**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.

Базовый класс: Животное (поля: кличка, рост – h в м) Метод: Вес животного – k \*h3 кг, где k-коэффициент. Потомок: Кошка (k= 15, поле – порода животного) Изменения в потомках: Перевести вес животного в граммы.

Листинг программы:

try

{

Cat cat = new Cat("Morty", 150, 15, "Red");

Console.WriteLine($"result = {cat.WeightAnimal()}");

//cat.SaYS();

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

class Animal

{

internal string \_Name { get; set; }

internal double \_Height { get; set; }

internal double \_Coefficient { get; set; }

public Animal(string name, double height, double coefficient)

{

this.\_Name = name;

this.\_Height = height;

this.\_Coefficient = coefficient;

}

public Animal()

{

this.\_Name = default(string);

this.\_Height = default(double);

this.\_Coefficient = default(double);

}

public int WeightAnimal()

{

double result = \_Coefficient \* \_Height;

return (int)result;

}

//public void SaYS()

//{

// Console.WriteLine("Тестирование наследования метода");

//}

}

class Cat : Animal

{

internal string \_Breed { get; set; }

public Cat(string name, double height, double coefficient, string breed) : base(name, height, coefficient)

{

this.\_Coefficient = 15;

this.\_Breed = breed;

}

public Cat(string breed)

{

this.\_Breed = breed;

}

public new int WeightAnimal()

{

var result = (\_Coefficient \* \_Height) \* 1000;

return (int)result;

}

}

Таблица 10.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | result = 2250000 |

Анализ результатов:



Рисунок 10.1 – Результаты работы программы

Источник: собственная разработка

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. Управляющий оператор – Do while.

Листинг программы:

try

{

A a = new A(10, 20);

B b = new B(0, 10, 2); // new B (a, b, d)

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

Console.ReadLine();

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

class A

{

protected int a;

protected int b;

public A(int a, int b)

{

this.a = a;

this.b = b;

}

public A()

{

this.a = default(int);

this.b = default(int);

}

}

class B : A

{

private int d;

public B(int a, int b, int d)

: base(a, b)

{

this.d = d;

}

public B(int d)

{

this.d = d;

}

public int c2

{

get

{

var sum = 0;

do

{

sum += b + d;

a++;

} while (a < 10);

return sum;

}

}

}

Таблица 10.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | Result = 120 |

Анализ результатов:



Рисунок 10.2 – Результаты работы программы

Источник: собственная разработка