Министерство науки и высшего образования Российской Федерации

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

«Владимирский государственный университет

имени Александра Григорьевича и Николая Григорьевича Столетовых»

(ВлГУ)

Кафедра информационных систем и программной инженерии

**Лабораторная работа №6**

**по дисциплине**

**«Технологии программирования»**

**РАСШИРЕННЫЕ ВОЗМОЖНОСТИ C#. ДЕЛЕГАТЫ**

**Выполнил**:

ст. гр. ПРИ-120

Д. А. Грачев

**Принял**:

Данилов В. В.

Владимир, 2021

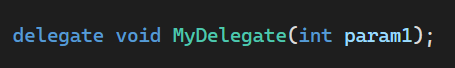
ЦЕЛЬ РАБОТЫ

Познакомиться с возможностями языка программирования C# для организации вызова методов с использованием делегатов.

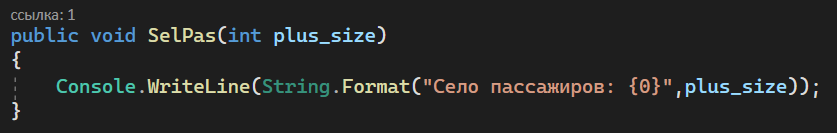
ВЫПОЛНЕНИЕ РАБОТЫ

**Предметная область:** «Морской грузовой порт» (примерные возможные объекты: порты, судна грузовые, пассажирские, речные объекты, грузы сотрудники, заказы на перевозку и т.д.);

1. Был создан делегат MyDelegate



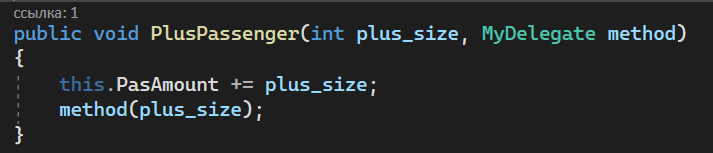
1. Создан метод по делегату



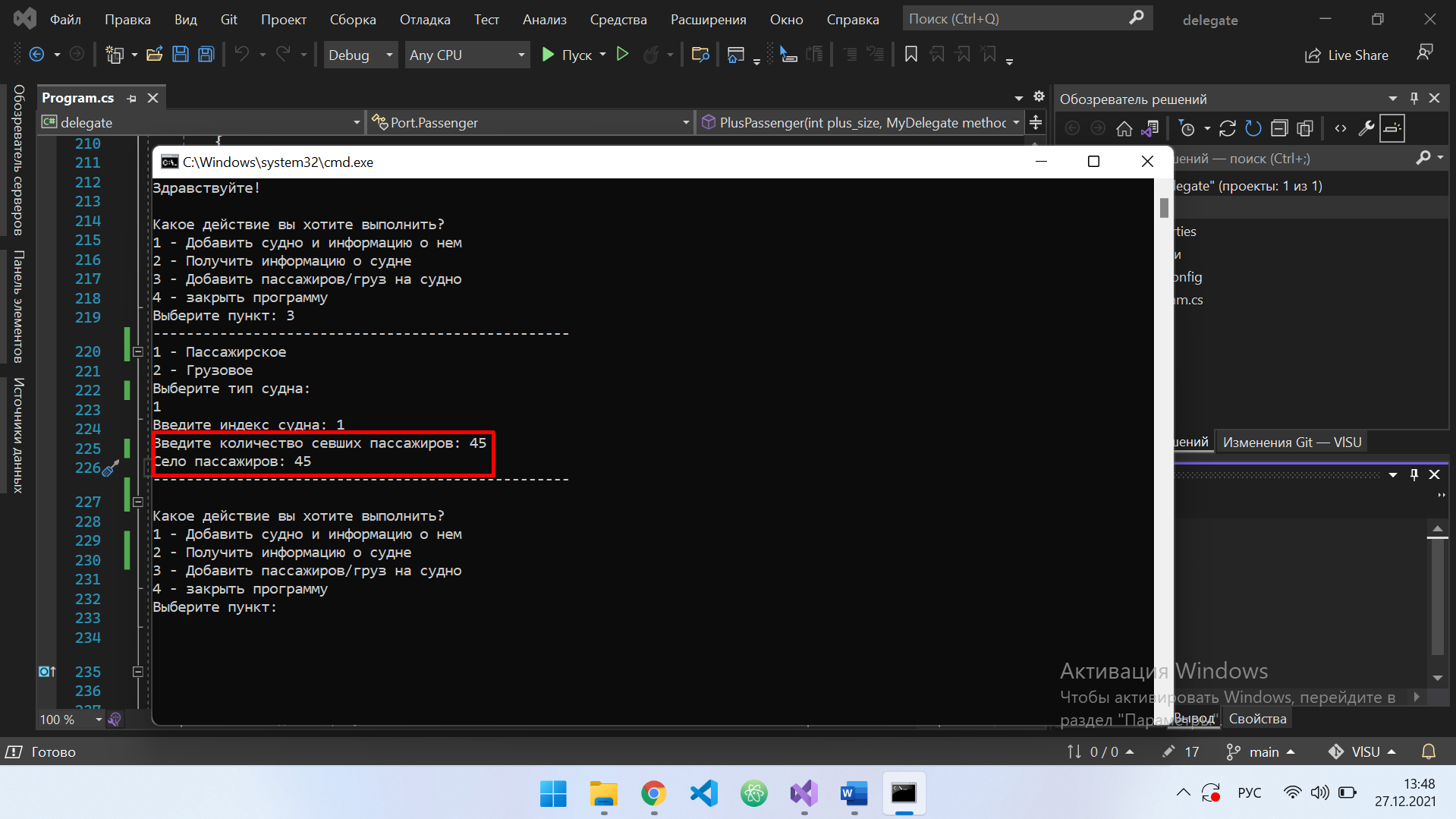
1. Создана переменная типа MyDelegate и присвоен ей метод



1. В метод PlusPаssenger был добавлен вызов этого метода



Скриншот работы программы представлен ниже



Листинг программы:

using System;

namespace Port

{

    delegate void MyDelegate(int param1);

    class Program

    {

        static void Main(string[] args)

        {

            Passenger[] SpisokPS = new Passenger[7];

            Gruzovoi[] SpisokGR = new Gruzovoi[7];

            SpisokPS[0] = new Passenger("Бригантина", 520, 1540000, 330);

            SpisokPS[1] = new Passenger("Победа", 100, 120000, 100);

            SpisokPS[2] = new Passenger("Беда", 10, 15400, 0);

            byte LastIndPS = 3, LastIndGR = 0;

            Console.WriteLine("Здравствуйте!");

            while (true)

            {

                Console.WriteLine();

                Console.WriteLine("Какое действие вы хотите выполнить?");

                Console.WriteLine("1 - Добавить судно и информацию о нем");

                Console.WriteLine("2 - Получить информацию о судне");

                Console.WriteLine("3 - Добавить пассажиров/груз на судно");

                Console.WriteLine("4 - закрыть программу");

                Console.Write("Выберите пункт: ");

                byte key = Convert.ToByte(Console.ReadLine());

                switch (key)

                {

                    case 1: *//Добавление судна*

                        Console.WriteLine("--------------------------------------------------");

                        Console.WriteLine("1 - Пассажирское ");

                        Console.WriteLine("2 - Грузовое ");

                        Console.WriteLine("Выберите тип судна: ");

                        key = Convert.ToByte(Console.ReadLine());

                        if (key == 1)*// Добавление Пассажирского*

                        {

                            if (LastIndPS == 6)

                            {

                                Console.WriteLine("НЕ ОСТАЛОСЬ МЕСТА В ПОРТУ");

                                break;

                            }

                            Console.WriteLine("--------------------------------------------------");

                            Console.Write("Введите название судна: ");

                            string name = Console.ReadLine();

                            Console.Write("Введите вместимость судна: ");

                            int vmest = Convert.ToInt32(Console.ReadLine());

                            Console.Write("Введите вес судна: ");

                            int ves = Convert.ToInt32(Console.ReadLine());

                            Console.Write("Введите кол-во людей на судне: ");

                            int pasamount = Convert.ToInt32(Console.ReadLine());

                            SpisokPS[LastIndPS] = new Passenger(name, vmest, ves, pasamount);

                            Console.WriteLine("Было создано пассажирское судно с индексом - " + LastIndPS);

                            LastIndPS++;

                            Console.WriteLine("--------------------------------------------------");

                        }

                        else if (key == 2) *//Добавление грузового*

                        {

                            if (LastIndGR == 6)

                            {

                                Console.WriteLine("НЕ ОСТАЛОСЬ МЕСТА В ПОРТУ");

                                break;

                            }

                            Console.WriteLine("--------------------------------------------------");

                            Console.Write("Введите название судна: ");

                            string name = Console.ReadLine();

                            Console.Write("Введите вместимость судна: ");

                            int vmest = Convert.ToInt32(Console.ReadLine());

                            Console.Write("Введите вес судна: ");

                            int ves = Convert.ToInt32(Console.ReadLine());

                            Console.Write("Введите кол-во груза на судне: ");

                            int gruzamount = Convert.ToInt32(Console.ReadLine());

                            SpisokGR[LastIndGR] = new Gruzovoi(name, vmest, ves, gruzamount);

                            Console.WriteLine("Было создано грузовое судно с индексом - " + LastIndGR);

                            LastIndPS++;

                            Console.WriteLine("--------------------------------------------------");

                        }

                        else

                        {

                            Console.WriteLine("Нет такого судна");

                        }

                        break;

                    case 2: *//Информация о судне по индексу*

                        Console.WriteLine("--------------------------------------------------");

                        Console.WriteLine("1 - Пассажирское ");

                        Console.WriteLine("2 - Грузовое ");

                        Console.WriteLine("Выберите тип судна: ");

                        key = Convert.ToByte(Console.ReadLine());

                        byte ind2;

                        switch (key)

                        {

                            case 1:

                                Console.Write("Введите индекс судна: ");

                                ind2 = Convert.ToByte(Console.ReadLine());

                                SpisokPS[ind2].PrintInfo();

                                break;

                            case 2:

                                Console.Write("Введите индекс судна: ");

                                ind2 = Convert.ToByte(Console.ReadLine());

                                SpisokGR[ind2].PrintInfo();

                                break;

                        }

                        break;

                    case 3: *//Добавление пассажиров/груза*

                        Console.WriteLine("--------------------------------------------------");

                        Console.WriteLine("1 - Пассажирское ");

                        Console.WriteLine("2 - Грузовое ");

                        Console.WriteLine("Выберите тип судна: ");

                        key = Convert.ToByte(Console.ReadLine());

                        byte ind3;

                        int plssize;

                        switch (key)

                        {

                            case 1:

                                Console.Write("Введите индекс судна: ");

                                ind3 = Convert.ToByte(Console.ReadLine());

                                Console.Write("Введите количество севших пассажиров: ");

                                plssize = Convert.ToInt32(Console.ReadLine());

                                MyDelegate method = SpisokPS[ind3].SelPas;

                                SpisokPS[ind3].PlusPassenger(plssize,method);

                                break;

                            case 2:

                                Console.Write("Введите индекс судна: ");

                                ind3 = Convert.ToByte(Console.ReadLine());

                                Console.Write("Введите количество добавленного груза: ");

                                plssize = Convert.ToInt32(Console.ReadLine());

                                SpisokGR[ind3].PlusGruz(plssize);

                                break;

                        }

                        Console.WriteLine("--------------------------------------------------");

                        break;

                    case 4: *//Выход*

                        Environment.Exit(0);

                        break;

                }

            }

        }

    }

    abstract class Sudno

    {

        protected string Name;

        protected int Vmest;

        protected int Ves;

        protected Sudno(string name, int vmest, int ves)

        {

            this.Name = name;

            this.Vmest = vmest;

            this.Ves = ves;

        }

        public abstract void PrintInfo();

    }

    class Passenger : Sudno

    {

        private int PasAmount;

        public Passenger(string name, int vmest, int ves, int pasamount) : base(name, vmest, ves)

        {

            this.PasAmount = pasamount;

        }

        private void Zagruzhen()

        {

            if (PasAmount == Vmest)

            {

                Console.WriteLine("{0} - Уже заполнено ", Name);

            }

            else

            {

                Console.WriteLine("{0} - Не заполнено или переполнено ", Name);

            }

        }

        public override void PrintInfo()

        {

            Console.WriteLine("--------------------------------------------------");

            Console.WriteLine("Название судна : " + Name);

            Console.WriteLine("Вместимость людей : " + Vmest);

            Console.WriteLine("Вес судна : " + Ves);

            Console.WriteLine("Находящиеся на борту люди : " + PasAmount);

            Zagruzhen();

            Console.WriteLine("--------------------------------------------------");

        }

        public void SelPas(int plus\_size)

        {

            Console.WriteLine(String.Format("Село пассажиров: {0}",plus\_size));

        }

        public void PlusPassenger(int plus\_size, MyDelegate method)

        {

            this.PasAmount += plus\_size;

            method(plus\_size);

        }

    }

    class Gruzovoi : Sudno

    {

        private int GruzAmount;

        public Gruzovoi(string name, int vmest, int ves, int gruzamount) : base(name, vmest, ves)

        {

            this.GruzAmount = gruzamount;

        }

        public override void PrintInfo()

        {

            Console.WriteLine("--------------------------------------------------");

            Console.WriteLine("Название судна : " + Name);

            Console.WriteLine("Вместимость людей : " + Vmest);

            Console.WriteLine("Вес судна : " + Ves);

            Console.WriteLine("Количество груза на борту : " + GruzAmount);

            Console.WriteLine("--------------------------------------------------");

        }

        public void PlusGruz(int plus\_size)

        {

            this.GruzAmount += plus\_size;

        }

    }

}

ВЫВОД

В ходе выполнения лабораторной работы познакомился с возможностями языка программирования C# для организации вызова методов с использованием делегатов.