**Лабораторная работа №18\_2**

## Постановка задачи:

**Вариант 9**

1. Даны две непустые очереди (вывести их на экран консоли), которые содержат одинаковое количество элементов. Объединить очереди в одну, в которой элементы исходных очередей чередуются. Вывести полученный результат на консоль.

2. Создать класс Игрушка, у которого будут поля название с большой буквы и тип (кукла, конструктор, пазл и т.д.). Создать словарь, содержащий список Игрушек (не менее 10).

* Упорядочить список по алфавиту типов игрушек. Результат вывести на экран консоли.
* Добавить в словарь еще одну игрушку, с полями, введенными с консоли. Вывести результат на экран.
* Удалить игрушки, у которых название типа содержит букву «К». Результат вывести на экран консоли.

Предусмотреть возникновение всех возможных исключений.

## Код:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text.RegularExpressions;

namespace Lab

{

class Toy

{

public string Name { get; set; }

public string Type { get; set; }

public Toy(string name, string type)

{

Name = name;

Type = type;

}

}

class Program

{

static void Main(string[] args)

{

//First();

Second();

}

static void First()

{

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

Console.WriteLine("Задание 1:");

try

{

int count = 3;

Queue<int> queue1 = new Queue<int>();

Queue<int> queue2 = new Queue<int>();

Queue<int> queueMain = new Queue<int>();

for (int i = 1; i < count \* 2; i += 2)

{

queue1.Enqueue(i);

queue2.Enqueue(i + 1);

}

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

{

queueMain.Enqueue(queue1.Dequeue());

queueMain.Enqueue(queue2.Dequeue());

}

foreach (int item in queueMain)

{

Console.WriteLine($"{item}");

}

}

catch

{

Console.WriteLine("Ошибка!");

}

Console.ReadLine();

}

static void Second()

{

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

Console.WriteLine("Задание 2:");

try

{

Dictionary<string, string> toys = new Dictionary<string, string>();

toys.Add("Раскраска", "Тысячелетний сокол");

toys.Add("Пазл", "Татуин");

toys.Add("Конструктор", "Лего: Star Wars");

toys = toys.OrderBy(pair => pair.Key).ToDictionary(pair => pair.Key, pair => pair.Value);

foreach (var item in toys)

{

Console.WriteLine(item.Key + " - " + item.Value);

}

Regex regex = new Regex(@"[А-Я]\w");

Console.WriteLine("Добавим игрушку в словарь:");

Console.Write("Тип: "); string type = Console.ReadLine();

Console.Write("Название: "); string name = Console.ReadLine();

while (!regex.IsMatch(name))

{

Console.WriteLine("Название с большой буквы!");

Console.Write("Тип: "); type = Console.ReadLine();

Console.Write("Название: "); name = Console.ReadLine();

}

bool correctness = false;

while (!correctness)

{

try

{

toys.Add(type, name);

correctness = true;

}

catch

{

Console.WriteLine("Скорее всего такой тип уже есть!");

correctness = false;

}

}

foreach (var item in toys.ToArray())

{

if (item.Key[0] == 'К' || item.Key[0] == 'к')

{

toys.Remove(item.Key);

}

}

foreach (var item in toys)

{

Console.WriteLine(item.Key + " - " + item.Value);

}

}

catch (Exception ex)

{

Console.WriteLine(ex.GetBaseException());

}

Console.ReadLine();

}

}

}

## Скриншоты:

