

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

ФАКУЛЬТЕТ ЭЛЕКТРОННО-ИНФОРМАЦИОННЫХ СИСТЕМ

Кафедра интеллектуальных информационных технологий

Отчет по лабораторной работе №4

Выполнила
Радиончик С.С.,
студентка группы ПО-5

Проверил
Крощенко А.А.,
ст. преп. Кафедры ИИТ,
«__» _____ 2021 г.

Брест, 2021

Цель работы: приобрести практические навыки в области объектно-ориентированного программирования.

Вариант 6.

Задание 1. Реализовать указанный класс, включив в него вспомогательный внутренний класс или классы. Реализовать 2-3 метода (на выбор).

Продemonстрировать использование реализованных классов.

*Создать класс **Catalog** (каталог) с внутренним классом, с помощью объектов которого можно хранить информацию об истории выдач книги читателям.*

Задание 2. Реализовать агрегирование. При создании класса агрегируемый класс объявляется как атрибут (локальная переменная, параметр метода).

Включить в каждый класс 2-3 метода на выбор. Продemonстрировать использование разработанных классов.

*Создать класс **Страница**, используя класс **Слово**.*

Задание 3. Построить модель программной системы с применением отношений (обобщения, агрегации, ассоциации, реализации) между классами. Задать атрибуты и методы классов. Реализовать (если необходимо) дополнительные классы. Продemonстрировать работу разработанной системы.

*Система **Телефонная станция**. Абонент оплачивает **Счет** за разговоры и **Услуги**, может попросить **Администратора** сменить номер и отказаться от услуг. **Администратор** изменяет номер, **Услуги** и временно отключает **Абонента** за неуплату.*

Выполнение:

Код программы.

1)

Catalog.cs

```
using System;
using System.Collections.Generic;

namespace Lab4_1
{
    class Catalog
    {
        public Dictionary<string, List<BookTracker>> Book;

        public Catalog()
        {
            Book = new Dictionary<string, List<BookTracker>>();
        }

        public void AddBook(string nameBook)
        {
            Book.Add(nameBook, new List<BookTracker>());
        }

        public void AddBookInfo(string bookTitle, BookTracker info)
        {
            foreach(var book in Book)
```

```

        {
            if(book.Key == bookTitle)
            {
                book.Value.Add(info);
            }
        }
    }

    public void List()
    {
        foreach (var book in Book)
        {
            Console.WriteLine($"{book.Key}:");
            var bookInfo = book.Value;
            foreach(var record in bookInfo)
                Console.WriteLine(record.GetRecord());
        }
    }

    public void ListOfReader(string bookTitle)
    {
        foreach (var book in Book)
        {
            if(book.Key == bookTitle)
            {
                Console.WriteLine($"{book.Key}:");
                var bookInfo = book.Value;
                foreach (var record in bookInfo)
                    Console.WriteLine(record.GetReader());
            }
        }
    }

    public class BookTracker
    {
        public string Reader { get; set; }
        public string Period { get; set; }

        public BookTracker(string reader, string period)
        {
            Reader = reader;
            Period = period;
        }

        public string GetRecord()
        {
            BookTracker bookTracker = this;
            return $"Reader: {Reader} Period: {bookTracker.Period}";
        }

        public string GetReader()
        {
            return $"{Reader}";
        }
    }
}

```

Program.cs

```

namespace Lab4_1
{
    class Program
    {
        static void Main(string[] args)

```

```

{
    Catalog catalog = new Catalog();

    Catalog.BookTracker LostInfo1 = new Catalog.BookTracker("Vakham V.N",
"12.10.2019 - 12.02.2020");
    Catalog.BookTracker LostInfo2 = new Catalog.BookTracker("Grenz K.L",
"21.09.2020 - 12.02.2021");
    Catalog.BookTracker LostInfo3 = new Catalog.BookTracker("Hall R.N",
"03.03.2021 - 12.03.2021");
    Catalog.BookTracker CinderellaInfo1 = new Catalog.BookTracker("Solla H.F",
"10.10.2020 - 12.12.2021");

    catalog.AddBook("Cinderella");
    catalog.AddBook("Lost");

    catalog.AddBookInfo("Lost", LostInfo1);
    catalog.AddBookInfo("Lost", LostInfo2);
    catalog.AddBookInfo("Lost", LostInfo3);
    catalog.AddBookInfo("Cinderella", CinderellaInfo1);

    catalog.List();
    catalog.ListOfReader("Lost");
}
}

```

2)

Word.cs

namespace Lab4_2

```

{
    class Word
    {
        public string NewWord { get; set; }

        public Word(string newWord)
        {
            NewWord = newWord;
        }

        public string GetWord()
        {
            return NewWord;
        }
    }
}

```

Page.cs

using System;

using System.Collections.Generic;

namespace Lab4_2

```

{
    class Page
    {
        public List<Word> TextPage;

        public Page()
        {
            TextPage = new List<Word>();
        }

        public void AddWord(Word word)
        {

```

```

        TextPage.Add(word);
    }

    public void GetPage()
    {
        foreach(var word in TextPage)
        {
            Console.Write($"{word.GetWord()} ");
        }
    }
}

```

Program.cs

```

namespace Lab4_2
{
    class Program
    {
        static void Main(string[] args)
        {
            Page page = new Page();

            page.AddWord(new Word("My"));
            page.AddWord(new Word("family"));
            page.AddWord(new Word("is"));
            page.AddWord(new Word("very"));
            page.AddWord(new Word("friendly."));

            page.GetPage();
        }
    }
}

```

3)

Bill.cs

```

namespace Lab4_3
{
    class Bill
    {
        public string Number { get; set; }
        public bool IsPaid { get; set; }
    }
}

```

Service.cs

```

using System.Collections.Generic;

namespace Lab4_3
{
    class Service
    {
        public List<string> RefusalService = new List<string>();

        public bool MMS { get; set; }
        public bool SMS { get; set; }
        public bool Internet { get; set; }

        public void setRefusalServiceList(string service)
        {
            RefusalService.Add(service);
        }
    }
}

```

```
}
```

SystemUser.cs

```
namespace Lab4_3
{
    class SystemUser
    {
        public string Name { get; set; }
        public UserPosition Position { get; set; }

        public enum UserPosition
        {
            Abonent,
            Administrator
        }

        public SystemUser()
        {
            Name = null;
            Position = 0;
        }

        public SystemUser(string name, UserPosition position)
        {
            Name = name;
            Position = position;
        }

        public string GetData()
        {
            return $"Name: {Name}\n Position: {nameof(Position)}";
        }
    }
}
```

Abonent.cs

```
using System;

namespace Lab4_3
{
    class Abonent : SystemUser
    {
        public string Number { get; set; }
        public bool IsBlocked { get; set; }
        public Service Services = new Service();
        public bool IsPhoneChangeRequest { get; set; }
        public Bill Bill = new Bill();
        public bool IsServiceRefusalRequest { get; set; }

        public Abonent(string number, string billNumber)
        {
            Number = number;
            IsBlocked = false;
            Services.SMS = true;
            Services.MMS = true;
            Services.Internet = true;
            IsPhoneChangeRequest = false;
            Bill.Number = billNumber;
            Bill.IsPayed = false;
            IsServiceRefusalRequest = false;
        }

        public void PayedBill()
        {
        }
    }
}
```

```

    {
        Bill.IsPaid = true;
        IsBlocked = false;
    }

    public void PhoneChangeRequest()
    {
        IsPhoneChangeRequest = true;
    }

    public void ServiceRefusalRequest(string service)
    {
        IsServiceRefusalRequest = true;
        Services.setRefusalSeviceList(service);
    }

    new public void GetData()
    {
        Console.WriteLine("Information:");
        Console.WriteLine($"Bill: {Bill.Number}");
        Console.WriteLine(Number);
        Console.WriteLine($"Blocked: {IsBlocked}");
        Console.WriteLine($"The bill is paid: {Bill.IsPaid }");
        Console.WriteLine($"Services: SMS - {Services.SMS}, MMS - {Services.MMS},
Internet - {Services.Internet}\n");
    }
}

```

Administrator.cs

```

using System;
using System.Collections.Generic;

namespace Lab4_3
{
    class Administrator : SystemUser
    {
        public List<Abonent> Abonents = new List<Abonent>();

        public void AddAbonent(Abonent abonent)
        {
            Abonents.Add(abonent);
        }

        public void ChangePhoneNumber(Abonent abonent, string number)
        {
            if (Abonents.Contains(abonent))
            {
                if (abonent.IsPhoneChangeRequest)
                    abonent.Number = number;
                else
                    Console.WriteLine("No such request");
            }
            else
                Console.WriteLine("Error. Undefined abonent");
        }

        public void ServiceRefusal(Abonent abonent)
        {
            if (Abonents.Contains(abonent))
            {
                if (abonent.IsServiceRefusalRequest)
                {
                    foreach (var servcive in abonent.Services.RefusalService)

```

```

        {
            if (sercvce == "MMS")
                abonent.Services.MMS = false;
            else if (sercvce == "SMS")
                abonent.Services.SMS = false;
            else if (sercvce == "Internet")
                abonent.Services.Internet = false;
            else
                Console.WriteLine("Error. Undefined service");
        }
    }
    else
        Console.WriteLine("No such request");
}
else
    Console.WriteLine("Error. Undefined abonent");
}

public void Block(Abonent abonent)
{
    if (!abonent.Bill.IsPayed)
        abonent.IsBlocked = true;
}
}
}

```

Program.cs

```

namespace Lab4_3
{
    class Program
    {
        static void Main(string[] args)
        {
            Abonent abonent = new Abonent("+375334579865", "001");
            Administrator admin = new Administrator();

            abonent.GetData();
            abonent.PhoneChangeRequest();
            abonent.ServiceRefusalRequest("MMS");

            admin.AddAbonent(abonent);
            admin.ChangePhoneNumber(abonent, "+375339043245");
            admin.ServiceRefusal(abonent);
            abonent.GetData();
            admin.Block(abonent);

            abonent.GetData();
            abonent.PayedBill();
            abonent.GetData();
        }
    }
}

```


Результаты работы программы:

1)

 Консоль отладки Microsoft Visual Studio

Cinderella:

Reader: Solla H.F Period: 10.10.2020 - 12.12.2021

Lost:

Reader: Vakham V.N Period: 12.10.2019 - 12.02.2020

Reader: Grenz K.L Period: 21.09.2020 - 12.02.2021

Reader: Hall R.N Period: 03.03.2021 - 12.03.2021


Lost:

Vakham V.N

Grenz K.L

Hall R.N

2)

 Консоль отладки Microsoft Visual Studio

My family is very friendly.

3)

 Консоль отладки Microsoft Visual Studio

Information:

Bill: 001

+375334579865

Blocked: False

The bill is paid: False

Services: SMS - True, MMS - True, Internet - True

Information:

Bill: 001

+375339043245

Blocked: False

The bill is paid: False

Services: SMS - True, MMS - False, Internet - True

Information:

Bill: 001

+375339043245

Blocked: True

The bill is paid: False

Services: SMS - True, MMS - False, Internet - True

Information:

Bill: 001

+375339043245

Blocked: False

The bill is paid: True

Services: SMS - True, MMS - False, Internet - True

Вывод: приобрела практические навыки в области объектно-ориентированного программирования.