## Supermarket

**Avaliable from**: Sunday, 10 July 2022, 10:00  
**Due date**: Sunday, 10 July 2022, 16:00  
**Requested files**: Program.cs, Supermarket.cs, Product.cs ([Download](https://it-kariera.mon.bg/e-learning/mod/vpl/views/downloadrequiredfiles.php?id=7624))  
**Type of work**: Individual work

# Супермаркет

### Общ преглед

Във вашата фирма постъпва проект за създаване на приложение, обслужващо супермаркет.

Вашият софтуер трябва да описва **супермаркет** **(Supermarket)** и **продукт (Product).**

Tрябва да реализирате функционалност, която да позволява някои основни команди с продуктите в супермаркета. Поредицата от команди приключва с "**END**". За ваше удобство ще получите готов **Program.cs** файл и ще трябва да реализирате само необходимите класове **Supermarket.cs** и **Product.cs**

## Подзадача 1: 30 точки

### Product

Всеки продукт има **име и цена**:

* **name** – текст, съставен от малки и / или големи латински букви
* **price** – дробно число

Създайте **конструктор**, който създава обект от класа Product по зададено **име и цена**. Създайте метод **ToString**, който връща следния текст за всеки един обект от класа Product:

* **"Product {име на продукт} costs {цена на продукта} lv."**, където цената на продукта трябва да бъде форматирана до **втория знак** след десетичната запетая.

|  |
| --- |
| **Product.cs** |
| private string name;  private double price;  public Product(string name, double price)  {  //TODO: Добавете вашия код тук …  }  public string Name  {  //TODO: Добавете вашия код тук …  }  public double Price  {  //TODO: Добавете вашия код тук …  }  public override string ToString()  {  //TODO: Добавете вашия код тук …  } |

### Supermarket

Всеки супермаркет има **име и списък с продукти**:

* **name** – текст, съставен от малки и / или големи латински букви
* **products** – списък с продукти

|  |
| --- |
| **Supermarket.cs** |
| private string name;  private List<Product> products;  public Supermarket (string name)  {  //TODO: Добавете вашия код тук …  }  public string Name  {  //TODO: Добавете вашия код тук …  }  public List<Product> Products  {  //TODO: Добавете вашия код тук …  }  public void AddProduct(string name, double price)  {  //TODO: Добавете вашия код тук …  }  public double AveragePriceInRange(double start, double end)  {  //TODO: Добавете вашия код тук …  }  public List<string> FilterProductsByPrice(double price)  {  //TODO: Добавете вашия код тук …  }  public List<Product> SortAscendingByName()  {  //TODO: Добавете вашия код тук …  }  public List<Product> SortDescendingByPrice()  {  //TODO: Добавете вашия код тук …  }  public bool CheckProductIsInSupermarket(string name)  {  //TODO: Добавете вашия код тук …  }  public string[] ProvideInformationAboutAllProducts()  {  //TODO: Добавете вашия код тук …  } |

### Командa за добавяне на продукт

Вашето приложение трябва да обслужва следната команда за добавяне на продукт:

* **Add** <**име цена**> - тази команда има за цел да добави продукт с даденото име и цена в списъка с продукти на супермаркета.

### Команда за извеждане на информация

Вашето приложение във всеки един момент може да получи заявка да отпечата информация за всички продукти. Командата за това е следната:

* **Print** - отпечатва за всички продукти в списъка с продуктите на супермаркета във формат:  
  **Product {име на продукт} costs {цена на продукта} lv.**
* **Имплементирайте метод string[] ProvideInformationAboutAllProducts() в Supermarket.cs.**
* За успешна реализация трябва да реализирате ваша версия на **ToString** метода за класа **Product.**
* **Цената на продукта трябва да бъде форматирана до втория знак след десетичната запетая.**

## Подзадача 2: 30 точки

* **SortByName** - Трябва да сортира продуктите в списъка по име във възходящ (нарастващ) ред.
* **SortByPrice** - Трябва да сортира продуктите в списъка по цена в низходящ (намаляващ) ред.
* **CheckProduct<name**> - Трябва да бъде намерен продукт, на който името отговаря на даденото. При успешно намиране трябва да върнете **true,** а в обратен случай **false.**

## Подзадача 3: 20 точки

* **FilterProducts <цена**> - Трябва да бъдат намерени всички продукти от списъка, на които цената е по-ниска от дадената в командата. Да се върне списък с имената на намерените продукти.

## Подзадача 4: 20 точки

* **AveragePrice<double start, double end**> - Трябва да бъдат взети цените на всички продукти с цена в диапазона от дробното число **start** (включително) до дробното число **end** (включително) и да се намери средноаритметичната им стойност.

### Вход

* Програмата ще получава множество редове с информация. Всеки ред представлява команда. Самият вход се обработва изцяло от примерния Program.cs.
* **Всички команди приключват с въвеждането на END**

### Изход

За някои от командите не е нужно да извеждате нищо. За други е необходимо форматиране на изход – напр. Product.ToString(), Product.Name()

### Ограничения

* Всички реални числа ще бъдат в диапазона **–10000.00** до **+10000.00**
* Имената няма да съдържат интервал

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| Add bread 4.50  Add milk 2.30  Add soap 3.40  Print  END | Added product bread.  Added product milk.  Added product soap.  Product bread costs 4.50 lv.  Product milk costs 2.30 lv.  Product soap costs 3.40 lv. |
| Add bread 4.50  Add soap 2.30  Add cheese 3.40  Add oil 3.50  Add orange 4.90  Add milk 3.40  Print  SortByName  Print  SortByPrice  Print  CheckProduct bread  CheckProduct carrot  AveragePrice 2.30 4.90  FilterProducts 3.60  END | Added product bread.  Added product soap.  Added product cheese.  Added product oil.  Added product orange.  Added product milk.  Product bread costs 4.50 lv.  Product soap costs 2.30 lv.  Product cheese costs 3.40 lv.  Product oil costs 3.50 lv.  Product orange costs 4.90 lv.  Product milk costs 3.40 lv.  First product is: bread  Product bread costs 4.50 lv.  Product cheese costs 3.40 lv.  Product milk costs 3.40 lv.  Product oil costs 3.50 lv.  Product orange costs 4.90 lv.  Product soap costs 2.30 lv.  The cheapest product is: soap  Product orange costs 4.90 lv.  Product bread costs 4.50 lv.  Product oil costs 3.50 lv.  Product cheese costs 3.40 lv.  Product milk costs 3.40 lv.  Product soap costs 2.30 lv.  Product bread is available.  Product carrot is not available.  Average price: 3.67  Filtered products: oil, cheese, milk, soap |

## Точки

Разбивката по подзадачи е следната:

1. **30** точки
2. **30** точки
3. **20** точки
4. **20** точки

Общ брой точки: **100**

## Requested files

#### Program.cs

using System;

using System.Collections.Generic;

namespace Program

{

public class Program

{

static Supermarket supermarket = new Supermarket("Supermarket1");

static void Main(string[] args)

{

string line;

while ("END" != (line = Console.ReadLine()))

{

string[] cmdArgs = line.Split(' ');

switch (cmdArgs[0])

{

case "Add":

AddProduct(cmdArgs[1], double.Parse(cmdArgs[2]));

break;

case "AveragePrice":

AveragePriceInRange(double.Parse(cmdArgs[1]), double.Parse(cmdArgs[2]));

break;

case "FilterProducts":

FilterProductsByPrice(double.Parse(cmdArgs[1]));

break;

case "SortByName":

SortAscendingByName();

break;

case "SortByPrice":

SortDescendingByPrice();

break;

case "CheckProduct":

CheckProductIsInSupermarket(cmdArgs[1]);

break;

case "Print":

ProvideInformationAboutAllProducts();

break;

}

}

}

private static void ProvideInformationAboutAllProducts()

{

string[] info = supermarket.ProvideInformationAboutAllProducts();

foreach (string item in info)

{

Console.WriteLine(item);

}

}

private static void CheckProductIsInSupermarket(string name)

{

if (supermarket.CheckProductIsInSupermarket(name))

{

Console.WriteLine($"Product {name} is available.");

}

else

{

Console.WriteLine($"Product {name} is not available.");

}

}

private static void SortDescendingByPrice()

{

supermarket.SortDescendingByPrice();

Console.WriteLine("The cheapest product is: " + supermarket.Products[supermarket.Products.Count - 1].Name);

}

private static void SortAscendingByName()

{

supermarket.SortAscendingByName();

Console.WriteLine("First product is: " + supermarket.Products[0].Name);

}

private static void FilterProductsByPrice(double grade)

{

List<string> leftStudents = supermarket.FilterProductsByPrice(grade);

Console.WriteLine("Filtered products: " + string.Join(", ", leftStudents));

}

private static void AveragePriceInRange(double start, double end)

{

double averageGrade = supermarket.AveragePriceInRange(start, end);

Console.WriteLine($"Average price: {averageGrade:f2}");

}

private static void AddProduct(string name, double price)

{

supermarket.AddProduct(name, price);

Console.WriteLine($"Added product {name}.");

}

}

}

#### Supermarket.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Program

{

internal class Supermarket

{

private string name;

private List<Product> products;

public Supermarket(string name)

{

//заменете следващия ред с вашия код

throw new NotImplementedException();

}

public string Name

{

get

{

//заменете следващия ред с вашия код

throw new NotImplementedException();

}

set

{

//заменете следващия ред с вашия код

throw new NotImplementedException();

}

}

public List<Product> Products

{

get

{

//заменете следващия ред с вашия код

throw new NotImplementedException();

}

set

{

//заменете следващия ред с вашия код

throw new NotImplementedException();

}

}

public void AddProduct(string name, double price)

{

//заменете следващия ред с вашия код

throw new NotImplementedException();

}

public double AveragePriceInRange(double start, double end)

{

//заменете следващия ред с вашия код

throw new NotImplementedException();

}

public List<string> FilterProductsByPrice(double price)

{

//заменете следващия ред с вашия код

throw new NotImplementedException();

}

public List<Product> SortAscendingByName()

{

//заменете следващия ред с вашия код

throw new NotImplementedException();

}

public List<Product> SortDescendingByPrice()

{

//заменете следващия ред с вашия код

throw new NotImplementedException();

}

public bool CheckProductIsInSupermarket(string name)

{

//заменете следващия ред с вашия код

throw new NotImplementedException();

}

public string[] ProvideInformationAboutAllProducts()

{

//заменете следващия ред с вашия код

throw new NotImplementedException();

}

}

}

#### Product.cs

using System;

using System.Collections.Generic;

using System.Text;

namespace Program

{

public class Product

{

private string name;

private double price;

public Product(string name, double price)

{

//заменете следващия ред с вашия код

throw new NotImplementedException();

}

public string Name

{

get

{

//заменете следващия ред с вашия код

throw new NotImplementedException();

}

set

{

//заменете следващия ред с вашия код

throw new NotImplementedException();

}

}

public double Price

{

get

{

//заменете следващия ред с вашия код

throw new NotImplementedException();

}

set

{

//заменете следващия ред с вашия код

throw new NotImplementedException();

}

}

public override string ToString()

{

//заменете следващия ред с вашия код

throw new NotImplementedException();

}

}

}