**Homework 10**

**Code (main):**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading;

using System.Threading.Tasks;

namespace HomeWork\_10

{

internal class Program

{

static void Main(string[] args)

{

// Object creation

Milk TestMilk = new Milk( 7.99, 2);

Bread TestBread = new Bread (1.99 , 3);

Eggs TestEggs = new Eggs (0.49, 8);

// Grocery obj creation

Grocery TestBasket = new Grocery (TestMilk, TestBread, TestEggs);

// Call of methods within classes

double total = TestBasket.expense();

Console.WriteLine("Your total today is: ${0:F2}", total);

Console.WriteLine(TestBasket.ToString());

}

}

}

**Code (grocery):**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace HomeWork\_10

{

internal class Grocery

{

// Data Members

private Milk my\_milk;

private Bread my\_bread;

private Eggs my\_eggs;

// Constructors

public Grocery (Milk milk, Bread bread, Eggs eggs)

{

my\_milk = milk;

my\_bread = bread;

my\_eggs = eggs;

}

// Methods

public double expense()

{

double total\_price = my\_milk.total\_price() + my\_bread.total\_price() + my\_eggs.total\_price();

return total\_price;

}

public override string ToString()

{

return string.Format("\nYour receipt can be broken down in: \n{0},\n{1},\n{2}", my\_milk.ToString(), my\_bread.ToString(), my\_eggs.ToString());

}

}

}

~Grocery () { }

**Code (eggs):**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace HomeWork\_10

{

internal class Eggs

{

// Data members

private double unit\_price;

private int quantity;

// Default C

public Eggs()

{

unit\_price = 0.00;

quantity = 0;

}

// Overload C

public Eggs(double x, int y)

{

unit\_price = x;

quantity = y;

}

//get n set

public double Unit\_Price

{

get { return unit\_price; }

set { unit\_price = value; }

}

public int Quantity

{

get { return quantity; }

set { quantity = value; }

}

// Methods

public double total\_price()

{

return unit\_price \* quantity;

}

public override string ToString()

{

string str = "The price of the eggs is " + unit\_price;

str += "\nThe amount of eggs bought is: " + quantity;

str += "\nWhich makes a total of: $" + total\_price() + "\n";

return str;

}

// Destructor

~Eggs() { }

}

}

**Code (bread):**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace HomeWork\_10

{

internal class Bread

{

// Data members

private double unit\_price;

private int quantity;

// Default C

public Bread()

{

unit\_price = 0.00;

quantity = 0;

}

// Overload C

public Bread(double x, int y)

{

unit\_price = x;

quantity = y;

}

//get n set

public double Unit\_Price

{

get { return unit\_price; }

set { unit\_price = value; }

}

public int Quantity

{

get { return quantity; }

set { quantity = value; }

}

// Methods

public double total\_price()

{

return unit\_price \* quantity;

}

public override string ToString()

{

string str = "The price of the bread is " + unit\_price;

str += "\nThe amount of bread bought is: " + quantity;

str += "\nWhich makes a total of: $" + total\_price() + "\n";

return str;

}

// Destructor

~Bread() { }

}

}

**Code (milk):**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace HomeWork\_10

{

internal class Milk

{

// Data members

private double unit\_price;

private int quantity;

// Default C

public Milk()

{

unit\_price = 0.00;

quantity = 0;

}

// Overload C

public Milk(double x, int y)

{

unit\_price = x;

quantity = y;

}

//get n set

public double Unit\_Price

{

get { return unit\_price; }

set { unit\_price = value; }

}

public int Quantity

{

get { return quantity; }

set { quantity = value; }

}

// Methods

public double total\_price()

{

return unit\_price \* quantity;

}

public override string ToString()

{

string str = "The price of the milk is " + unit\_price;

str += "\nThe amount of milk bought is: " + quantity;

str += "\nWhich makes a total of: $" + total\_price() + "\n";

return str;

}

// Destructor

~Milk() { }

}

}

**Output ScreanShot**

**Text

Description automatically generated**

**UML**

|  |
| --- |
| Program |
|  |
| +Main(string) |

|  |
| --- |
| Grocery |
|  |
| -my\_milk: Milk |
| -my\_bread: Bread |
| -my\_eggs: Eggs |
|  |
| +Grocery(Milk, Bread, Eggs) |
| +Expense(): double |
| +ToString (): string |
| ~Grocery() |

|  |
| --- |
| Milk |
|  |
| -unit\_price: double |
| -quantity: int |
|  |
| +Milk() |
| +Milk(double x, int y) |
| +Unit\_Price: double |
| +Quantity: int |
| +total\_price(): double |
| +ToString(): string |
| ~Milk() |

|  |
| --- |
| Bread |
|  |
| -unit\_price: double |
| -quantity: int |
|  |
| +Bread() |
| +Bread(double x, int y) |
| +Unit\_Price: double |
| +Quantity: int |
| +total\_price(): double |
| +ToString(): string |
| ~Bread() |

|  |
| --- |
| Eggs |
|  |
| -unit\_price: double |
| -quantity: int |
|  |
| +Eggs() |
| +Eggs(double x, int y) |
| +Unit\_Price: double |
| +Quantity: int |
| +total\_price(): double |
| +ToString(): string |
| ~Bread() |

**Notes:**

* The 3 classes interact with Grocery that later on interacts with the main code (I don’t know if we need to implement arrows of interactions).