**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: Стол (поля: название, площадь S в см) Метод: Стоимость С=S2/3+k, где k – коэффициент. Потомок: Письменный (поле – используемый материал, стоимость отделки) Изменения в потомках: найти стоимость отделки, определяемую как 10% от стоимости и полную стоимость.

Листинг программы:

class Table{ private string name; private double area;

public Table(string name, double area) {this.name = name; this.area = area; }

public string Name{get { return name; } set { name = value; } }

public double Area {get { return area; } set { area = value; }}

public double GetCost(double k) {return (area \* 2 / 3 + k); }}

class WritingTable : Table{ private string material; private double decorationCost; public WritingTable(string name, double area, string material, double decorationCost) : base(name, area) // вызов конструктора базового класса

{this.material = material; this.decorationCost = decorationCost;}

public string Material{get { return material; }set { material = value; }}

public double DecorationCost{get { return decorationCost; }set { decorationCost = value; }} public new double GetCost(double k){return base.GetCost(k) + decorationCost + 0.1 \* base.GetCost(k);}}

class a1{static void Main(string[] args){Console.Write("Введите название стола:"); string tableName = Console.ReadLine();

Console.Write("Введите площадь стола:");

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

Table table = new Table(tableName, tableArea);

WritingTable writingTable = new WritingTable("Письменный стол", 800, "Дуб", 200); Console.WriteLine("Стоимость стола: {0}", table.GetCost(10));

Console.WriteLine(); Console.WriteLine("Название стола: {0}", writingTable.Name); Console.WriteLine("Площадь стола: {0}", writingTable.Area);

Console.WriteLine("Материал стола: {0}", writingTable.Material); Console.WriteLine("Стоимость отделки: {0}", writingTable.DecorationCost);

Console.WriteLine("Стоимость письменного стола: {0}", writingTable.GetCost(10));}}

Таблица 10.1 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | Введите название стола: Стол неудобный  Введите площадь стола:10  Стоимость стола: 16,666666666666668  Название стола: Письменный стол  Площадь стола: 800  Материал стола: Дуб  Стоимость отделки: 200  Стоимость письменного стола: 797,6666666666667 |

Источник: собственная разработка

Анализ результатов:

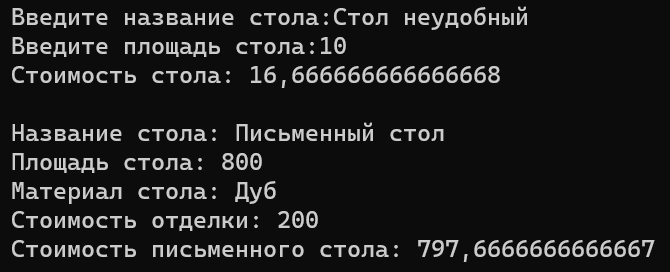


Рисунок 10.1 – Результат работы программы

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. 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 if.

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:

class A{protected int a = 10; protected int b = 20;

public int C {get{return a + b;}}}

class B : A{private int d = 30;

public int D {get{int result = C + d; if (result > 100) {return 100;}

else {return result;}}}

public B() : base() { }

public B(int a, int b, int d) { this.a = a; this.b = b; this.d = d; }

public void SetD(int newD){d = newD;}

public void PrintCD()

{Console.WriteLine("Значение свойства C класса А: " + C);

Console.WriteLine("Значение свойства D класса B: " + D);}}

class a2{static void Main(string[] args){

A aObj = new A();

Console.WriteLine("Значение свойства C класса А: " + aObj.C);

B bObj1 = new B();

Console.WriteLine("Значение свойства D класса B (конструктор 1): " + bObj1.D); B bObj2 = new B(5, 15, 25);

Console.WriteLine("Значение свойства D класса B (конструктор 2): " + bObj2.D);

bObj2.SetD(50); bObj2.PrintCD(); }

Таблица 10.2 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | Значение свойства C класса А: 30  Значение свойства D класса B (конструктор 1): 60  Значение свойства D класса B (конструктор 2): 45  Значение свойства C класса А: 20  Значение свойства D класса B: 70 |

Источник: собственная разработка

Анализ результатов:

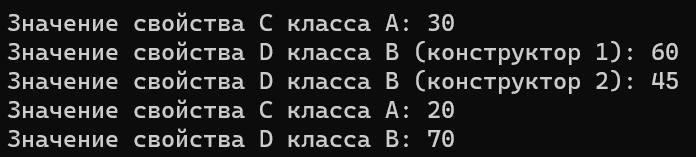


Рисунок 10.2 – Результат работы программы

Источник: собственная разработка