**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 № 2**

**Дисциплина: ООП**

### Тема: Структуры. Типы значений и ссылочные типы

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

Ежов Тимофей Алексеевич

(Фамилия И.О.)

**Дата, подпись** 15.09.2023

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

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

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

(Дата) (Подпись)

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

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

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

**Москва2023**

# Листинг:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Security.Cryptography;

using System.Text;

using System.Threading.Tasks;

namespace LAB\_2

{

    internal struct Atributes

    {

        internal int age = 0;

        internal string name = "Default";

        internal float mass = 1.0f;

        internal Atributes(int ag, string nam, float mas)

        {

            age = ag;

            name = nam;

            mass = mas;

        }

        internal void ChangeAge(int age)

        {

            this.age = age;

        }

        internal readonly int GetAge()

        {

            return age;

        }

        internal readonly float GetMass()

        {

            return mass;

        }

        internal void ChangeMass(float mass)

        {

            this.mass = mass;

        }

        internal void ChangeName(string name)

        {

            this.name = name;

        }

        internal readonly string GetName()

        {

            return name;

        }

        internal readonly void PrintAll()

        {

            Console.WriteLine("Name: {1} \nAge: {0} \nMass: {2}", age, name, mass);

        }

    }

    internal class Creature

    {

        private Atributes \_atributes = new();

        private string \_master;

        private int \_id;

        internal Creature(Atributes temp\_Atr, string master, int id = 0)

        {

            Random temp\_rand = new();

            \_atributes = temp\_Atr;

            \_master = master;

            if (id != 0)

            {

                \_id = id;

            }

            else

            {

                \_id = temp\_rand.Next(1000000);

            }

        }

        internal void GrowUpp()

        {

            \_atributes.ChangeAge(\_atributes.GetAge()+1);

        }

        internal void GetBigger()

        {

            \_atributes.ChangeMass(\_atributes.GetMass()+1);

        }

        internal void ReName(string NamaeWa)

        {

            \_atributes.ChangeName(NamaeWa);

        }

        internal void ChangeMaster(string MasterName)

        {

            \_master = MasterName;

        }

        internal Atributes GetAtributes()

        {

            return \_atributes;

        }

        internal string GetMaster()

        {

            return \_master;

        }

        internal int GetId()

        {

            return \_id;

        }

        internal object Clone(Atributes tempAtr, string tempMastrer) => new Creature(tempAtr, tempMastrer, this.GetId());

    }

    internal class Dog : Creature

    {

        internal Dog(Atributes temp\_Atr, string master, int id = 0) : base(temp\_Atr, master, id)

        {

        }

        internal void WOOOF()

        {

            Console.WriteLine("WOOOF");

        }

        internal object Clone(Atributes tempAtr, string tempMastrer) => new Dog(tempAtr, tempMastrer, this.GetId());

    }

}

using LAB\_2;

Atributes Alfreds = new(2, "Alfred", 10);

Dog Alfred = new(Alfreds, "Oleg");

Alfreds.PrintAll();

Alfred.WOOOF();

Atributes Mias = Alfreds with { name = "Mia" };

Dog Mia = (Dog)Alfred.Clone(Mias, "Sandra");

Mia.WOOOF();

# Результат

