**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: Тетрадь (поле: название, количество листов –k). Метод: Стоимость: 153k. Потомок: Общая (поле – материал обложки). Изменения в потомках: Изменить стоимость с учетом надбавки за обложку на 50р.

Листинг программы:

class Program

{

static void Main(string[] args)

{

NoteBook note = new NoteBook("MyNotebook", 10);

Сommonplacebook general = new Сommonplacebook("MyCPbook", 10, "Leather");

Console.WriteLine(note.Price());

Console.WriteLine(general.Price());

}

}

class NoteBook

{

private string \_name;

private int \_k;

public string Name

{

get { return \_name; }

set { \_name = value; }

}

public int K

{

get { return \_k; }

set { \_k = value; }

}

public double Price()

{

return 15 \* \_k;

}

public NoteBook(string nameData, int kData)

{

\_name = nameData;

\_k = kData;

}

}

class Сommonplacebook : NoteBook

{

private string \_material;

public string Material

{

get { return \_material; }

set { \_material = value; }

}

public Сommonplacebook(string nameData, int kData, string material)

: base(nameData, kData)

{

Material = material;

}

public double Price()

{

return 15 \* K + 50;

}

}

Таблица 10.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| \_name, \_k, \_material | 150, 200 |

Источник: собственная разработка

Анализ результатов представлен на рисунке 10.1.



Рисунок 10.1 – Результат работы программы

Источник: собственная разработка

Задание №2. В рамках консольного приложения разработать класс B – наследник класса A (класс A с полями a и b и свойством c). Свойство – значение выражения над полями a и b. Поля инициализировать при объявлении класса. Конструктор оставить по умолчанию. Проследить, чтобы поля a и b напрямую в других классах были недоступны. С полем d и свойством c2. Свойство c2 – результат вычисления выражения над полями a, b, d. В теле свойства использовать управляющий оператор Do While. Для класса B определить 2 конструктора: один – наследуется от конструктора класса A, второй – собственный. В теле программы создать объекты классов A и B, продемонстрировав работу всех конструкторов. Вывести значения свойства на экран.

Листинг программы:

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;

}

}

}

internal class Program

{

static void Main(string[] args)

{

B b = new B(1, 16, 5);

Console.WriteLine($"Expression result: {b.c2}");

Console.ReadLine();

}

}

Таблица 10.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 1, 16, 5 | 189 |

Источник: собственная разработка

Анализ результатов представлен на рисунке 10.2.



Рисунок 10.2 – Результат работы программы

Источник: собственная разработка