МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «Гомельский государственный технический

университет имени П.О. Сухого»

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

Отчёт по лабораторной работе №5

По дисциплине «Объектно-ориентированное программирование»

**«Парсинг текстовых данных»**

Выполнил: студент

группы ИТИ-21

Говядкова П. Ю.

Принял: преподаватель

Карабчикова Е. А.

Гомель 2020

**Цель работы:** изучить основы синтаксиса объектно-ориентированного языка программирования, реализацию свойств, методов класса.

**Задание:**

1. Необходимо решить задачу, согласно варианту (Рисунок 1).
2. При создании классов руководствоваться Code Convention
3. Весь код должен содержать элементы документирования
4. Разработать модульные тесты для верификации созданного класса
5. Класс должен быть размещён в библиотеке классов
6. Модульные тесты — в отдельном проекте
7. В отдельном проекте реализовать консольный интерфейс

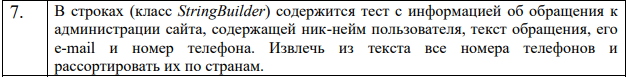


Рисунок 1 – Вариант задания

На рисунке 2 изображена структура решения.

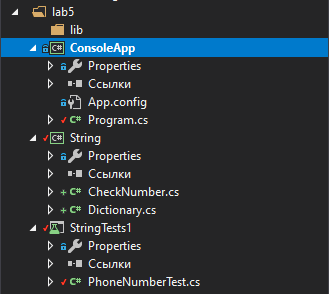


Рисунок 2 – Структура решения

Начало выполнения программы (Рисунок 3).

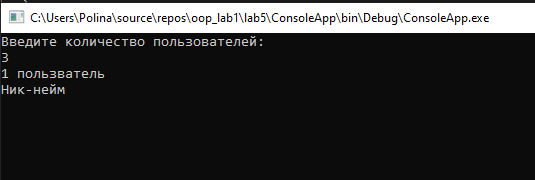


Рисунок 3 – Начало выполнения программы, ввод количества пользователей

Выполнение 1-ого пункта добавления данных о пользователе и сообщения администрации (Рисунок 4 – Рисунок 5).

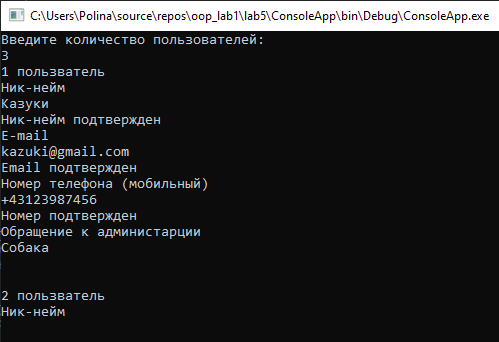


Рисунок 4 – Выполнение 1-ого пункта: добавления данных о пользователе и сообщения администрации

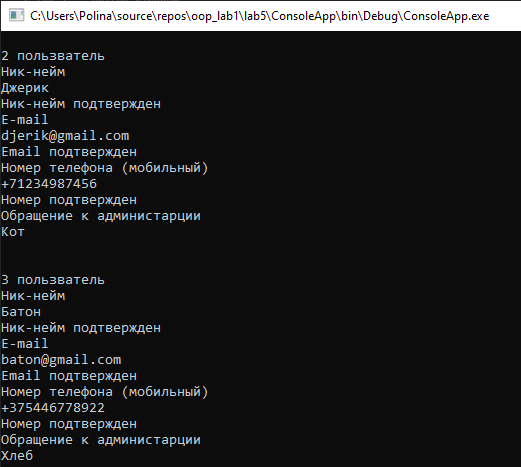


Рисунок 5 – Выполнение 1-ого пункта добавления данных о пользователе и сообщения администрации

Выполнение 2-ого пункта: распределение номеров телефонов по странам (Рисунок 6).

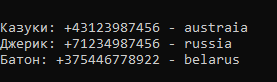


Рисунок 6 – Выполнение 2-ого пункта меню: распределение номеров телефонов по странам

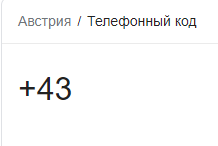


Рисунок 7 – Проверка корректности кода

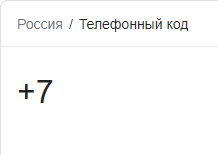


Рисунок 8 – Проверка корректности кода

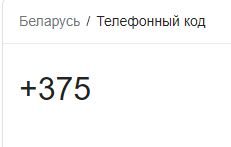


Рисунок 9 – Проверка корректности кода

Если номер страны не определён, то будет выводиться пустая строка (Рисунок 10).

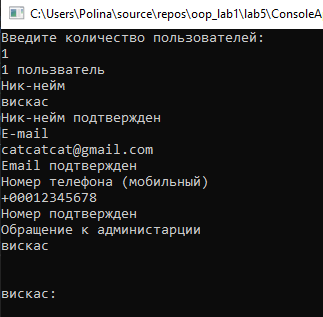


Рисунок 10 – Случай с несуществующим номером

Результаты прохождения тестов (Рисунок 11).

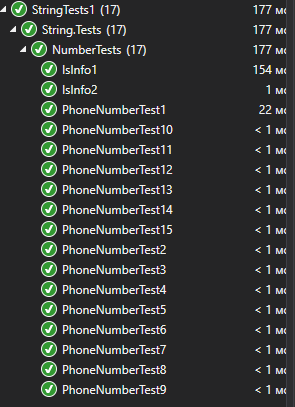


Рисунок 11 – Результат выполнения тестов

Создание xml файлов (Рисунок 12 – 13).

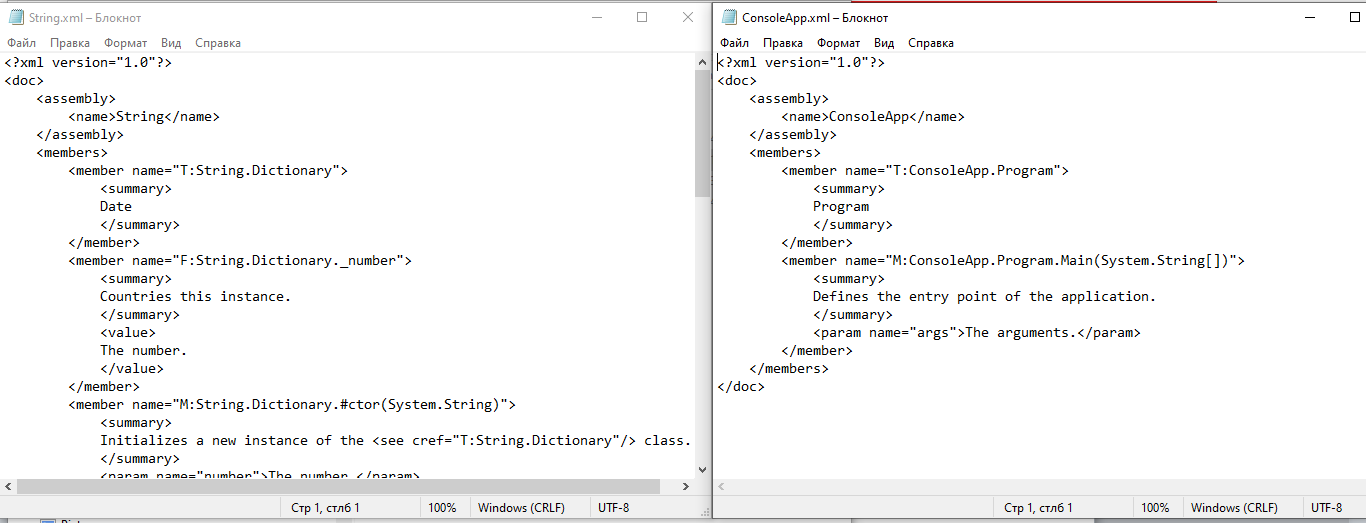


Рисунок 12 – Xml файлы для Program и String

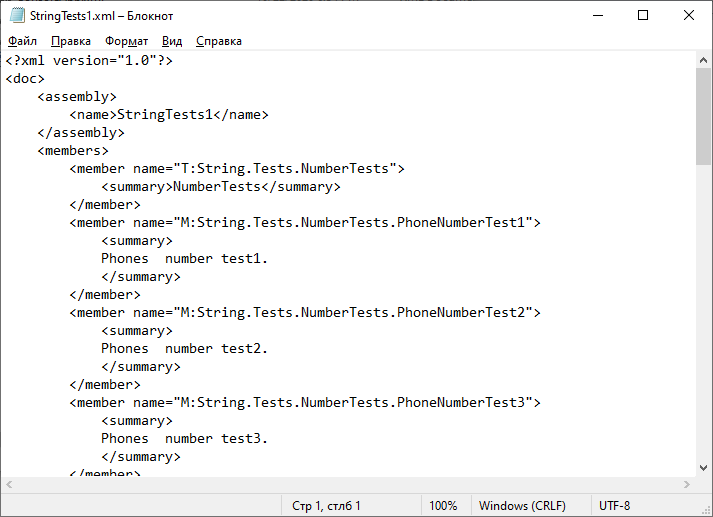


Рисунок 13 – Xml файл для тестов

**Вывод:** в ходе лабораторной работы были изучены основы синтаксиса объектно-ориентированного языка программирования, реализация свойств, методов класса.

**ПРИЛОЖЕНИЕ А**

**Листинг программы:**

Program.

using String;

using System;

namespace ConsoleApp

{

/// <summary>

/// Program

/// </summary>

internal class Program

{

/// <summary>

/// Defines the entry point of the application.

/// </summary>

/// <param name="args">The arguments.</param>

private static void Main(string[] args)

{

int number = 0;

Console.WriteLine("Введите количество пользователей:");

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

string[] nickname = new string[number];

string[] email = new string[number];

string[] phone = new string[number];

string[] text = new string[number];

CheckNumber Info = new CheckNumber();

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

{

Console.WriteLine(i + 1 + " пользватель");

Console.WriteLine("Ник-нейм");

nickname[i] = Convert.ToString(Console.ReadLine());

//nickname[i] = "ник ник";

while (true)

{

if (Info.IsInfo(nickname[i]) == true)

{

Console.WriteLine("Ник-нейм подтвержден");

break;

}

else

{

Console.WriteLine("Некорректный ник-нейм");

Console.WriteLine("Введите ник-нейм");

nickname[i] = Convert.ToString(Console.ReadLine());

}

}

Console.WriteLine("E-mail");

email[i] = Convert.ToString(Console.ReadLine());

//email[i] = "niknik821@gmail.com";

while (true)

{

if (Info.IsInfo(email[i]) == true)

{

Console.WriteLine("Email подтвержден");

break;

}

else

{

Console.WriteLine("Некорректный email");

Console.WriteLine("Введите E-mail");

email[i] = Convert.ToString(Console.ReadLine());

}

}

Console.WriteLine("Номер телефона (мобильный)");

phone[i] = Convert.ToString(Console.ReadLine());

//phone[i] = "+075445337922";

while (true)

{

if (Info.IsInfo(phone[i]) == true)

{

Console.WriteLine("Номер подтвержден");

break;

}

else

{

Console.WriteLine("Некорректный номер телефона");

Console.WriteLine("Введите номер телефона (мобильный)");

phone[i] = Convert.ToString(Console.ReadLine());

}

}

Console.WriteLine("Обращение к администарции");

text[i] = Convert.ToString(Console.ReadLine());

Console.WriteLine("\n");

}

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

{

CheckNumber Number = new CheckNumber(phone[i]);

Console.WriteLine(nickname[i] + ": " + Number.PhoneNumber(phone[i]));

}

Console.ReadKey();

}

}

}

CheckNumber.

using System;

using System.Collections.Generic;

using System.Text;

using System.Text.RegularExpressions;

namespace String

{

/// <summary>

/// UserInfo

/// </summary>

public class CheckNumber

{

/// <summary>

/// The number

/// </summary>

private string \_number;

/// <summary>

/// Initializes a new instance of the <see cref="Number"/> class.

/// </summary>

/// <param name="number">The number.</param>

public CheckNumber(string number)

{

this.\_number = number;

}

/// <summary>

/// Initializes a new instance of the <see cref="Number"/> class.

/// </summary>

public CheckNumber()

{

}

/// <summary>

/// Phones the number.

/// </summary>

/// <param name="phone">The phone.</param>

/// <returns>Array with codes and countries</returns>

public string PhoneNumber(string phone)

{

Dictionary Info = new Dictionary(phone);

StringBuilder new\_text = new StringBuilder();

string result = "";

string phone\_whith\_code\_1 = phone.Substring(0, phone.Length - 11);

string phone\_whith\_code\_2 = phone.Substring(0, phone.Length - 10);

string phone\_whith\_code\_3 = phone.Substring(0, phone.Length - 9);

string phone\_whith\_code\_4 = phone.Substring(0, phone.Length - 8);

Dictionary<string, string> array = Info.CountriesWithCodes();

foreach (KeyValuePair<string, string> keyValue in array)

{

try

{

if (keyValue.Key == phone\_whith\_code\_1 || keyValue.Key == phone\_whith\_code\_2 || keyValue.Key == phone\_whith\_code\_3 || keyValue.Key == phone\_whith\_code\_4) result = phone + " - " + keyValue.Value;

}

catch

{

result = "";

}

}

new\_text.Append(result);

return new\_text.ToString();

}

/// <summary>

/// Determines whether the specified text is information.

/// </summary>

/// <param name="text">The text.</param>

/// <returns>

/// <c>true</c> if the specified text is information; otherwise, <c>false</c>.

/// </returns>

public bool IsInfo(string text)

{

bool result = false;

string pattern\_for\_email = @"([a-zA-Z0-9\_\-\.]+)@([a-zA-Z0-9\_\-\.]+)\.([a-zA-Z]{2,5})";

string pattern\_for\_nickname = @"([А-Я]{1}[а-я]{1,})";

string pattern\_for\_phone = @"\d{1,3}[\-|\s]?\d{2}[\-|\s]?\d{3}[\-|\s]?\d{2}[\-|\s]?\d{2}";

if (Regex.IsMatch(text, pattern\_for\_email, RegexOptions.IgnoreCase) || Regex.IsMatch(text, pattern\_for\_nickname, RegexOptions.IgnoreCase) || Regex.IsMatch(text, pattern\_for\_phone, RegexOptions.IgnoreCase)) result = true;

else result = false;

return result;

}

}

}

Dictionary.

using System.Collections.Generic;

namespace String

{

/// <summary>

/// Date

/// </summary>

public class Dictionary

{

/// <summary>

/// Initializes a new instance of the <see cref="Dictionary"/> class.

/// </summary>

public Dictionary()

{

}

/// <summary>

/// Countrieses with codes.

/// </summary>

/// <returns>Dictionary with countries and codes</returns>

public Dictionary<string, string> CountriesWithCodes()

{

Dictionary<string, string> countries\_with\_codes = new Dictionary<string, string>()

{

["+43"] = "аustria",

["+61"] = "australia",

["+54"] = "argentina",

["+93"] = "afghanistan",

["+32"] = "belgium",

["+55"] = "brazil",

["+379"] = "vatican",

["+44"] = "united\_kingdom",

["+36"] = "hungary",

["+58"] = "venezuela",

["+84"] = "vietnam",

["+49"] = "germany",

["+30"] = "greece",

["+45"] = "denmark",

["+20"] = "egypt",

["+91"] = "india",

["+98"] = "iran",

["+34"] = "spain",

["+39"] = "italy",

["+7172"] = "kazakhstan",

["+86"] = "china",

["+57"] = "colombia",

["+53"] = "cuba",

["+994"] = "azerbaijan",

["+355"] = "albania",

["+213"] = "algeria",

["+244"] = "angola",

["+376"] = "andorra",

["+374"] = "armenia",

["+880"] = "bangladesh",

["+591"] = "bolivia",

["+973"] = "bahrain",

["+375"] = "belarus",

["+501"] = "belize",

["+32"] = "belgium",

["+229"] = "benin",

["+359"] = "bulgaria",

["+387"] = "bosnia\_and\_herzegovina",

["+267"] = "botswana",

["+55"] = "brazil",

["+673"] = "brunei",

["+226"] = "burkina\_faso",

["+257"] = "burundi",

["+975"] = "bhutan",

["+678"] = "vanuatu",

["+670"] = "east\_timor",

["+241"] = "gabon",

["+509"] = "haiti",

["+592"] = "guyana",

["+220"] = "gambia",

["+233"] = "ghana",

["+502"] = "guatemala",

["+224"] = "guinea",

["+245"] = "guinea\_bissau",

["+504"] = "honduras",

["+995"] = "georgia",

["+253"] = "djibouti",

["+260"] = "zambia",

["+263"] = "zimbabwe",

["+972"] = "israel",

["+962"] = "jordan",

["+964"] = "iraq",

["+353"] = "ireland",

["+354"] = "iceland",

["+967"] = "yemen",

["+238"] = "cape\_verde",

["+855"] = "cambodia",

["+237"] = "cameroon",

["+974"] = "qatar",

["+254"] = "kenya",

["+357"] = "cyprus",

["+996"] = "kyrgyzstan",

["+686"] = "kiribati",

["+269"] = "comoros",

["+243"] = "congo\_democratic\_republic",

["+242"] = "congo\_republic",

["+506"] = "costa\_rica",

["+225"] = "cote\_divoire",

["+965"] = "kuwait",

["+856"] = "laos",

["+371"] = "latvia",

["+266"] = "lesotho",

["+231"] = "liberia",

["+961"] = "lebanon",

["+218"] = "libya",

["+370"] = "lithuania",

["+423"] = "liechtenstein",

["+352"] = "luxembourg",

["+1268"] = "antigua\_and\_barbuda",

["+1242"] = "bahamas",

["+1246"] = "barbados",

["+1767"] = "dominica",

["+1809"] = "dominican\_republic",

["+1"] = "north\_america",

["+230"] = "mauritius",

["+222"] = "mauritania",

["+261"] = "madagascar",

["+389"] = "macedonia",

["+265"] = "malawi",

["+60"] = "malaysia",

["+223"] = "mali",

["+960"] = "maldives",

["+356"] = "malta",

["+212"] = "morocco",

["+692"] = "marshall\_islands",

["+52"] = "mexico",

["+259"] = "mozambique",

["+373"] = "moldova",

["+377"] = "monaco",

["+976"] = "mongolia",

["+95"] = "myanmar",

["+264"] = "namibia",

["+674"] = "nauru",

["+977"] = "nepal",

["+227"] = "niger",

["+234"] = "nigeria",

["+31"] = "netherlands",

["+505"] = "nicaragua",

["+64"] = "new\_zealand",

["+47"] = "norway",

["+971"] = "united\_arab\_emirates",

["+968"] = "oman",

["+92"] = "pakistan",

["+680"] = "palau",

["+507"] = "panama",

["+675"] = "papua\_new\_guinea",

["+595"] = "paraguay",

["+51"] = "peru",

["+48"] = "poland",

["+351"] = "portugal",

["+7"] = "russia",

["+250"] = "rwanda",

["+40"] = "romania",

["+503"] = "el\_salvador",

["+685"] = "samoa",

["+378"] = "san\_marino",

["+239"] = "sao\_tome\_and\_principe",

["+966"] = "saudi\_arabia",

["+268"] = "swaziland",

["+850"] = "north\_korea",

["+221"] = "senegal",

["+1784"] = "saint\_vincent\_and\_the\_grenadines",

["+1869"] = "saint\_kitts\_and\_nevis",

["+1758"] = "saint\_lucia",

["+381"] = "serbia",

["+65"] = "singapore",

["+963"] = "syria",

["+421"] = "slovakia",

["+986"] = "slovenia",

["+677"] = "solomon\_islands",

["+252"] = "somalia",

["+249"] = "sudan",

["+597"] = "suriname",

["+232"] = "sierra\_leone",

["+992"] = "tajikistan",

["+66"] = "thailand",

["+255"] = "tanzania",

["+228"] = "togo",

["+676"] = "tonga",

["+1868"] = "trinidad\_and\_tobago",

["+688"] = "tuvalu",

["+216"] = "tunisia",

["+993"] = "turkmenistan",

["+90"] = "turkey",

["+256"] = "uganda",

["+998"] = "uzbekistan",

["+380"] = "ukraine",

["+598"] = "uruguay",

["+691"] = "federated\_states\_of\_micronesia",

["+679"] = "fiji",

["+63"] = "philippines",

["+358"] = "finland",

["+33"] = "france",

["+385"] = "croatia",

["+236"] = "central\_african\_republic",

["+235"] = "chad",

["+381"] = "montenegro",

["+420"] = "czech\_republic",

["+56"] = "chile",

["+41"] = "switzerland",

["+46"] = "sweden",

["+94"] = "sri\_lanka",

["+593"] = "ecuador",

["+240"] = "equatorial\_guinea",

["+291"] = "eritrea",

["+372"] = "estonia",

["+251"] = "ethiopia",

["+82"] = "south\_korea",

["+27"] = "south\_africa",

["+1876"] = "jamaica",

["+81"] = "japan",

["+1604"] = "british\_columbia, CA",

["+1403"] = "alberta, CA",

["+1306"] = "saskatchewan, CA",

["+1204"] = "manitoba, CA",

["+1807"] = "ontario, CA",

["+1705"] = "ontario, CA",

["+1819"] = "Quebec, CA",

["+1418"] = "Quebec, CA",

["+1506"] = "new\_brunswick, CA",

["+1902"] = "prince\_edward\_island, nova\_scotia, CA",

["+1709"] = "newfoundland, CA"

};

return countries\_with\_codes;

}

}

}