**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: 1. Базовый класс: Квартира (поля: название, стоимость 1м2, площадь) Метод: Стоимость квартиры. Потомок: Квартира в центре (поле название района) Изменения в потомках: увеличить стоимость с учетом надбавки за расположение на 0.01 стоимости квартиры.

Листинг программы

internal class Flat

{

public string Name;

public double Cost;

public double Square;

public Flat(string name, double cost, double square)

{

Name = name;

Cost = cost;

Square = square;

}

public virtual double CalculateCost()

{

return Cost \* Square;

}

public override string ToString()

{

return $"{Name}; {Cost}; {Square}";

}

}

internal class FlatInTheCenter : Flat

{

private const double Coeff = 0.01;

public FlatInTheCenter(string name, double cost, double square) :

base(name, cost, square)

{

}

public override double CalculateCost()

{

double baseCost = base.CalculateCost();

return baseCost + baseCost \* Coeff;

}

}

internal class Program

{

static void Main(string[] args)

{

Flat flat = new Flat("Falt", 15, 5);

FlatInTheCenter flatInTheCenter = new FlatInTheCenter("Flat", 15, 5);

Console.WriteLine($"flat: {flat}; Cost: {flat.CalculateCost()}");

Console.WriteLine($"flatInTheCenter: {flatInTheCenter}; Cost: {flatInTheCenter.CalculateCost()}");

}

}

}

Таблица 10.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Name  Square  Cost | Name  Square  Cost |

Анализ результатов:



Рисунок 10.1 – Результат работы программы

Источник: собственная разработка

Задание 2. В рамках консольного приложения разработать класс Внаследник 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 телепрограммы создать объекты классов А и В, продемонстрировав работу всех конструкторов. Вывести значения свойства на экран.

Листинг программы:

public class A

{

private decimal a;

private decimal b;

public decimal c { get; set; }

public A()

{

}

public A(decimal a, decimal b)

{

this.a = a;

this.b = b;

this.c = a - b;

}

}

public class B : A

{

private decimal d;

public decimal c2

{

get

{

while (true)

{

return d + c;

}

}

set { d = value; }

}

public B(decimal d, decimal c) : base()

{

this.d = d;

this.c2 = d + c;

}

public B()

{

}

}

}

internal class Program

{

static void Main(string[] args)

{

A a = new A(3, 2);

B b = new B(3, a.c);

Console.WriteLine($"a.c = {a.c}");

Console.WriteLine($"b.c2 = {b.c2}");

}

}

}

Таблица 10.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 3  2  3 | a.c = 1  b.c2 = 4 |

Анализ результатов:



Рисунок 10.2 – Результат работы программы

Источник: собственная разработка