**10 Механизм наследования. Отношения между классами**

Задание 1. Составить программу с одним родительским классом и потомком. Все поля должны быть закрытыми. Базовый класс должен содержать конструкторы с параметрами, методы доступа к закрытым полям, вывод полей и указанный метод. Производный класс содержит дополнения и изменения, организовать вывод новых полей потомка, при этом имена методов совпадают с именами методов базового класса. Составить тестирующую программу с выдачей результатов. Создать объекты базового и производного типов. В программе должна выполняться проверка всех разработанных элементов класса.

Варианты заданий: Базовый класс: Студент (поле: имя, средний балл s) Метод: Стипендия 300000+10000([s]-5) Потомок: Магистр (поле – специальность) Изменения в потомках: увеличить стипендию на m руб.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Xml.Linq;

namespace tsak\_1

{

class Student

{

private string \_name;

private double \_sr;

private int \_scholarship;

public string Name

{

get { return \_name; }

set { \_name = value; }

}

public double Sr

{

get { return \_sr; }

}

public int Scholarship

{

get { return \_scholarship; }

}

public Student(string name, double sr)

{

\_name = name;

\_sr = sr;

\_scholarship = 0;

}

public int СountingScholarship()

{

return 300000 + 10000 \* ((int)Sr - 5);

}

public override string ToString()

{

return $"Имя: {Name}, Cредний балл: {Sr} , Степендия: {СountingScholarship()} ";

}

}

class Master : Student

{

private string \_specialization;

public string Specialization

{

get { return \_specialization; }

}

public Master(string specialization, string name, double sr) : base(name, sr)

{

\_specialization = specialization;

}

public int СountingScholarship(int m)

{

return 300000 + 10000 \* ((int)Sr - 5) + m;

}

public override string ToString()

{

return $"Имя: {Name}, Cредний балл: {Sr} ,Специальность: {Specialization}";

}

}

}

namespace tsak\_1

{

internal class Program

{

static void Main(string[] args)

{

Console.WriteLine("Введите m: ");

int m =int.Parse(Console.ReadLine());

Student student = new Student("Ivan", 8);

Console.WriteLine(student.ToString());

Master master = new Master("programmer", "Evlampiy", 8.4);

Console.WriteLine(master.ToString() + $" Степендия: {master.СountingScholarship(m)}");

}

}

Таблица 1.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 1000 | Имя: Ivan, Cредний балл: 8 , Степендия: 330000  Имя: Evlampiy, Cредний балл: 8,4 ,Специальность: programmer Степендия: 340000 |

Анализ результатов:

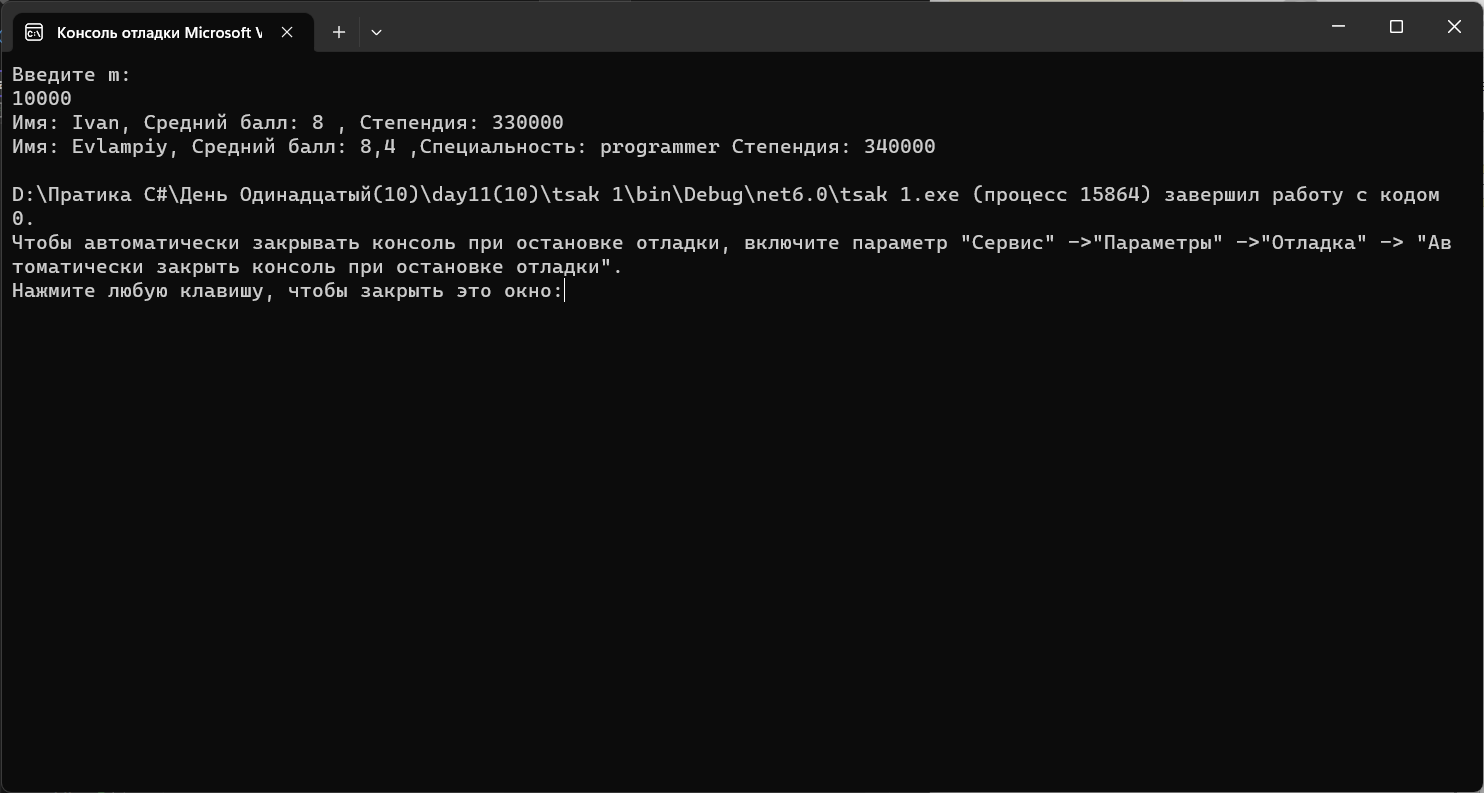


Рисунок 1.1 – Результат работы программы

Задание 2. В рамках консольного приложения разработать класс В- наследник класса А (класс А с полями а и b и свойством с. Свойство – значение выражения над полями а и b (выражение и типы полей – см. вариант в таблице 1). Поля инициализировать при объявлении класса. Конструктор оставить по умолчанию. Проследить, чтобы поля а и b напрямую в других классах были недоступны.) с полем d и свойством с2. Свойство с2 – результат вычисления выражения над полями a, b, d. В теле свойства использовать управляющий оператор (см. вариант в таблице 1) Для класса В определить 2 конструктора: один – наследуется от конструктора класса А, второй –собственный. В теле программы создать объекты классов А и В, продемонстрировав работу всех конструкторов. Вывести значения свойства на экран.

internal class Program

{

static void Main(string[] args)

{

A a = new A();

B b = new B(4.3f);

Console.WriteLine(a.C);

Console.WriteLine(b.C);

Console.WriteLine(b.C2);

Console.ReadLine();

}

}

public class A

{

private float a = 3.2f;

private float b = 5.1f;

public float C

{

get { return a \*= b / a; }

}

}

class B : A

{

private float \_d;

public float C2

{

get

{

if (base.C > 5)

return base.C \* \_d;

else

return base.C / \_d;

}

}

public B(float d) : base()

{

\_d = d;

}

Таблица 1.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | 5,1  5,1  21,93 |

Анализ результатов:

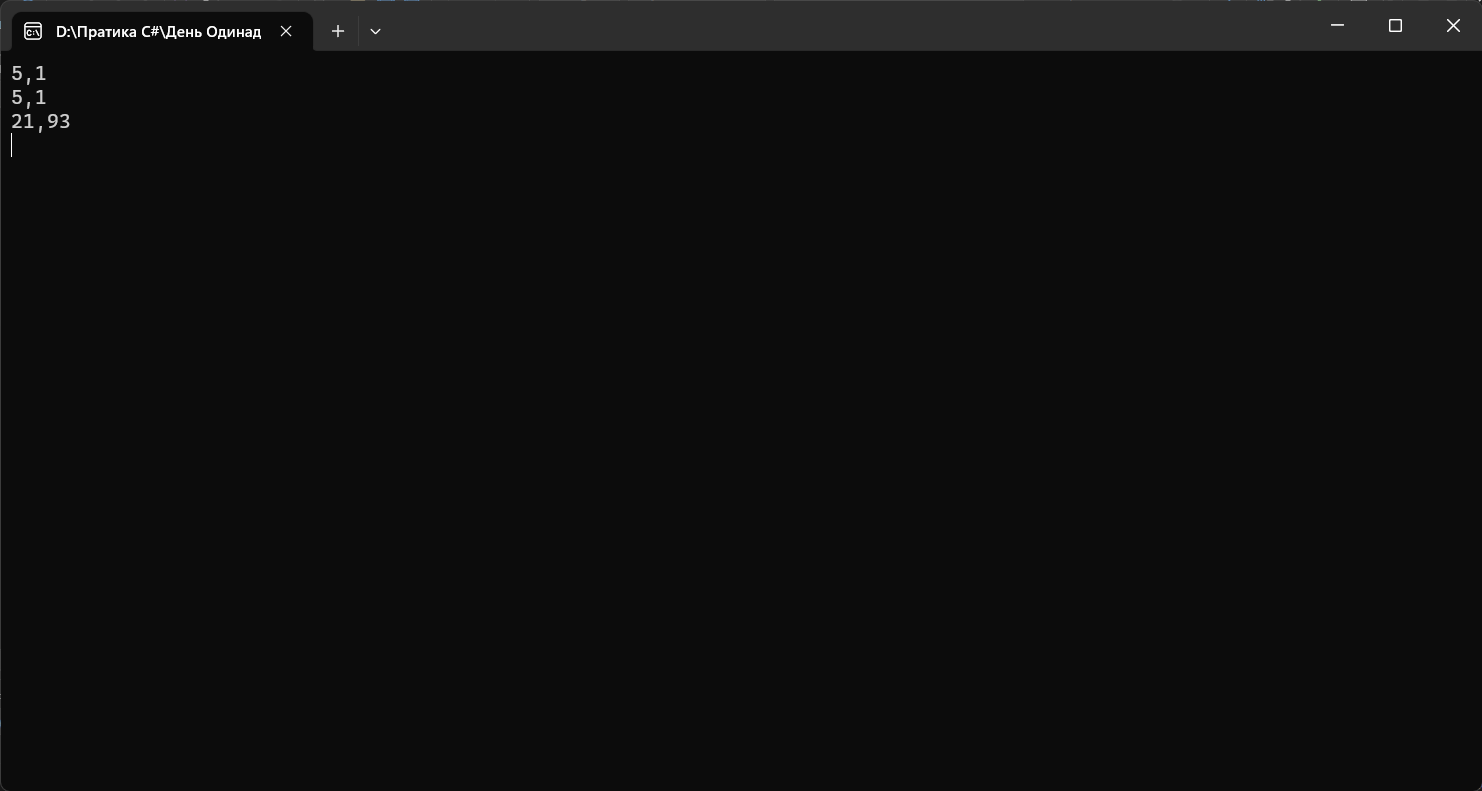
**

Рисунок 1.2 – Результат работы программы

работы программы