

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний аерокосмічний університет ім. М.Є. Жуковського
“Харківський авіаційний інститут”

Кафедра комп’ютерних систем, мереж і кібербезпеки

Лабораторна робота № 2

з дисципліни “Технології програмування”

Абстрактные сущности и связи между ними. Работа
с библиотеками классов.

ХАІ.503.525Б.190.123, ПЗ 1805059

Виконав студент гр. 525Б Озеров О. І

(№ групи) (П.І.Б.)

(підпис, дата)

Перевірів асистент каф. 503

(науковий ступінь, вчене звання, посада)

(підпис, дата)

Стрелкіна А А

(П.І.Б.)

2019

Тема: Абстрактные сущности и связи между ними. Работа с библиотеками классов.

Цель: Научиться описывать объекты реального мира в виде абстрактных сущностей. Получить практические навыки работы с классами и объектами. Освоить создание библиотек классов.

Постановка задачи: Необходимо разработать класс, эмулирующий работу бытового прибора, а также программу, которая демонстрирует работу с данным классом. Класс должен быть реализован в отдельной библиотеке классов. Программа должна позволять создавать несколько приборов (не более 5), выполнять через меню в консольном приложении те или иные действия с ними, например, включать и выключать выбранный прибор, изменять режим работы и т.п. При этом после каждого действия пользователя программа должна отображать на экране текущее состояние выбранного прибора и меню всех возможных действий.

Вариант 13: йогуртница

Задание 1.

У йогуртница есть такие состояния, как **выключена** и **включена**, а основные режимы работы это: **приготовление йогурта, приготовление сметаны, приготовление творога, приготовление сыра**.

На нашем абстрактном приборе включение любого режима автоматически переводит аппарат во включенное состояние, а выключение автоматически выключает ранее заданный режим работы.

Задание 2.

На рисунке ниже представлена диаграмма классов:

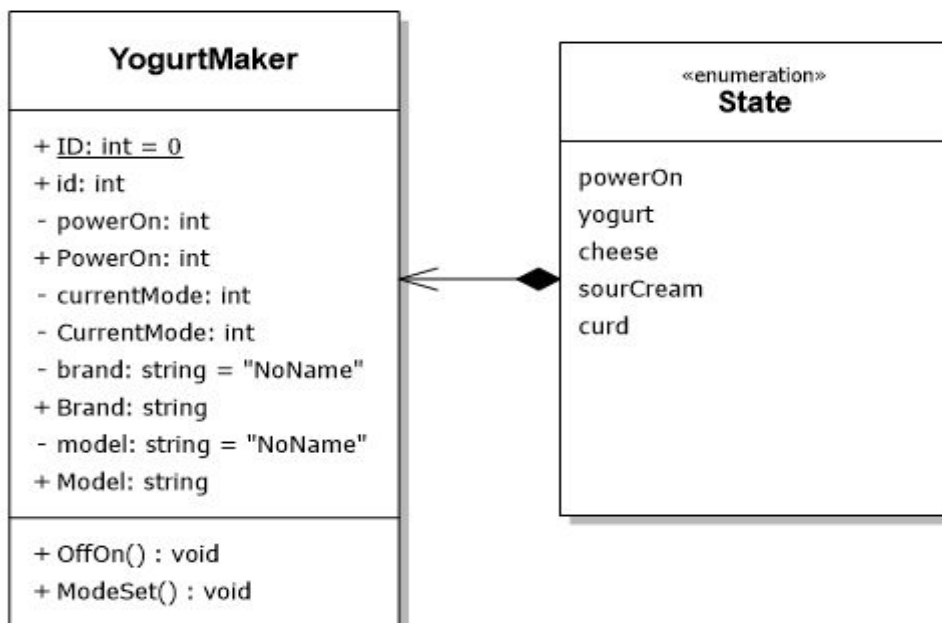


Рис. 1. Диаграмма классов

Исходные тексты программ:

```
/* **** */
/* Лабораторная работа № 2 */
/* Абстрактные сущности и связи между ними */
/* Задание 2 */
/* Выполнил студент гр. 525Б Озеров А.И. */
/* Project: HomeAppliances
 * File: YogurtMaker.cs */
/* **** */

using System;
namespace HomeAppliances
{
    // Status
    public enum Status
    {
        powerOn = 0,
        yogurt = 0,
        cheese = 0,
        sourCream = 0,
        curd = 0,
    };
    // YogurtMaker class
    public class YogurtMaker
    {
        // id of current YogurtMaker
        public int id;
        // id getter
        public int Id{get => id;}
        // id setter and blank constructor
        public static int ID = 0;
        public YogurtMaker()
        {
            id = ++ID;
        }
        // branded constructor
        public YogurtMaker(string brand)
        {
            id = ++ID;
            this.brand = brand;
        }
        // full constructor
        public YogurtMaker(string brand, string model)
        {
            id = ++ID;
            this.brand = brand;
```

```

        this.model = model;
    }
    // powerOn field and it's getter/setter
    private int powerOn = 0;
    public int PowerOn
    {
        get => powerOn;
        set => powerOn = value;
    }
    // currentMode field and it's getter/setter
    public int currentMode = 0;
    public int CurrentMode
    {
        get => currentMode;
        set => currentMode = value;
    }
    // brand field and it's getter/setter
    private string brand = "NoName";
    public string Brand
    {
        get => brand;
        set => brand = value;
    }
    // model field and it's getter/setter
    private string model = "Unknown";
    public string Model
    {
        get => model;
        set => model = value;
    }
    // OnOff switch
    public void OffOn()
    {
        if (this.PowerOn == 1)
        {
            this.PowerOn = 0;
            this.CurrentMode = 0;
        }
        else
        {
            this.PowerOn = 1;
        }
    }
    // Mode picker
    public void ModeSet(int currentMode)
    {

```

```

        if (currentMode > 4)
            currentMode %= 4;
        if (currentMode == 0)
            currentMode++;
        this.currentMode = currentMode;
    }

}
}

```

11

```

/*****/
/* Лабораторная работа № 2 */
/* Абстрактные сущности и связи между ними */
/* Задание 2 */
/* Выполнил студент гр. 525Б Озеров А.И. */
/* Project: Lab02App
 * File: Program.cs */
/*****/

using System;
using System.Collections;
using System.Collections.Generic;
using HomeAppliances;
namespace Lab02App
{
    class Program
    {
        static void Main(string[] args)
        {
            // arraylist of machines
            List<YogurtMaker> makers = new List<YogurtMaker>();

            //main loop
            while (true)
            {
                //writing list of machines
                Console.ForegroundColor = ConsoleColor.Green;
                Console.WriteLine("List of your YogurtMakers:");
                Console.WriteLine("-----");
                for (int j = 0; j < makers.Count; j++)
                {
                    Console.WriteLine("ID: {0}", makers[j].Id);
                    Console.WriteLine("Mode: {0}", makers[j].CurrentMode);
                    Console.WriteLine("Power: {0}", makers[j].PowerOn);
                }
            }
        }
    }
}

```

```

        Console.WriteLine("Brand: {0}", makers[j].Brand);
        Console.WriteLine("Model: {0}", makers[j].Model);
        Console.WriteLine("-----");
    }

    // menu
    Console.ForegroundColor = ConsoleColor.Yellow;
    Console.WriteLine("List of commands:\n" +
        "1 - create new YogurtMaker (max 5);\n" +
        "2 - delete specified machine;\n" +
        "3 - turn off/on specified machine;\n" +
        "4 - change mode of specified machine.");

    Console.WriteLine("Type number of command you want to run:");
    int command = Convert.ToInt32(Console.ReadLine());
    Console.ForegroundColor = ConsoleColor.White;
    //command handler
    if (command == 1) //creator
    {
        if (makers.Count < 5) //checking count of active machines
        {
            Console.WriteLine("Do you want to specify brand? (1- y, 2- n)");
            int ans = Convert.ToInt32(Console.ReadLine());
            if (ans == 1)
            {
                Console.WriteLine("Write it:");
                string brand = Console.ReadLine();

                Console.WriteLine("Do you want to specify model? (1- y, 2- n)");
                ans = Convert.ToInt32(Console.ReadLine());
                if (ans == 1)
                {
                    Console.WriteLine("Write it:");
                    string model = Console.ReadLine();
                    makers.Add(new YogurtMaker(brand, model));
                }
                else
                    makers.Add(new YogurtMaker(brand));
            }
            else
                makers.Add(new YogurtMaker());

            Console.WriteLine("Success!");
        }
        else
    }

```

```

        Console.WriteLine("You already have maximum amount of machines!
<-----");
    }
    else if (command == 2) //deleting
    {
        Console.WriteLine("Type ID of YogurtMaker you want to move to trash:");
        int toRemove = Convert.ToInt32(Console.ReadLine());

        bool flag = true;
        for (int i = 0; i < makers.Count; i++)
        {
            if (makers[i].Id == toRemove)
            {
                makers.RemoveAt(toRemove - 1);
                Console.WriteLine("Success!");
                flag = !flag;
            }
        }
        if (flag)
        {
            Console.ForegroundColor = ConsoleColor.Red;
            Console.WriteLine("Something weng wrong...");
        }
    }
    else if (command == 3) // on/off switch
    {
        Console.WriteLine("Type ID of YogurtMaker you want to turn off/on
(triggering):");
        int toTurn = Convert.ToInt32(Console.ReadLine());
        bool flag = true;
        for (int i = 0; i < makers.Count; i++)
        {
            if (makers[i].Id == toTurn)
            {
                makers[i].OffOn();
                Console.WriteLine("Success!");
                flag = !flag;
            }
        }
        if (flag)
        {
            Console.ForegroundColor = ConsoleColor.Red;
            Console.WriteLine("Something weng wrong...");
        }
    }
    else if (command == 4) // working with specified machine

```

picker

```
{
    Console.WriteLine("Type ID of YogurtMaker you want to work with:"); // machine
    int toTurn = Convert.ToInt32(Console.ReadLine());
    bool flag = true;
    for (int i = 0; i < makers.Count; i++)
    {
        if (makers[i].Id == toTurn)
        {
            if (makers[i].PowerOn == 0)
                makers[i].OffOn();

            // modes menu
            Console.WriteLine("Choose what to do:\n" +
                "1 - yogurt;\n" +
                "2 - cheese;\n" +
                "3 - sourCream;\n" +
                "4 - curd");
            int mode = Convert.ToInt32(Console.ReadLine());

            makers[i].ModeSet(mode);
            Console.WriteLine("Success!");
            flag = !flag;
        }
    }
    if (flag)
    {
        Console.ForegroundColor = ConsoleColor.Red;
        Console.WriteLine("Something weng wrong...");
    }
}
else if (command == 5) // brand/model changer
{
    Console.WriteLine("Type ID of YogurtMaker you want to work with:");
    int toTurn = Convert.ToInt32(Console.ReadLine());
    bool flag = true;
    for (int i = 0; i < makers.Count; i++)
    {
        if (makers[i].Id == toTurn)
        {
            Console.WriteLine("Success!");
            flag = !flag;
            Console.WriteLine("Choose what to do:\n" +
                "1 - Change brand;\n" +
                "2 - Change model.");
        }
    }
}
```



```
List of your YogurtMakers:
-----
List of commands:
1 - create new YogurtMaker (max 5);
2 - delete specified machine;
3 - turn off/on specified machine;
4 - change mode of specified machine.
Type number of command you want to run:
1
Do you want to specify brand? (1- y, 2- n)
1
Write it:
TestBrand
Do you want to specify model? (1- y, 2- n)
1
Write it:
TestModel
Success!
List of your YogurtMakers:
-----
ID: 1
Mode: 0
Power: 0
Brand: TestBrand
Model: TestModel
-----
List of commands:
1 - create new YogurtMaker (max 5);
2 - delete specified machine;
3 - turn off/on specified machine;
4 - change mode of specified machine.
Type number of command you want to run:
_
```

```
List of commands:
1 - create new YogurtMaker (max 5);
2 - delete specified machine;
3 - turn off/on specified machine;
4 - change mode of specified machine.
Type number of command you want to run:
1
Do you want to specify brand? (1- y, 2- n)
2
Success!
List of your YogurtMakers:
-----
ID: 1
Mode: 0
Power: 0
Brand: TestBrand
Model: TestModel
-----
ID: 2
Mode: 0
Power: 0
Brand: NoName
Model: Unknown
-----
List of commands:
1 - create new YogurtMaker (max 5);
2 - delete specified machine;
3 - turn off/on specified machine;
4 - change mode of specified machine.
Type number of command you want to run:
```

```
List of commands:
1 - create new YogurtMaker (max 5);
2 - delete specified machine;
3 - turn off/on specified machine;
4 - change mode of specified machine.
Type number of command you want to run:
2
Type ID of YogurtMaker you want to move to trash:
1
Success!
List of your YogurtMakers:
-----
ID: 2
Mode: 0
Power: 0
Brand: NoName
Model: Unknown
-----
List of commands:
1 - create new YogurtMaker (max 5);
2 - delete specified machine;
3 - turn off/on specified machine;
4 - change mode of specified machine.
Type number of command you want to run:
_
```

```
List of commands:
1 - create new YogurtMaker (max 5);
2 - delete specified machine;
3 - turn off/on specified machine;
4 - change mode of specified machine.
Type number of command you want to run:
1
You already have maximum amount of machines! <-----
List of your YogurtMakers:
-----
ID: 2
Mode: 0
Power: 0
Brand: NoName
Model: Unknown
-----
ID: 3
Mode: 0
Power: 0
Brand: NoName
Model: Unknown
-----
ID: 4
Mode: 0
Power: 0
Brand: NoName
Model: Unknown
-----
ID: 5
Mode: 0
Power: 0
Brand: NoName
Model: Unknown
-----
ID: 6
Mode: 0
Power: 0
Brand: NoName
Model: Unknown
-----
List of commands:
1 - create new YogurtMaker (max 5);
2 - delete specified machine;
3 - turn off/on specified machine;
4 - change mode of specified machine.
Type number of command you want to run:

```

```
List of commands:
1 - create new YogurtMaker (max 5);
2 - delete specified machine;
3 - turn off/on specified machine;
4 - change mode of specified machine.
Type number of command you want to run:
3
Type ID of YogurtMaker you want to turn off/on (triggering):
2
Success!
List of your YogurtMakers:
-----
ID: 2
Mode: 0
Power: 1
Brand: NoName
Model: Unknown
-----
ID: 3
Mode: 0
Power: 0
Brand: NoName
Model: Unknown
-----
ID: 4
Mode: 0
Power: 0
Brand: NoName
Model: Unknown
-----
ID: 5
Mode: 0
Power: 0
Brand: NoName
Model: Unknown
-----
ID: 6
Mode: 0
Power: 0
Brand: NoName
Model: Unknown
-----
List of commands:
1 - create new YogurtMaker (max 5);
2 - delete specified machine;
3 - turn off/on specified machine;
4 - change mode of specified machine.
Type number of command you want to run:

```



```
List of commands:
1 - create new YogurtMaker (max 5);
2 - delete specified machine;
3 - turn off/on specified machine;
4 - change mode of specified machine.
Type number of command you want to run:
4
Type ID of YogurtMaker you want to work with:
2
Choose what to do:
1 - yogurt;
2 - cheese;
3 - sourCream;
4 - curd
4
Success!
List of your YogurtMakers:
-----
ID: 2
Mode: 4
Power: 1
Brand: NoName
Model: Unknown
-----
ID: 3
Mode: 0
Power: 0
Brand: NoName
Model: Unknown
-----
ID: 4
Mode: 0
Power: 0
Brand: NoName
Model: Unknown
-----
ID: 5
Mode: 0
Power: 0
Brand: NoName
Model: Unknown
-----
ID: 6
Mode: 0
Power: 0
Brand: NoName
Model: Unknown
-----
List of commands:
1 - create new YogurtMaker (max 5);
2 - delete specified machine;
3 - turn off/on specified machine;
4 - change mode of specified machine.
Type number of command you want to run:
```

Вывод: Я разработал копию абстрактной йогуртницы, описал ее работу при помощи классов, а также реализовал взаимодействие пользователя с ней. Также улучшил свои навыки по работы с ООП и C# в частности, и по работе в IDE Visual Studio 2019.