noName = new Airport();

Console.WriteLine(noName.ToString());

Console.WriteLine("\n");

Airport penza = new Airport("Penza", "PNZ");

Airport domodedovo = new Airport("Domodedovo", "DME", 2950, 282, 2657, 25.065F, 3.2D);

penza.PrintName();

Console.WriteLine("Код ИАТА: " + penza.GetCodeIATA());

Console.WriteLine("\n");

Console.WriteLine("Название: " + domodedovo.GetName());

Console.WriteLine("Код ИАТА: " + domodedovo.GetCodeIATA());

Console.WriteLine("Количество рейсов: " + domodedovo.GetCountOfFlights());

Console.WriteLine("Количество самолётов: " + domodedovo.GetCountOfAirplanes());

Console.WriteLine("Количество сотрудников: " + domodedovo.GetCountOfEmployees());

Console.WriteLine("Количество пассажиров в год в млн: " + domodedovo.GetCountOfPassengers());

Console.WriteLine("Годовой доход в млрд.руб: " + domodedovo.GetRevenue());

Console.WriteLine("\n");

Console.WriteLine($"Введите новое название аэропорта {penza.GetName()}:");

penza.SetName(Console.ReadLine());

Console.WriteLine($"Введите новое количество сотрудников аэропорта {penza.GetName()}:");

penza.SetCountOfEmployees(Convert.ToInt32(Console.ReadLine()));

Console.WriteLine($"Введите новое значение дохода в млрд.руб аэропорта {penza.GetName()}:");

penza.SetRevenue(Convert.ToDouble(Console.ReadLine()));

Console.WriteLine("\n");

penza.ToString();

}

}

}

Файл Airport.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace laba1

{

/// <summary>

/// Класс Аэропорт

/// </summary>

class Airport

{

/// <summary> название аэропорта </summary>

private string \_name;

/// <summary> код ИАТА </summary>

private string \_codeIATA;

/// <summary> количество рейсов </summary>

private int \_countOfFlights;

/// <summary> количество самолётов </summary>

private int \_countOfAirplanes;

/// <summary> количество сотрудников </summary>

private int \_countOfEmployees;

/// <summary> количество пассажиров в год в млн </summary>

private float \_countOfPassengers;

/// <summary> доход за последний год в млрд.руб </summary>

private double \_revenue;

/// <summary>

/// конструктор без параметров

/// </summary>

public Airport()

{

\_name = "";

\_codeIATA = "";

\_countOfFlights = 0;

\_countOfAirplanes = 0;

\_countOfEmployees = 0;

\_countOfPassengers = 0F;

\_revenue = 0D;

}

/// <summary>

/// конструктор с одним параметром

/// </summary>

/// <param name="n"> Название аэропорта </param>

public Airport(string n)

{

\_name = n;

\_codeIATA = "";

\_countOfFlights = 0;

\_countOfAirplanes = 0;

\_countOfEmployees = 0;

\_countOfPassengers = 0F;

\_revenue = 0D;

}

/// <summary>

/// конструктор с двумя параметрами

/// </summary>

/// <param name="n"> Название аэропорта </param>

/// <param name="code"> Код ИАТА аэропорта </param>

public Airport(string n, string code)

{

\_name = n;

\_codeIATA = code;

\_countOfFlights = 0;

\_countOfAirplanes = 0;

\_countOfEmployees = 0;

\_countOfPassengers = 0F;

\_revenue = 0D;

}

/// <summary>

/// конструктор со всеми полями

/// </summary>

/// <param name="n"> Название аэропорта </param>

/// <param name="code"> Код ИАТА аэропорта</param>

/// <param name="flights"> Количество рейсов </param>

/// <param name="airplanes"> Количество самолётов </param>

/// <param name="employees"> Количество сотрудников </param>

/// <param name="passengers"> Количество пассажиров в год в млн </param>

/// <param name="r"> Доход за последний год в млрд.руб </param>

public Airport(string n, string code, int flights, int airplanes, int employees, float passengers, double r)

{

\_name = n;

\_codeIATA = code;

\_countOfFlights = flights;

\_countOfAirplanes = airplanes;

\_countOfEmployees = employees;

\_countOfPassengers = passengers;

\_revenue = r;

}

/// <summary>

/// Переопределенный метод ToString

/// </summary>

/// <returns> Строка с полными данными аэропорта </returns>

public override string ToString()

{

return $"Название: {\_name}\nКод ИАТА: {\_codeIATA}\nКоличество рейсов: {\_countOfFlights}\n" +

$"Количество самолётов: {\_countOfAirplanes}\nКоличество сотрудников: {\_countOfEmployees}\n" +

$"Количество пассажиров в год в млн: {\_countOfPassengers}\nГодовой доход в млрд.руб: {\_revenue}";

}

/// <summary>

/// Геттер названия аэропорта

/// </summary>

/// <returns> Название аэропорта </returns>

public string GetName()

{

return \_name;

}

/// <summary>

/// Геттер кода ИАТА аэропорта

/// </summary>

/// <returns> Код ИАТА </returns>

public string GetCodeIATA()

{

return \_codeIATA;

}

/// <summary>

/// Геттер количества рейсов

/// </summary>

/// <returns> Количество рейсов </returns>

public int GetCountOfFlights()

{

return \_countOfFlights;

}

/// <summary>

/// Геттер количества самолётов

/// </summary>

/// <returns> Количество самолётов </returns>

public int GetCountOfAirplanes()

{

return \_countOfAirplanes;

}

/// <summary>

/// Геттер количества сотрудников

/// </summary>

/// <returns> Количество сотрудников </returns>

public int GetCountOfEmployees()

{

return \_countOfEmployees;

}

/// <summary>

/// Геттер количества пассажиров в год в млн

/// </summary>

/// <returns> Количество пассажиров в год в млн </returns>

public float GetCountOfPassengers()

{

return \_countOfPassengers;

}

/// <summary>

/// Геттер дохода аэропорта за последний год в млрд.руб

/// </summary>

/// <returns> Доход за последний год в млрд.руб </returns>

public double GetRevenue()

{

return \_revenue;

}

/// <summary>

/// Сеттер названия аэропорта

/// </summary>

/// <param name="s"> Название аэропорта </param>

public void SetName(string s)

{

\_name = s;

}

/// <summary>

/// Сеттер кода ИАТА аэропорта

/// </summary>

/// <param name="s"> Код ИАТА аэропорта </param>

public void SetCodeIATA(string s)

{

\_codeIATA = s;

}

/// <summary>

/// Сеттер количества рейсов

/// </summary>

/// <param name="x"> Количество рейсов </param>

public void SetCountOfFlights(int x)

{

\_countOfFlights = x;

}

/// <summary>

/// Сеттер количества самолётов

/// </summary>

/// <param name="x"> Количество самолётов </param>

public void SetCountOfAirplanes(int x)

{

\_countOfAirplanes = x;

}

/// <summary>

/// Сеттер количества сотрудников

/// </summary>

/// <param name="x"> Количество сотрудников </param>

public void SetCountOfEmployees(int x)

{

\_countOfEmployees = x;

}

/// <summary>

/// Сеттер количества пассажиров в год в млн

/// </summary>

/// <param name="x"> Количество пассажиров в год в млн </param>

public void SetCountOfPassengers(float x)

{

\_countOfPassengers = x;

}

/// <summary>

/// Сеттер дохода аэропорта за последний год в млрд.руб

/// </summary>

/// <param name="x"> Доход аэропорта за последний год в млрд.руб</param>

public void SetRevenue(double x)

{

\_revenue = x;

}

/// <summary>

/// Вывод названия аэропорта в консоль

/// </summary>

public void PrintName()

{

Console.WriteLine($"Название: {\_name}");

}

/// <summary>

/// Вывод кода ИАТА аэропорта в консоль

/// </summary>

public void PrintCodeIATA()

{

Console.WriteLine($"Код ИАТА: {\_codeIATA}");

}

/// <summary>

/// Вывод количества рейсов в консоль

/// </summary>

public void PrintCountOfFlights()

{

Console.WriteLine($"Количество рейсов: {\_countOfFlights}");

}

/// <summary>

/// Вывод количества самолётов в консоль

/// </summary>

public void PrintCountOfAirplanes()

{

Console.WriteLine($"Количество самолётов: {\_countOfAirplanes}");

}

/// <summary>

/// Вывод количества сотрудников в консоль

/// </summary>

public void PrintCountOfEmployees()

{

Console.WriteLine($"Количество сотрудников: {\_countOfEmployees}");

}

/// <summary>

/// Вывод количества пассажиров в год в млн в консоль

/// </summary>

public void PrintCountOfPassengers()

{

Console.WriteLine($"Количество пассажиров в год в млн: {\_countOfPassengers}");

}

/// <summary>

/// Вывод годовой дохода аэропорта в млрд.руб в консоль

/// </summary>

public void PrintRevenue()

{

Console.WriteLine($"Годовой доход в млрд.руб: {\_revenue}");

}

/// <summary>

/// Переводит количество самолётов в шестнадцатиричную систему счисления

/// </summary>

/// <returns> Количество самолётов в шестнадцатиричной системе счисления </returns>

public string CountOfAirplanesHex()

{

return Convert.ToString(\_countOfAirplanes, 16);

}

}

}

# Пояснительный текст к программе

Класс Airport – класс, содержащий следующие поля со спецификатором доступа private:

* \_name – переменная типа string, хранящая название аэропорта;
* \_codeIATA **–** переменная типа string, хранящая код ИАТА аэропорта;
* \_countOfFlights – переменная типа int, хранящая количество рейсов;
* \_countOfAirplanes – переменная типа int, хранящая количество самолётов;
* \_countOfEmployees – переменная типа int, хранящая количество сотрудников;
* \_countOfPassengers – переменная типа float, хранящая количество пассажиров в год в млн;
* \_revenue – переменная типа double, хранящая доход за последний год в млрд.руб.

Класс Airport содержит следующие конструкторы:

* public Airport() – конструктор без параметров;
* public Airport(string n) – конструктор с одним параметром (название);
* public Airport(string n, string code) – конструктор с двумя параметрами (название, код ИАТА);
* public Airport(string n, string code, int flights, int airplanes, int employees, float passengers, double r) – конструктор с двумя параметрами (название, код ИАТА, кол-во рейсов, кол-во самолётов, кол-во сотрудников, кол-во пассажиров в год в млн, доход за последний год в млрд. руб).

Класс Airport содержит следующий нестатические методы:

* public override string ToString() – переопределенный метод ToString, возвращает строку с полными данными аэропорта;
* public string GetName() – геттер названия аэропорта;
* public string GetCodeIATA() – геттер кода ИАТА аэропорта;
* public int GetCountOfFlights() – геттер количества рейсов;
* public int GetCountOfAirplanes() – геттер количества самолётов;
* public int GetCountOfEmployees() – геттер количества сотрудников;
* public float GetCountOfPassengers() – геттер количества пассажиров в год в млн;
* public double GetRevenue() – геттер дохода аэропорта за последний год в млрд. руб;
* public void SetName(string s) – сеттер названия аэропорта;
* public void SetCodeIATA(string s) – сеттер кода ИАТА аэропорта;
* public void SetCountOfFlights(int x) – сеттер количества рейсов;
* public void SetCountOfAirplanes(int x) – сеттер количества самолётов;
* public void SetCountOfEmployees(int x) – сеттер количества сотрудников;
* public void SetCountOfPassengers(float x) – сеттер количества пассажиров в год в млн;
* public void SetRevenue(double x) – сеттер дохода аэропорта за последний год в млрд.руб;
* public void PrintName() – вывод названия аэропорта в консоль;
* public void PrintCodeIATA() – вывод кода ИАТА аэропорта в консоль;
* public void PrintCountOfFlights() – вывод количества рейсов в консоль;
* public void PrintCountOfAirplanes() – вывод количества самолётов в консоль;
* public void PrintCountOfEmployees() – вывод количества сотрудников в консоль;
* public void PrintCountOfPassengers() – вывод количества пассажиров в год в млн в консоль;
* public void PrintRevenue() – вывод годового дохода аэропорта в млрд.руб в консоль;
* public string CountOfAirplanesHex() - возвращает количество самолётов в шестнадцатиричную систему счисления.

# Результаты работы программы

Результаты работы программы показаны на рисунке 1.

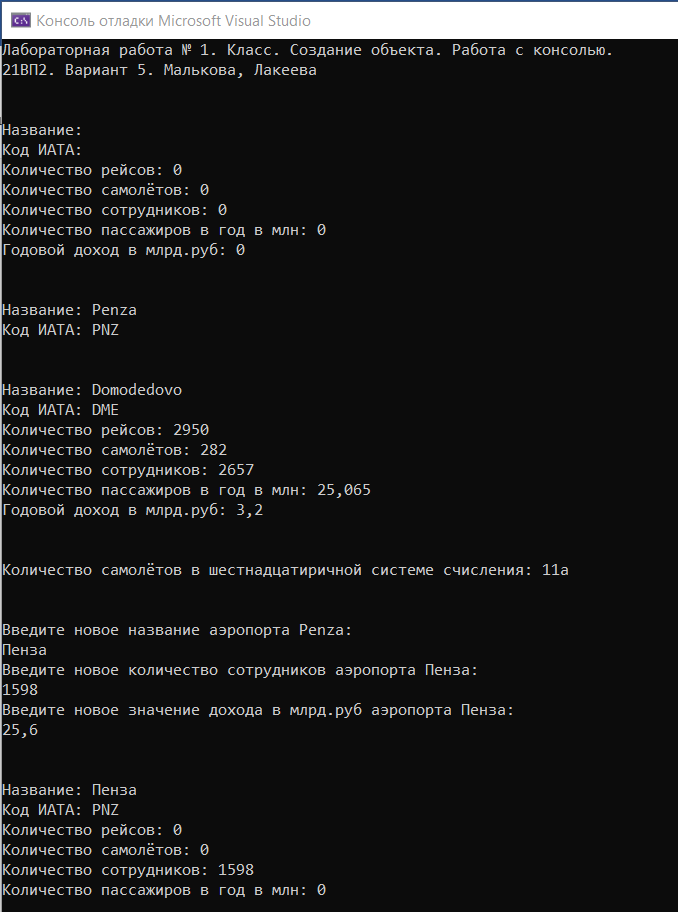


Рисунок 1 — Результаты работы программы

# Выводы

В ходе выполнения лабораторной работы был реализовать класс Аэропорт, создан на основе класса объект, выведено и изменено содержимое его полей, запрошены новые значения полей от пользователя и изучен класс System.Console.