



VILNIUS GEDIMINAS TECHNICAL UNIVERSITY
FACULTY OF FUNDAMENTAL SCIENCES
DEPARTMENT OF INFORMATION TECHNOLOGIES

[COFFEE VENDING MACHINE]

Object-Oriented Programming
Course Work

Author: ITfuc-21, Arailym Issayeva
Lecturer: Julius Cepukenas

VILNIUS 2022

TABLE OF CONTENTS

Abstract.....	2
1. The main logic of the program.....	3
1.1. Inheritance and Encapsulation.....	3
1.2. Abstraction and Polymorphism.....	4
2. Program code.....	4
3. Program output.....	9

ABSTRACT

A coffee vending machine is an automated machine that provides coffee to consumers.

In my program I will use 8 classes that based on object oriented programming.

Link to repository (GitHub) :

<https://github.com/araylimIT/git-course/tree/araylimIT-patch-1>

1. THE MAIN LOGIC OF THE PROGRAM

The logic of my program is as follows, for example, our client needs coffee, there are 9 coffee options in the menu, after the consumer chooses which coffee he wants, the machine will show the customer the cost of the product, and the machine will receive a message about the coffee recipe. I want to focus on all concepts of object oriented programming and describe them step by step

1.1. Inheritance and Encapsulation

All implemented classes use the principle of inheritance, especially we use Hierarchical Inheritance ("see Figure 1", for example). In this type of inheritance, more than one subclass is inherited from a single base class. i.e. more than one derived class is created from a single base class.

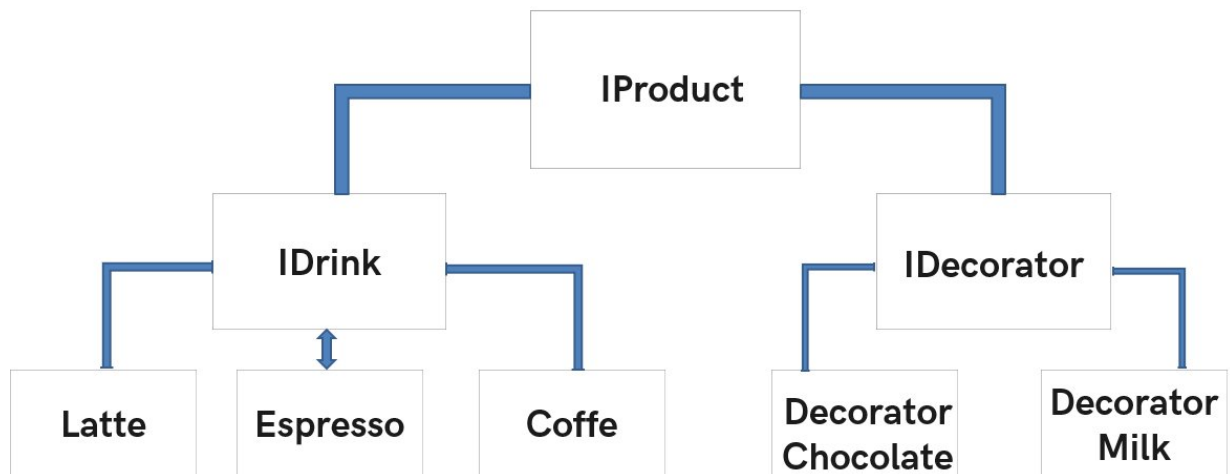


Figure 1. The figure shows the hierarchical inheritance of our program.

As a basis for all classes I used "iproducts", to show the types of coffee (latte, coffee, espresso) I used "idrink" and, "idecorator" to show additional ingredients for coffee, such as milk or chocolate.

To access the data, we used constructors (member initialization). Access methods: We mainly used protected and public methods.

The program is divided into various .cpp modules (class Implementation) and .h files (contain definitions of classes and functions) according to their contextual/functional values.

1.2. Abstraction and Polymorphism

The abstraction principle examples are idrink and idecorator, which have 2 or even 3 child elements.

Polymorphism means the presence of many forms, in our program we use it with the function of coffee preparation steps (recipe, function overriding).

2. PROGRAM CODE

Base class in “iproduct” we make all functions initially equal to zero.

```
#pragma once //'h.file'
class IProduct
{
protected:
    const double price_;
public:
    //I am writing that initially all of them are equal to 0
    IProduct(double price = 0); // constructor
    virtual double price() = 0; // function price is equal to 0
    virtual ~IProduct() {} //destructor,(virtual)
    /*I tell the program that this is a function of the base class.
    and to avoid any errors in the derived class;*/
};

#include "iproduct.h"//'cpp file'
#include <iostream>
IProduct::IProduct(double price) : price_(price) // constructor
// (member initialization)
{}
```

Derived classes of IProduct

1) IDrink this class work as a base for beverages

```
idrink.h
#pragma once
#include "iproduct.h"
class IDrink : public IProduct // inheritance
{
protected:
    virtual void recStepTwo() = 0; // functions initially zero
    virtual void recStepFour() = 0;
public:
    IDrink(double price = 0); // initially price equal to zero
    void reciepe(); //declaring function in header file
};

#include "idrink.h" // cpp file

#include <iostream>

IDrink::IDrink(double price) : IProduct(price)
{}
void IDrink::reciepe() // writing receipe to machine
{
    std::cout << "Boil water." << std::endl;
    recStepTwo();
    std::cout << "Pouring into cup." << std::endl;
    recStepFour();
}
```

2) "IDecorate" this work serves as the basis for "adding ingredients".

```
#pragma once
#include "iproduct.h" // iproduct as library to our file
#include <memory>
class IDecorator : public IProduct
{
public:
    IDecorator(IProduct* product = 0, double price = 0);
protected:
    std::shared_ptr<IProduct> m_product_;
/*A smart pointer with reference counting,
If this variable becomes zero, then the object will destroy*/ };

```

```

#include "idecorator.h"
#include <iostream>
IDecorator::IDecorator(IProduct* product, double price) :
// constructor
    IProduct(price), m_product_(product)
{}

```

1.1) Latte will cost \$ 1.8

```

// here we will work with latte
#pragma once
#include "idrink.h"
class Latte : public IDrink // inheritance
{
protected: // accessing method
    virtual void recStepTwo(); // functions of recipe
    virtual void recStepFour();
public:
    Latte(double price = 1.8); // initial price
    virtual double price();
};

#include "latte.h"
#include <iostream>
Latte::Latte(double price) : IDrink(price) // constructor
{}
double Latte::price() // function
{
    return price_;
}
void Latte::recStepTwo() // functions code for recipe
{
    std::cout << "Dripping coffe through filter." << std::endl;
    std::cout << "Heat up some milk." << std::endl;
}
void Latte::recStepFour()
{
    std::cout << "Add milk." << std::endl;
}

```

1.2) Espresso will cost \$ 1.5

```
#pragma once
#include "idrink.h"
class Espresso : public IDrink //inheritance
{
protected:
    virtual void recStepTwo(); // functions of recipe
    virtual void recStepFour();
public:
    Espresso(double price = 1.5); // initial price
    virtual double price();
};
#include "espresso.h"
#include <iostream>
Espresso::Espresso(double price) : IDrink(price) //constructor
{}
double Espresso::price() // function
{
    return price_;
}
void Espresso::recStepTwo() functions code for recipe
{
    std::cout << "Make espresso." << std::endl;
}
void Espresso::recStepFour()
{}

```

1.3) Coffee will cost \$ 1.3

```
#pragma once
#include "idrink.h"
class Coffe : public IDrink // inheritance
{
protected:
    virtual void recStepTwo(); functions of recipe
    virtual void recStepFour();
public:
    Coffe(double price = 1.3); // initial price;
    virtual double price();
};

```



```

#include "coffe.h"
#include <iostream>
Coffe::Coffe(double price) : IDrink(price) // constructor
{}
double Coffe::price() // function
{
    return price_;
}
void Coffe::recStepTwo() // functions code for recipe
{
    std::cout << "Make coffe." << std::endl;
}
void Coffe::recStepFour()
{
    std::cout << "Add sugar." << std::endl;
}

```

2.1) DecoratorChocolate

```

#pragma once
#include "idecorator.h"
class DecoratorChocolate : public IDecorator // inheritance
{
public:
    DecoratorChocolate(IProduct* product = 0, double price = 0.8); // price of chocolate
    topping will cost 0.8$
    virtual double price(); // function
};

#include "decoratorchocolate.h"

DecoratorChocolate::DecoratorChocolate(IProduct* product, double price) : // constructor
    IDecorator(product, price)
{}

double DecoratorChocolate::price() // function code
{
    return m_product_->price() + price_;
}

```

2.2) DecoratorMilk

```
#pragma once
#include "idecorator.h"
class DecoratorMilk : public IDecorator // inheritance
{
public:
    DecoratorMilk(IProduct* product = 0, double price = 0.4); // the price of milk is 0.4 $
    virtual double price(); // function
};

#include "decoratormilk.h"
DecoratorMilk::DecoratorMilk(IProduct* product, double price) : // constructor
    IDecorator(product, price)
{}

double DecoratorMilk::price() // function code
{
    return m_product->price() + price_;
}
```

3. PROGRAM OUTPUT

Our output will be like this

1st choice output will be Just Coffee cost and recipe.

```
MENU
1) Coffee
2) Coffee with milk
3) Coffee with chocolate and milk
4) Espresso
5) Espresso with milk
6) Espresso with chocolate and milk
7) Latte
8) Latte with milk
9) Latte with chocolate and milk
Enter your choice : 1

It will cost you : $ 1.3
Messeage to machine (Reciepe of Coffee is) :

Boil water.
Make coffe.
Pouring into cup.
Add sugar.

Press Y or y to confritm your order
If you want to change your order press N or n: y
```

2nd choice output (price and recipe) will be Coffee with Milk.

```
MENU
1) Coffee
2) Coffee with milk
3) Coffee with chocolate and milk
4) Espresso
5) Espresso with milk
6) Espresso with chocolate and milk
7) Latte
8) Latte with milk
9) Latte with chocolate and milk
Enter your choice : 2

It will cost you : $ 1.7
Messeage to machine (Reciepe of Coffee is) :

Boil water.
Make coffe.
Pouring into cup.
Add sugar.
Add Milk

Press Y or y to confritm your order
If you want to change your order press N or n: y
```

Our output will be like this
 3rd choice output(price and recipe) will be
 Coffee with chocolate and milk

```

MENU
1) Coffee
2) Coffee with milk
3) Coffee with chocolate and milk
4) Espresso
5) Espresso with milk
6) Espresso with chocolate and milk
7) Latte
8) Latte with milk
9) Latte with chocolate and milk
Enter your choice : 3

It will cost you : $ 2.5
Messeage to machine (Reciepe of Coffee is) :

Boil water.
Make coffe.
Pouring into cup.
Add sugar.
Add Milk and Chocolate

Press Y or y to confritm your order
If you want to change your order press N or n: y

```

Our output will be like this
 5th choice output(price and recipe) will be
 Espresso with milk

```

MENU
1) Coffee
2) Coffee with milk
3) Coffee with chocolate and milk
4) Espresso
5) Espresso with milk
6) Espresso with chocolate and milk
7) Latte
8) Latte with milk
9) Latte with chocolate and milk
Enter your choice : 5

It will cost you : $ 1.9
Messeage to machine (Reciepe of Coffee is) :

Boil water.
Make espresso.
Pouring into cup.
Add Milk

Press Y or y to confritm your order
If you want to change your order press N or n: y

```

4th choice output (price and recipe) will be just
 Espresso

```

MENU
1) Coffee
2) Coffee with milk
3) Coffee with chocolate and milk
4) Espresso
5) Espresso with milk
6) Espresso with chocolate and milk
7) Latte
8) Latte with milk
9) Latte with chocolate and milk
Enter your choice : 2

It will cost you : $ 1.7
Messeage to machine (Reciepe of Coffee is) :

Boil water.
Make coffe.
Pouring into cup.
Add sugar.
Add Milk

Press Y or y to confritm your order
If you want to change your order press N or n: y

```

6th choice output (price and recipe) will be just
 Espresso with milk and chocolate

```

MENU
1) Coffee
2) Coffee with milk
3) Coffee with chocolate and milk
4) Espresso
5) Espresso with milk
6) Espresso with chocolate and milk
7) Latte
8) Latte with milk
9) Latte with chocolate and milk
Enter your choice : 6

It will cost you : $ 2.7
Messeage to machine (Reciepe of Coffee is) :

Boil water.
Make espresso.
Pouring into cup.
Add Milk and Chocolate

Press Y or y to confritm your order
If you want to change your order press N or n: y

```

Our output will be like this
 7th choice output(price and recipe) will be just
 Latte

```

      MENU

1) Coffee
2) Coffee with milk
3) Coffee with chocolate and milk
4) Espresso
5) Espresso with milk
6) Espresso with chocolate and milk
7) Latte
8) Latte with milk
9) Latte with chocolate and milk
Enter your choice : 7

It will cost you : $ 1.8
Messeage to machine (Reciepe of Coffee is) :

Boil water.
Dripping coffe through filter.
Heat up some milk.
Pouring into cup.
Add milk.

Press Y or y to confritm your order
If you want to change your order press N or n: y
  
```

8th choice output (price and recipe) will be just
 Latte with milk

```

      MENU

1) Coffee
2) Coffee with milk
3) Coffee with chocolate and milk
4) Espresso
5) Espresso with milk
6) Espresso with chocolate and milk
7) Latte
8) Latte with milk
9) Latte with chocolate and milk
Enter your choice : 8

It will cost you : $ 2.2
Messeage to machine (Reciepe of Coffee is) :

Boil water.
Dripping coffe through filter.
Heat up some milk.
Pouring into cup.
Add milk.
Add Milk

Press Y or y to confritm your order
If you want to change your order press N or n: y
  
```

Our output will be like this
 9th choice output(price and recipe) will be
 Latte with milk and chocolate

```

      MENU

1) Coffee
2) Coffee with milk
3) Coffee with chocolate and milk
4) Espresso
5) Espresso with milk
6) Espresso with chocolate and milk
7) Latte
8) Latte with milk
9) Latte with chocolate and milk
Enter your choice : 9

It will cost you : $ 3
Messeage to machine (Reciepe of Coffee is) :

Boil water.
Dripping coffe through filter.
Heat up some milk.
Pouring into cup.
Add milk.
Add Milk and Chocolate

Press Y or y to confritm your order
If you want to change your order press N or n:
  
```