**10. МЕХАНИЗМ НАСЛЕДОВАНИЯ. ОТНОШЕНИЯ МЕЖДУ КЛАССАМИ**

Задание 1. Базовый класс: Товар (поле – количество, цена.) Метод

Стоимость товара. Потомок: Фломастеры (поле – название, сорт – s) Изменения в потомках: Изменить стоимость фломастеров с учетом сорта в 1/√s раз.

Листинг программы:

Product prod = new Product(15, 40);

prod.Print();

try

{

Console.Write("Введите название фломастера: ");

string name = Console.ReadLine();

Console.Write("Введите сорт фломастера: ");

int variety = Convert.ToInt32(Console.ReadLine());

Console.Write("Введите количество фломастеров: ");

int quantity = Convert.ToInt32(Console.ReadLine());

Console.Write("Введите цену фломастера: ");

double price = Convert.ToDouble(Console.ReadLine());

FeltPen fP = new FeltPen(name, variety, quantity, price);

fP.ChangePrice();

fP.Print();

}

catch (Exception ex)

{

Console.WriteLine(ex.Message)

}

class Product

{

internal int Quantity { get; set; }

internal double Price { get; set; }

public Product(int quantity, double price)

{

this.Quantity = quantity;

this.Price = price;

}

public void Print()

{

Console.WriteLine($"Количество товара: {Quantity}, стоимость товара: {Price}");

}

}

class FeltPen : Product

{

private string Name { get; set; }

private int Variety { get; set; }

public FeltPen(string name, int variety, int quantity, double price) : base(quantity, price)

{

this.Name = name;

this.Variety = variety;

}

internal void ChangePrice()

{

this.Price \*= 1 / Math.Sqrt(this.Variety);

}

public new void Print() => Console.WriteLine($"Название фломастера: {Name}, сорт фломастера: {Variety}, цена фломастера: {Price}, количество фломастеров: {Quantity}");

}

Анализ результатов:

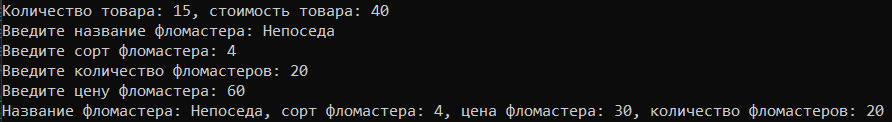


Рисунок 10.1 – Результат работы программы

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 (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:

A aObj = new(5, 3);

aObj.Print();

B bObj = new(0, 3, 4);

bObj.Print();

bObj.C2 = 3;

Console.WriteLine($"Результат: {bObj.C2}");

class A

{

public double a;

public double b;

public A(double a, double b) { this.a = a; this.b = b; }

public void Print() => Console.WriteLine($"a: {a} b: {b}");

}

class B : A

{

public double d;

public double result;

public double C2

{

get { return result; }

set

{

switch (d)

{

case 0: result = a + b + value; break;

case 1: result = a - b - value; break;

case 2: result = a \* b \* value; break;

case 3: result = a / b / value; break;

case 4: result = a % b % value; break;

}

}

}

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

{

this.d = d;

}

public new void Print() => Console.WriteLine($"a: {a} b: {b} d: {d}");

}

Таблица 10.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | 10 |

Анализ результатов:

C:\Users\danii\Downloads\пися2.png

Рисунок 10.2 – Результат работы программы

Источник: собственная разработка