**Отчёт по практике 8.**

**Студент: Гуртякова А.П.**

**Задача:**Выполнить теоретическую и практическую часть по пройденному материалу

**Цель задания:**

1. Описать понятие индексации и его работы
2. Выполнить практическую часть работы

**Задание:**

1. Описать футболиста, у которого есть имя и номер.

Описать футбольную команду, состоящую из 11 футболистов

Написать индексатор, позволяющий обращаться к игроку в команде.

1. Характеристика футболиста должна конвертироваться в строку, поместить объект футболиста в Console.Writeline() и вывести характеристики
2. В написанном коде сделать так, чтобы не возникла ошибка при передаче отрицательных индексов и индексов больше 11
3. Создать словарь, использующий класс Word,

где Origin - слово, Transate - перевод слова

\*определить индексатор для обращения к словарю

**Ответить на вопросы:**

Напишите своими словами, как Вы понимаете, что такое индексирование и как оно происходит.

**Код программы 1:**

using System;

namespace Практика\_8

{

class FootballPlayer

{

public string Name { get; set; }

public int Number { get; set; }

public FootballPlayer(string name, int number)

{

Name = name;

Number = number;

}

public override string ToString() => $"{Number}:{Name}";

}

class FootballTeam

{

private readonly FootballPlayer[] \_players = new FootballPlayer[11];

public FootballTeam() { }

public FootballTeam(FootballPlayer[] players)

{

if (players.Length != 11)

throw new ArgumentOutOfRangeException("Игроков должно быть 11 человек");

\_players = players;

}

public FootballPlayer this[int index]

{

get => CheckIfIndexCorrect(index) ? \_players[index] : null;

set

{

if (!CheckIfIndexCorrect(index))

return;

\_players[index] = value;

}

}

private bool CheckIfIndexCorrect (int index) => index >= 0 && index <= 10;

}

class Program

{

static void Main(string[] args)

{

FootballTeam team = new FootballTeam();

team[8] = new FootballPlayer("Arasakov", 25);

Console.WriteLine(team[8]);

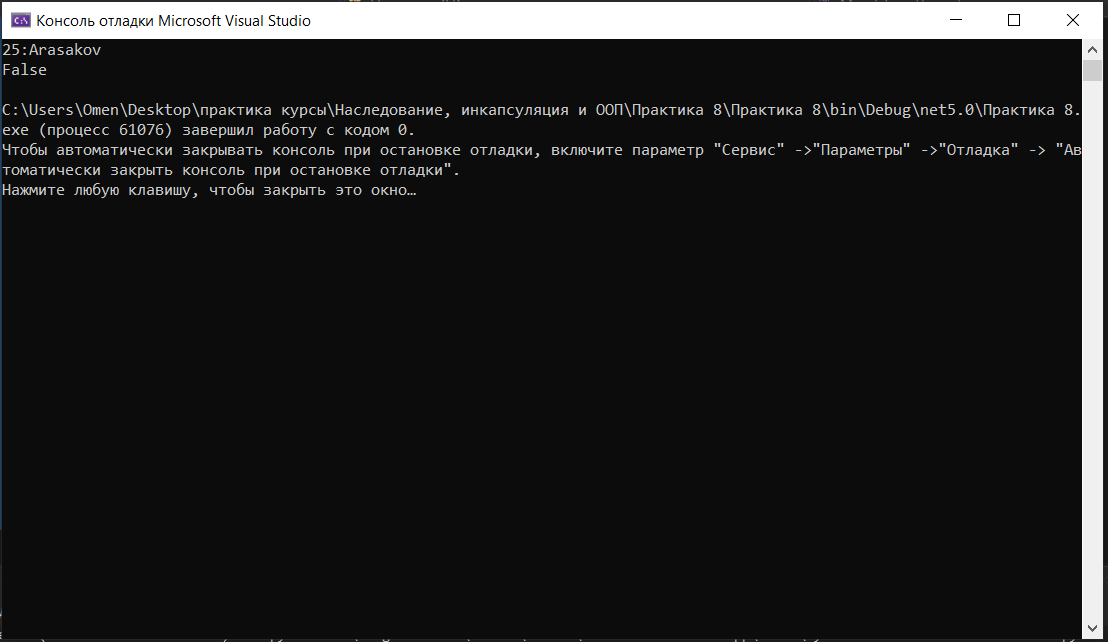
Console.WriteLine(team[8] is null);

}

}

}

**Скриншот работы программы 1:**



**Код программы 2:**

using System;

using System.Collections.Generic;

namespace Практика\_8\_часть\_2\_

{

class Word

{

public string Origin { get; set; }

public string Translate { get; set; }

public Word(string origin, string translate)

{

Origin = origin;

Translate = translate;

}

}

class Dictionary

{

private IList<Word> \_words = new List<Word>();

public string this [string origin]

{

get => FindWord(origin).Translate;

set => FindWord(origin).Translate = value;

}

public void AddWord(Word word) => \_words.Add(word);

private Word FindWord (string origin)

{

Word word = null;

foreach (var item in \_words)

if (item.Origin == origin)

{

word = item;

break;

}

return word;

}

}

class Program

{

static void Main(string[] args)

{

Dictionary dictionary = new Dictionary();

dictionary.AddWord(new Word("Water", "Вода"));

dictionary.AddWord(new Word("Fire", "Огонь"));

Console.WriteLine(dictionary["Water"]);

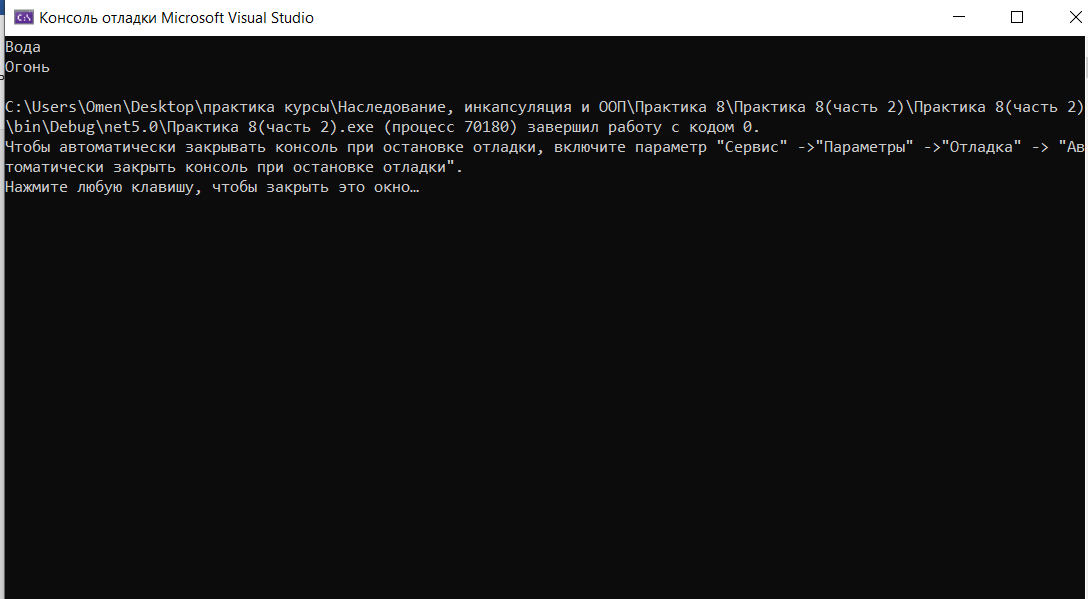
Console.WriteLine(dictionary["Fire"]);

}

}

}

**Скриншот работы программы 2:**



**Ответ:**

Индексаторы позволяют индексировать объекты и обращаться к данным по индексу. Фактически с помощью индексаторов мы можем работать с объектами как с массивами. По форме они напоминают свойства со стандартными блоками get и set, которые возвращают и присваивают значение.

Формальное определение индексатора:

возвращаемый\_тип this [Тип параметр1, ...]

{

get { ... }

set { ... }

}

В отличие от свойств индексатор не имеет названия. Вместо него указывается ключевое слово this, после которого в квадратных скобках идут параметры. Индексатор должен иметь как минимум один параметр.