Лабораторная работа №4

Цель работы:

* Понятие Класса
* Создание простого Класса
* Ввод и вывод на Консоль

#Address.cs

Практическая работа №7-9

using System;

using System.Collections.Generic;

using System.Text;

namespace Constructor\_Class

{

class Address

{

public string Full\_Name;

public string Country;

public string Region;

public string Town;

public string Street;

public int Apartment = 0;

public int Room = 0;

public Address(string Full\_Name = "", string Country = "",

string Region = "", string Town = "", string Street = "", int Apartment = 0, int Room = 0)

{

this.Full\_Name = Full\_Name;

this.Country = Country;

this.Region = Region;

this.Town = Town;

this.Apartment = Apartment;

this.Room = Room;

}

}

}

#Program.cs

using System;

namespace Constructor\_Class

{

class Program

{

static void Main(string[] args)

{

Address adr1;

adr1 = new Address();

adr1.Full\_Name = "Иванов И.И.";

adr1.Country = "Россия";

adr1.Region = "С3";

adr1.Town = "Псков";

adr1.Street = "Ленина";

adr1.Apartment = 2;

adr1.Room = 101;

Address adr2 = new Address("Петров П.П.", "Беларусь", "Ц", "Минск", "Ленина", 2, 101);

Console.WriteLine($"Адрес Иванова И.И.: \n" +

$"{adr1.Country}, {adr1.Region} регион, город {adr1.Town}, улица {adr1.Street}, " +

$"д. {adr1.Apartment}, кв. {adr1.Room}");

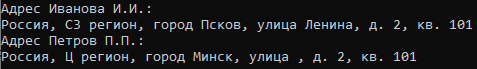
Console.WriteLine($"Адрес Петров П.П.: \n" +

$"{adr1.Country}, {adr2.Region} регион, город {adr2.Town}, улица {adr2.Street}, " +

$"д. {adr2.Apartment}, кв. {adr2.Room}");

}

}

}

Вывод 1:

В программе “Address.cs”, создаётся класс “Address”, в котором объявляются глобальные переменные, иначе именуемые “полями”. Затем используется конструктор “public Address”, в качестве параметров передаются поля, с пустым присваиванием (пустая строка, ноль). Внутри тела конструктора, используется метод “this”, который принимает значения параметров внутри ДАННОГО конструктора, затем присваивает полям (переменным) параметры.

Вывод 2:

В программе “Program.cs”, определяется тип переменной “adr1”, затем создаётся экземпляр класса “adr1”, дальше присваиваются значения, к этому экземпляру класса, по методам (полям которым мы объявили в классе). Тоже самое происходит с переменной “adr2”, но вместо многострочного присваивания значений к каждому экземпляру класса, мы используем конструктор, который создали в “Address.cs”, и присваиваем значения в ПОРЯДКЕ ОБЪЯВЛЕНИЯ в классе.

Практическая работа №8

using System;

class Tel

{

public string Full\_Name = "";

public int number = 0;

}

namespace Telephone\_Book

{

class Program

{

static void Main(string[] args)

{

Tel t1;

t1 = new Tel();

t1.Full\_Name = "Иванов";

t1.number = 111213;

int y1 = t1.number / 10000;

if (y1 == 33)

{

Console.WriteLine($"{t1.Full\_Name}, {t1.number}");

}

Tel t2;

t2 = new Tel();

t2.Full\_Name = "Петров";

t2.number = 222324;

int y2 = t2.number / 10000;

if(y2 == 33)

{

Console.WriteLine($"{t2.Full\_Name}, {t2.number}");

}

Tel t3;

t3 = new Tel();

t3.Full\_Name = "Сидоров";

t3.number = 333435;

int y3 = t3.number / 10000;

if (y3 == 33)

{

Console.WriteLine($"{t3.Full\_Name}, {t3.number}");

}

Tel t4;

t4 = new Tel();

t4.Full\_Name = "Никитин";

t4.number = 332324;

int y4 = t4.number / 10000;

if (y4 == 33)

{

Console.WriteLine($"{t4.Full\_Name}, {t4.number}");

}

}

 }

}

Вывод 1:

Создаём класс “Tel” с полями “Full\_Name”, “number”.

Вывод 2:

Определяем тип переменной “t1”, создаём экземпляр класса “t1”, присваиваем значения полей экземпляра “t1” класса “Tel”. Объявляем переменную “y2”, которой присваиваем значение первых двух цифр числа “t2.number”, деля число на 10000. Проверяем равенство: (22 == 33)? => False, иначе выписали бы Фамилию и номер “t2”. По такому же принципу определяем типы новых переменных, создаём экземпляры, проверяем условие.

Вторая часть Практического задания №8

Изменения:

20:25 - 20:26;

t1.number = 111213;

t1.number = 771213;

22:24 – 22:25;

if (y1 == 33)

if (y1 == 77)

31:31;

t2.number = 222324;

t2.number = 222325;

Третья часть Практического задания №8

Изменения:

32:35 - 32:39

int y2 = t2.number / 10000;

int y2 = t2.number % 10;

34:23 - 34:24

if(y2 == 33)

if(y2 == 5)

Общий вывод после изменений:



Практическое задание №9

#SqEq.cs

using System;

using System.Collections.Generic;

using System.Security.Cryptography.X509Certificates;

using System.Text;

namespace Squared\_equations

{

class SqEq

{

private double a, b, c;

public SqEq(double a, double b, double c)

{

this.a = a;

this.b = b;

this.c = c;

}

private double Discr()

{

return Math.Pow(b, 2) - 4 \* a \* c;

}

public int CountK()

{

double d = Discr();

if (d > 0)

{

return 2;

}

else if(d == 0){

return 1;

}

else

{

return 0;

}

}

public string Out()

{

int n = CountK();

if (n == 0)

{

return "Нет корней";

}

else if (a == 0)

{

return "Вырожденное квадратное уравение";

}

else if (n == 1)

{

return $"x = {-b / 2 / a}";

}

else

{

double d = Math.Sqrt(Discr());

double x1 = (-b + d) / 2 / a;

double x2 = (-b - d) / 2 / a;

return $"x1 = {x1} \nx2 = {x2}";

}

}

}

}

#Program.cs

using System;

namespace Squared\_equations

{

class Program

{

static void Main(string[] args)

{

double a, b, c;

Console.WriteLine("a = ");

a = double.Parse(Console.ReadLine());

Console.WriteLine("b = ");

b = double.Parse(Console.ReadLine());

Console.WriteLine("c = ");

c = double.Parse(Console.ReadLine());

SqEq equation = new SqEq(a, b, c);

Console.WriteLine($"Количество решений уравнения {equation.CountK()}");

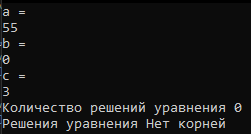
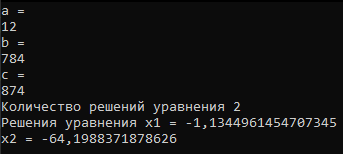
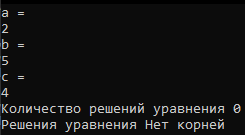
Console.WriteLine($"Решения уравнения {equation.Out()}");

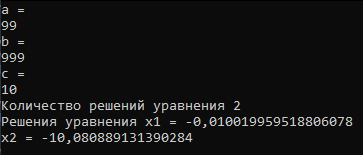
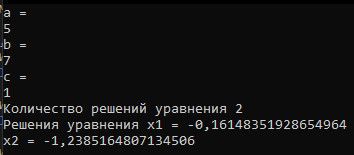
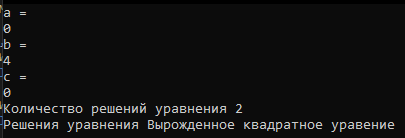
Console.ReadLine();

}

}

}





Вывод 1: Использованы те же методы при объявлении типов переменных, создание экземпляров классов, решение уравнений.

Практическая работа №10

using System;

namespace Try\_catch

{

class Program

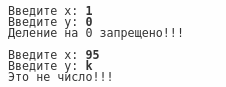
{

static void Main(string[] args)

{

link1:

try

{

Console.Write("Введите x: ");

int x = int.Parse(Console.ReadLine());

Console.Write("Введите x: ");

int y = int.Parse(Console.ReadLine());

int result = x / y;

Console.WriteLine($"Результат {result}");

}

catch (DivideByZeroException)

{

Console.WriteLine("Деление на 0 запрещено!!!\n");

goto link1;

}

catch (FormatException)

{

Console.WriteLine("Это не число!!!\n");

goto link1;

}

}

}

}

Практическая работа №11

#Person.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Security.Cryptography.X509Certificates;

using System.Text;

namespace Expections

{

class Person

{

private string Surname;

private string Name;

private System.DateTime dr;

private double Height;

private int Weight;

public Person()

{

Console.Write("Фамилия ");

Surname = Console.ReadLine();

Console.Write("Имя ");

Name = Console.ReadLine();

Console.Write("Дата рождения ");

try

{

string s = Console.ReadLine();

dr = Convert.ToDateTime(s);

}

catch (FormatException)

{

dr = Convert.ToDateTime("01/01/2001");

}

Console.Write("Рост, м ");

try

{

string s = Console.ReadLine();

Height = Convert.ToDouble(s);

}

catch (FormatException)

{

Height = 1.6;

}

Console.Write("Вес, кг");

try

{

string s = Console.ReadLine();

Weight = Convert.ToInt32(s);

}

catch (FormatException)

{

Weight = 50;

}

}

public void Output()

{

Console.Clear();

Console.WriteLine($"Фамилия {this.Surname}");

Console.WriteLine($"Имя {this.Name}");

Console.WriteLine($"Дата рождения {this.dr}");

Console.WriteLine($"Рост {this.Height}");

Console.WriteLine($"Вес {this.Weight}");

}

}

}

#Program.cs

using System;

namespace Expections

{

class Program

{

static void Main(string[] args)

{

Person p = new Person();

p.Output();

}

}

}

Вывод 1:

Метод “try” замещает кусок вашего кода, для последующей “ловли” его в метод “catch”, который в свою очередь производит действия, операции, в зависимости от полученной ошибки. В данном примере, этот метод при получении ошибки, меняет значения переменных на средние значения в мире.

**Лабораторная работа 4.2**

using System;

using System.Collections.Generic;

using System.Text;

using System;

using System.Collections.Generic;

namespace Lab4

{

public class Teacher

{

private string name; // Фамилия преподавателя

private string \_class; // класс

private uint hours; // нагрузка (количество часов)

public Teacher()

{

Console.WriteLine("Введите Фамилию");

name = Console.ReadLine();

Console.WriteLine("Введите название класса");

\_class = Console.ReadLine();

Console.WriteLine("Введите количество часов");

link:

try

{

string s = Console.ReadLine();

hours = Convert.ToUInt32(s);

}

catch(FormatException)

{

Console.WriteLine("Неверный формат ввода: не является неотрицательным числом из допустимого диапазона");

goto link;

}

catch(OverflowException)

{

Console.WriteLine("Неверный формат ввода: не является неотрицательным числом из допустимого диапазона");

goto link;

}

}

public uint get\_hours()// возвращает количество часов hours

{

return hours;

}

public string get\_name()// возвращает фамилию преподавателя name

{

return name;

}

}

}

namespace Lab4

{

public class Program

{

public static void Main(string[] args)

{

uint min = 0;// наименьшая нагрузка

string name\_min ="";// фамилия преподавателя с наименьшей нагрузкой

uint max = 0;// наибольшая нагрузка

string name\_max ="";// фамилия преподавателя с наибольшей нагрузкой

uint n; //количество преподавателей

Console.WriteLine("Введите количество преподавателей");

Link:

try

{

n = Convert.ToUInt32(Console.ReadLine());

}

catch (FormatException)

{

Console.WriteLine("Неверный формат ввода: не является неотрицательным числом из допустимого диапазона");

goto Link;

}

catch (OverflowException)

{

Console.WriteLine("Неверный формат ввода: не является неотрицательным числом из допустимого диапазона");

goto Link;

}

Console.WriteLine();

for (int i = 0; i<n; i++)

{

Console.WriteLine("{0}-й преподаветель", i + 1);

Teacher t = new Teacher();

if (i == 0)

{

min = t.get\_hours();

name\_min = t.get\_name();

}

if (t.get\_hours() >= max)

{

max = t.get\_hours();

name\_max = t.get\_name();

}

if (t.get\_hours() <= min)

{

min = t.get\_hours();

name\_min = t.get\_name();

}

Console.WriteLine();

}

if (n != 0)

{

Console.WriteLine("Преподаватель с самой высокой нагрузкой: {0}, нагрузка: {1}", name\_max, max);

Console.WriteLine("Преподаватель с самой низкой нагрузкой: {0}, нагрузка: {1}", name\_min, min);

}

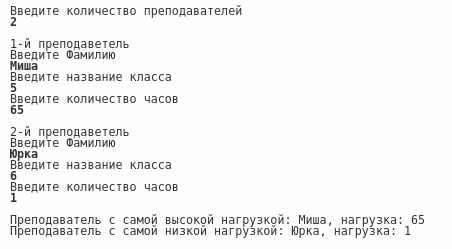
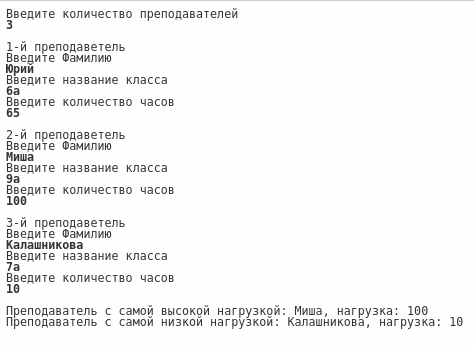
else

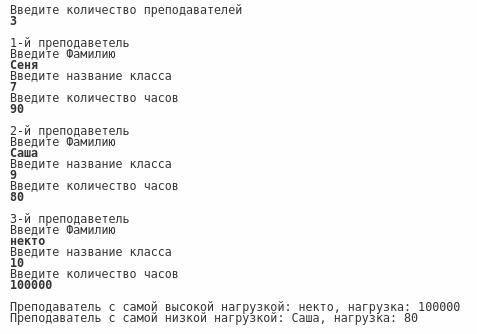
Console.WriteLine("Отсутствует список преподавателей");

}

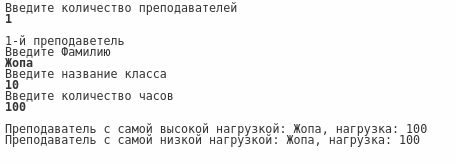
}

}









**Вывод:** данная программа очень удобная, особенно в работе для поиска и вывода нужной информации в крочайшие сроки и наименьшими затратами.

**Лабораторная работа 4.7**

//Опишите, используя Класс, почтовую сорти-ровку (город, улица, дом, квартира, кому, цен-ность). Составьте программу, определяющую: 1) сколько посылок отправлено в г. Москву; 2) сколько и куда (список городов) отправленно посылок ценностью выше 10 рублей; 3) есть ли адреса, куда отправлено более 1 посылки, если есть то сколько и кому

using System;

using System.Collections.Generic;

using System.Linq;

public class Program

{

public static void Main()

{

var list = new List<Post>();

list.Add(new Post("city1", "street1", 8, 32, "Mike", 34));

list.Add(new Post("city2", "street2", 5, 33, "Donald", 4));

list.Add(new Post("city1", "street3", 43, 72, "Tom", 8));

list.Add(new Post("city3", "street4", 1, 44, "Bob", 3));

list.Add(new Post("city5", "street5", 3, 1, "Frank", 412));

list.Add(new Post("city3", "street6", 6, 4, "Lola", 3));

list.Add(new Post("city5", "street7", 11, 2, "Adam", 60));

Console.Write(true);

}

public static int Foo1(string city, List<Post> list) => list.Count(p => p.City == city);

public static (int, List<string>) Foo2(int price, List<Post> list)

{

List<string> cities = new List<string>();

foreach(var post in list)

if(post.Price > price)

cities.Add(post.City);

return (cities.Count, cities);

}

}

public class Post

{

public Post(string city, string street, int home, int appartment, string to, int price)

{

City = city;

Street = street;

Home = home;

Appartment = appartment;

To = to;

Price = price;

}

public string City { get; set; }

public string Street { get; set; }

public int Home { get; set; }

public int Appartment { get; set; }

public string To { get; set; }

public int Price { get; set; }

public override string ToString() => "Город: {City}\nУлица: {Street}\nДом: {Home}\nКвартира: {Appartment}\nКому: {To}\nЦена: {Price}\n";

}

**Вывод:** данная программа есть в почте россии, но конечно улучшенная, но тем не менее. Расскрыли для себя новые классы, а так же их работу.