**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: Телефон (поле: количество функций –k.) Метод: Стоимость: 40ln(k) Потомок: Сотовый (поле – модель) Изменения в потомках: Увеличить стоимость в 3 раза.

Листинг программы:

using \_1;

Console.Write("Enter phone model: ");

string model = Console.ReadLine();

Console.Write("Enter the number of phone functions: ");

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

CellularTelephone phone = new CellularTelephone(model,numOfFunc);

Console.WriteLine(phone);

Console.WriteLine($"Cost {phone.Cost():F2}");

class CellularTelephone : Phone

{ string \_phoneModel {get; set; }

const int mult = 3;

public CellularTelephone(string phoneModel, int k):base(k)

{\_phoneModel = phoneModel;

}

public override double Cost()

{ return base.Cost() \* mult;

}

public override string ToString()

{ return $"Model: {\_phoneModel}\n" + base.ToString();

}

}

internal class Phone

{ public int \_k { get; set; }

const int koef= 40;

public Phone(int k)

{ \_k = k;

}

public virtual double Cost()

{ return koef \* Math.Log(\_k);

}

public override string ToString()

{ return $"Number of functions {\_k}";

}

Таблица 10.1 – Входные и выходные данных задание №1

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Модель телефона: Samsung  Количество функций, которые он выполняет:45 | Модель: Samsung  Стоимость: 456,80 |

Анализ результатов:

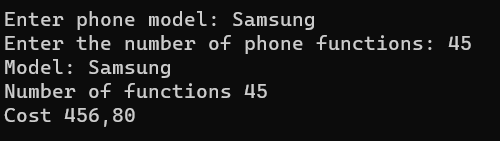


Рисунок 10.1 – Результат работы программы из задания №1

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.

Листинг программы:

using \_2;

Console.Write("Введите а: ");

double a1 = double.Parse(Console.ReadLine());

Console.Write("Введите b: ");

double b1 = double.Parse(Console.ReadLine());

Console.Write("Введите d: ");

double d1 = double.Parse(Console.ReadLine());

A a = new A();

A b = new A();

B d = new B(a1,b1,d1);

Console.WriteLine($"Сумма: {d.c2}");

public class B : A

{ private double \_d { get; set; }

public B(double d, double a, double b) : base(a, b)

{ \_d = d;

}

public B(double d)

{ \_d = d;

}

public double c2

{

get

{

int counter = 0;

do

{

\_a = \_a + \_b + \_d;

}

while (++counter < 4);

return \_a;

}

}

public class A

{

public double \_a { get; set; }

public double \_b { get; set; }

public A()

{

}

public A(double a, double b)

{

\_a = a;

\_b = b;

}

}

Таблица 10.2 – Входные и выходные данных задание №2

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| A=4  B=5  D=3 | Сумма = 33 |

Анализ результатов:

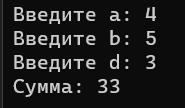


Рисунок 10.2 – Результат работы программы из задания №2