**федеральное государственное автономное образовательное учреждение высшего образования**



**МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**(ВЫСШАЯ ШКОЛА ПЕЧАТИ И МЕДИАИНДУСТРИИ)**

**(Факультет информационных технологий)**

***(Институт Принтмедиа и информационных технологий)***

***Кафедра Информатики и информационных технологий***

**направление подготовки**

**09.03.02 «Информационные системы и технологии»**

**ЛАБОРАТОРНАЯ РАБОТА № 5**

**Дисциплина: ООП**

### Тема: Добавление в программу различных статических членов

**Выполнил(а): студент(ка) группы 221-3711**

Ежов Тимофей Алексеевич

(Фамилия И.О.)

**Дата, подпись** 20.11.2023

**Проверил:**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  ***\_\_\_\_\_\_\_\_\_\_\_***

(Фамилия И.О., степень, звание) **(Оценка)**

**Дата, подпись** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  ***\_\_\_\_\_\_\_\_\_\_\_***

(Дата) (Подпись)

**Замечания: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Москва2023**

# Листинг:

namespace LAB\_5

{

internal class Human

{

static int IdCount = 0;

protected int \_id = 0;

protected string \_name;

protected string \_secondName;

protected DateTime \_birdthDate;

protected int \_hieght;

protected bool \_isInMarrage;

public Human(string name, string secondName, DateTime birdth, int h, bool marrage)

{

\_id = IdCount++;

IdCount = IdCount + 1;

\_birdthDate = birdth;

\_name = name;

\_secondName = secondName;

\_hieght = h;

\_isInMarrage = marrage;

Humanity.AddHuman(this);

}

public int GetId()

{

return \_id;

}

public string GetInfo()

{

return $"Имя: {\_name} /nФамилая: {\_secondName} /nДата рождения: {\_birdthDate} /nРост: {\_hieght} /nВ браке ли?: {\_isInMarrage}";

}

}

}

namespace LAB\_5

{

internal class Student : Human

{

private long \_idStud;

private string \_nameStud = string.Empty;

private string \_secondNameStud = string.Empty;

private int \_Age;

private float \_AvgPoint;

public Student(string name, string secondName, int age, float avgPoint, DateTime birdthDate, int hieght, bool marry)

:base (name, secondName, birdthDate, hieght, marry)

{

if (CheckCorr.Check(name, secondName, age, avgPoint))

{

\_idStud = Convert.ToInt32(DateTime.Now.Ticks);

\_nameStud = name;

\_secondNameStud = secondName;

\_Age = age;

\_AvgPoint = avgPoint;

Console.WriteLine("Success");

}

else

{

Console.WriteLine("Try better, uncorrect data");

throw new Exception("Invalid data");

}

}

public string GetName() { return \_nameStud;}

public int GetAge() { return \_Age; }

public string GetSecondName() { return \_secondNameStud;}

public float GetAvgPoint() { return \_AvgPoint;}

public long GetIdStud() { return \_idStud;}

public string TryToChangeName(string Name)

{

if (CheckCorr.Check(Name))

{

\_nameStud = Name;

return "Name changed";

}

else

{

return "Try better, uncorrect data";

}

}

public string TryToChangeSecondName(string SecondName)

{

if (CheckCorr.Check(SecondName))

{

\_secondNameStud = SecondName;

return "Second name changed";

}

else

{

return "Try better, uncorrect data";

}

}

public string TryToCahngeAge(int age)

{

if (CheckCorr.Check(age))

{

\_Age = age;

return "Age changed";

}

else

{

return "Try better, uncorrect data";

}

}

public string TryToChangeAvgPoint(float AvgPoint)

{

if (CheckCorr.Check(AvgPoint))

{

\_AvgPoint = AvgPoint;

return "AvgPoint changed";

}

else

{

return "Try better, uncorrect data";

}

}

}

}

namespace LAB\_5

{

internal static class Humanity

{

private static readonly List<Human> humans = new();

public static void AddHuman(Human human)

{

humans.Add(human);

}

public static Human? FindHuman(int id)

{

foreach (Human human in humans)

{

if (id == human.GetId())

{

return human;

}

}

return null;

}

}

}

namespace LAB\_5

{

internal class University

{

private List<Student> \_Students = new();

private List<long> \_idCount = new();

public string AddStud(Student Stud)

{

if (Stud != null)

{

\_Students.Add(Stud);

\_idCount.Add(Stud.GetIdStud());

return "Student added";

}

else { return "Try better, uncorrect data"; }

}

public Student? FindStud(string Name, string SecondName, long Id)

{

return Search(Name, SecondName, Id);

}

public Human? FindHuman(int id)

{

if (\_Students.Count > 0)

{

if (id != 0)

{

foreach (Student Human in \_Students)

{

if (Human.GetId() == id)

{

return Human;

}

}

}

}

return null;

}

public Student? FindStud()

{

return Search();

}

public Student? FindStud(long Id)

{

return Search(id: Id);

}

private Student? Search(string name = "", string secondName = "", long id = 0)

{

// Предположим что все имена и фамилии уникальны

if (\_Students.Count > 0)

{

if (id != 0)

{

foreach (Student student in \_Students)

{

if (student.GetIdStud() == id)

{

return student;

}

}

}

if (name != "")

{

foreach (Student student in \_Students)

{

if (student.GetName() == name)

{

return student;

}

}

}

if (secondName != "")

{

foreach (Student student in \_Students)

{

if (student.GetSecondName() == secondName)

{

return student;

}

}

}

}

return null;

}

}

}