**ФГБОУ ВО**

**Национальный исследовательский университет**

**«МЭИ»**

**Индивидуальное задание**

по курсу

«Технология программирования»

**Реализация темы «Государство» средствами ООП на языке С# или С++**

**Выполнил:**

студент группы А-12-19 Саутов Д.С.

**Москва, 2017**

**Индивидуальное задание по теме «Государство»**

**Саутов Давид Сергеевич А-12-18**

**Задание:**

Реализовать средствами ООП на языке С# или С++ выбранную студентом тему.

1. **Постановка задачи.**

Разработать программу, реализующую средствами ООП выбранную тему. В данном случае будет создана программа на тему «Государство». В государстве будет 4 сферы со своими полями. Пользователь будет в роли президента: принимать решения в гос. вопросах, распределять бюджет, инвестировать из одной сферы в другую, а также создавать резервные фонды, в случае необходимости.

У каждой сферы будет по 2 поля: Текущее значение благосостояния сферы и коэффициент развития сферы.

При распределении бюджета, некоторое значение вычитается из одной сферы, и прибавляется к другой с понижающим коэффициентом в 0.9.

При инвестициях в сферу, можно не только повышать значение благосостояния сферы, но еще и коэффициент развития сферы с понижающим коэффициентом в 0.8.

Создание резерва возможно при значении благополучия сферы превышающим начальное значение. Резерв забирает ¼ от максимального значения благополучия сферы, но повышает коэффициент сферы на 1/10 от базового значения.

Возврат резерва делает все в точности, как и его создание, но наоборот. Ограничения не позволяющих снимать резерв отсутствуют.

Интерфейс взаимодействия с пользователем будет реализован через средства **Windows Form (.NET FRAMEWORK).**

В качестве входных и выходных данных будут выступать элементы формы **Windows Form.**

**Ограничения:**

1. Значение сферы не может принимать отрицательные значения
2. Коэффициент сферы не может принимать отрицательное значение

**Обработка ограничений:**

1. Пользователю сообщается о плохом управлении странной, после чего программа закрывается.
2. Пользователю сообщается о плохом управлении странной, после чего программа закрывается.

**Среда разработки:**

Необходимо разработать визуальное приложение в среде разработки Visual Studio Community 2019 на языке C#, используя Windows Form (.NET FRAMEWORK).

**2.Разработка программы**

**2.1 Разработка структуры программы.**

В основе программы будет форма (**Windows Form .NET FRAMEWORK),** которую будет вызывать программа.

В программе будет реализовано 7 классов, один из которых будет абстрактным, а также 1 интерфейс. Их названия, свойства, методы указаны ниже в Табл. 1-8.

Табл. 1

*Интерфейс Ruling*

|  |  |
| --- | --- |
| Методы | Смысл методов |
| ruling  (public) | Позволяет менять значения одной сферы, за счет значений другой сферы, с понижающим коэффициентом. |

Табл. 2

*Абстрактный класс Person*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поля | Тип данных | Смысл поля | Методы | Смысл методов |
| fio (protected) | string | Ф.И.О. человека | greeting | Возвращает должность человека, а также ФИО |

Табл. 3

*класс Accident*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поля | Тип данных | Смысл поля | Методы | Смысл методов |
| Counts (Public) | int[,] | Значения, на которые поменяются сферы, при определенном выборе президента | greeting | Возвращает должность человека, а также ФИО |
| situation (Public) |  |  |  |  |

Табл. 4

*класс Ministr*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поля | Тип данных | Смысл поля | Методы | Смысл методов |
| list (public) | List<Accident> | Список проблем, требующих решения президента. | greeting | Возвращает должность человека, а также ФИО |

Табл. 5

*Класс Sphere*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поля | Тип данных | Смысл поля | Методы | Смысл методов |
| \_count | int | Текущее значение благосостояния сферы | Свойство count{get;set;} | Позволяет получать значение из поля \_соunt, а также менять его значение |
| \_effefficiency  (private) | int | коэффициент развития сферы | Свойство effefficiency {get; set;} | Позволяет получать значение из поля \_ effefficiency, а также менять его значение |
| reserve  (private) | bool | Показывает, есть ли у данной сферы резерв | endGame | Проверяет на наличие у полей нулевого значения |
|  |  |  | setReserve | Создает резерв |
|  |  |  | getReserve | Распустить резерв |
|  |  |  | outOfRange | Проверяет поля на выход за границы |

Табл. 6

*Класс Country*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поля | Тип данных | Смысл поля | Методы | Смысл методов |
| \_ecologicalSphere  (private) | Sphere | Экологическая сфера | Свойство ecologicalSphere{get;set;} | Позволяет получать значение из поля \_соunt, а также менять его значение |
| \_economicalSphere (private) | Sphere | Экономическая сфера | Свойство economicalSphere{get; set;} | Позволяет получать значение из поля \_effefficiency, а также менять его значение |
| \_militarySphere (private) | Sphere | Военная сфера | Свойство militarySphere {get; set;} | Позволяет получать значение из поля \_militarySphere, а также менять его значение |
| \_socialSphere  (private) | Sphere | Социальная сфера | Свойство socialSphere {get; set;} | Позволяет получать значение из поля \_socialSphere, а также менять его значение |
| nameOfCountry  (public) | string | Название страны | changeCountSphere | Изменяет значение выбранной сферы |
|  |  |  | ruling | Позволяет управлять странной |

Табл. 7

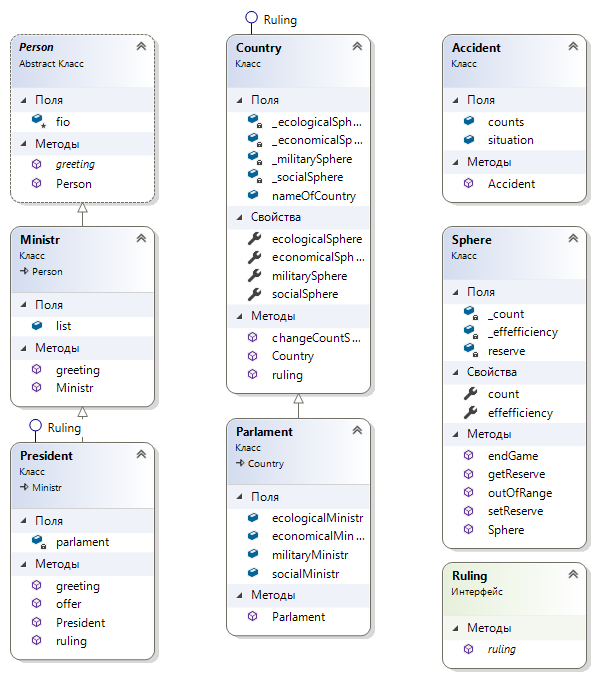
*Класс Parlament*

|  |  |  |
| --- | --- | --- |
| Поля | Тип данных | Смысл поля |
| ecologicalMinistr (public) | Sphere | Министр экологической сферы |
| economicalMinistr (public) | Sphere | Министр экономической сферы |
| militaryMinistr (public) | Sphere | Министр военной сферы |
| socialMinistr (public) | Sphere | Министр социальной сферы |

Табл. 8

*Класс President*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поля | Тип данных | Смысл поля | Методы | Смысл методов |
| parlament  (public) | Parlament | Парламент | greeting | Возвращает должность человека, а также ФИО |
|  |  |  | offer | Принимает решение для проблемы, указанной министром. |
|  |  |  | ruling | Управление государством |



*Рис. 1 Диаграмма классов*

**2.2 Разработка схема алгоритма**

Функция по управлению странной из класса President ruling(int[] massiv)

На вход принимает массив из целочисленных значений.

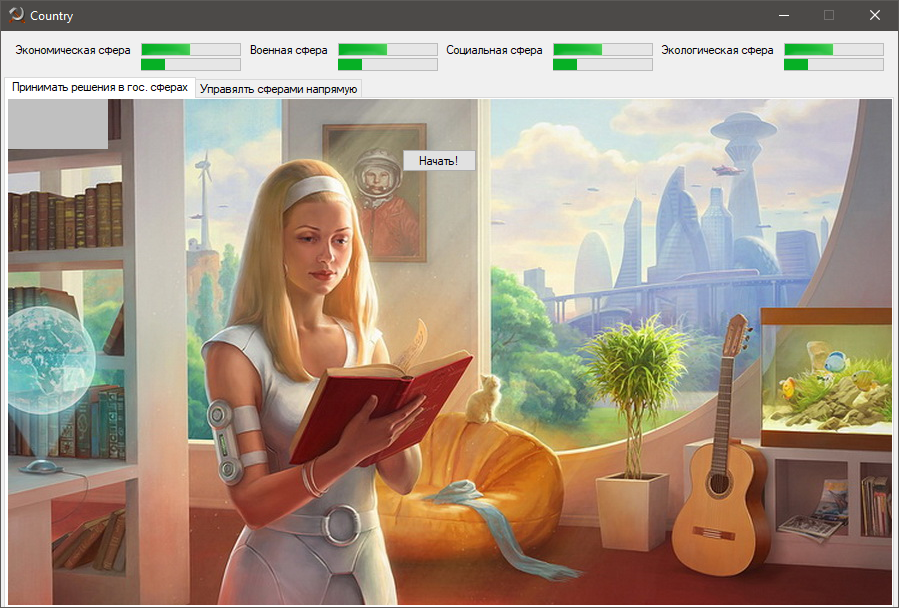
1 элемент – 1-я сфера из которой вычитаем значение

2 элемент – 2-я сфера к которой прибавляем значение \* 0.9

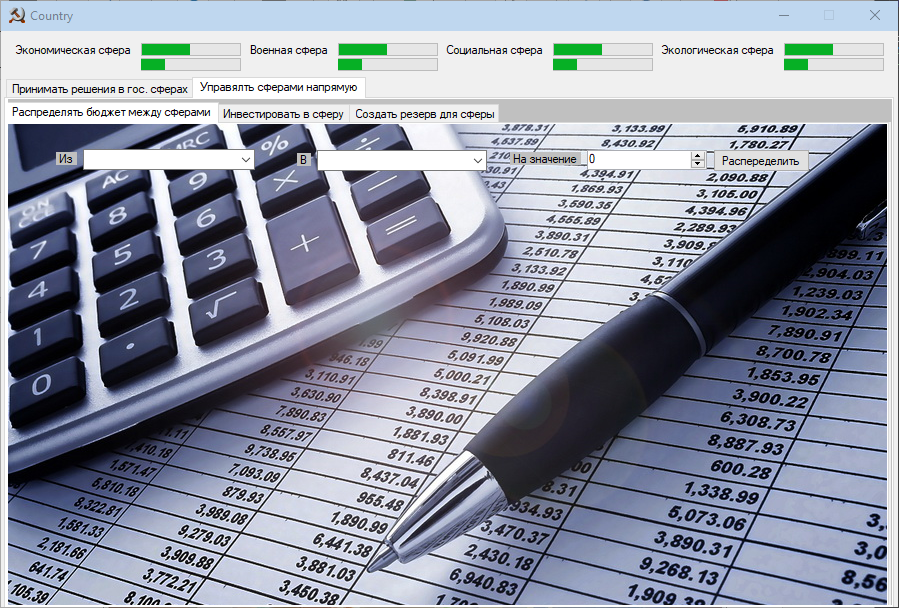
3 элемент – само значение, на которое мы меняем показатели сферы

**2.3 Разработка пользовательского интерфейса**

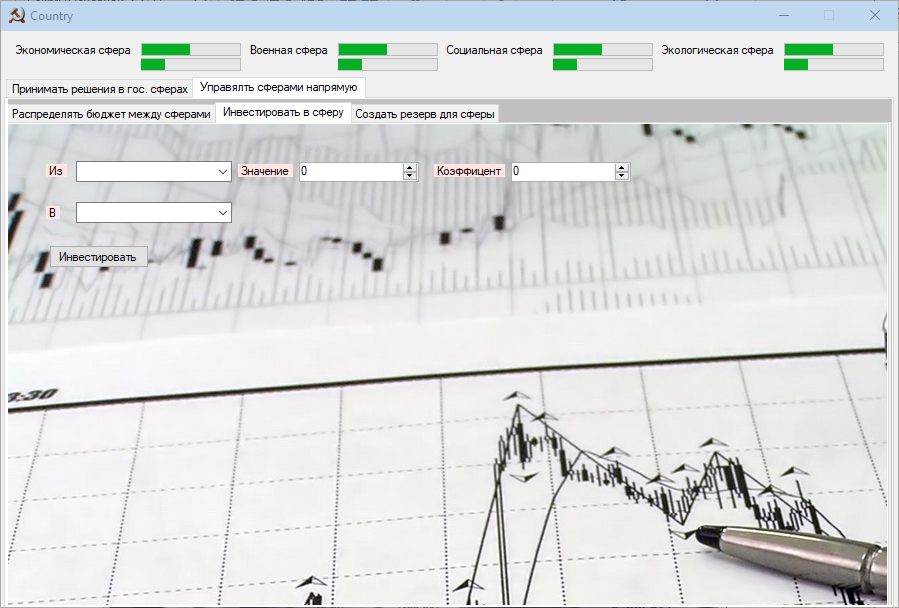
Программа предназначен для любого пользователя. Вид формы представлен на Рис. 2 - 5, используемые элементы управления в форме приведены в табл. 8.



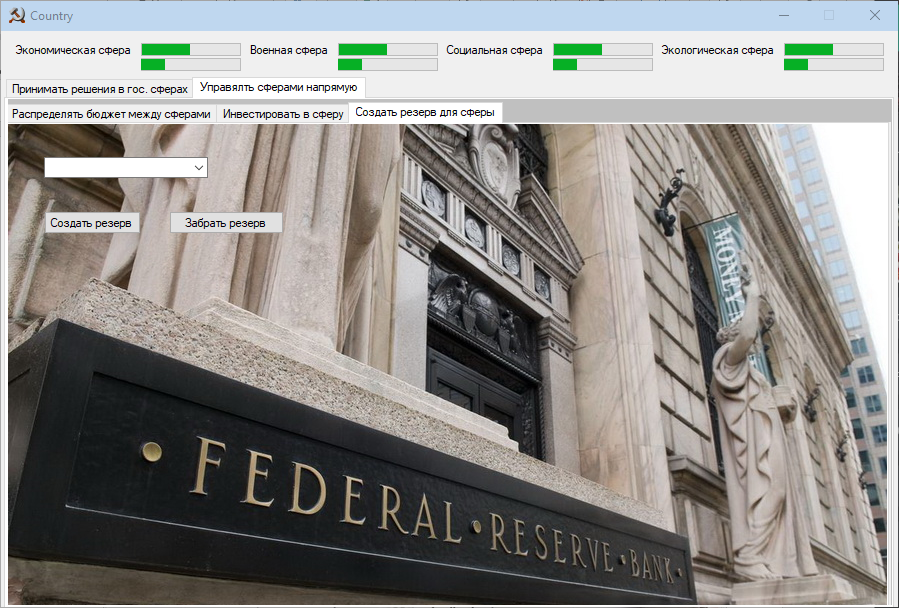
*Рис. 2 Раздел формы для принятия решения*



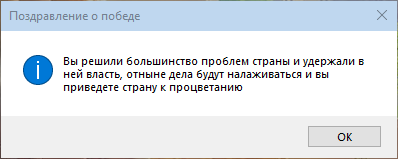
*Рис. 3 Раздел формы для распределения бюджета между сферами*



*Рис. 4 Раздел формы для инвестирования в сферу*



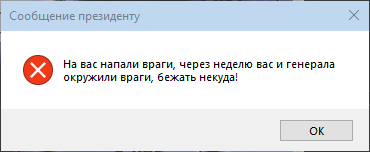
*Рис. 5 Раздел формы для инвестирования в сферу*



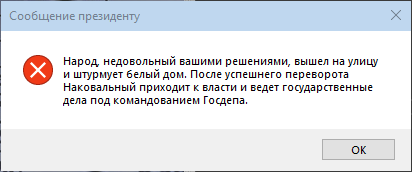
*Рис. 6 Сообщение при решении всех проблем*



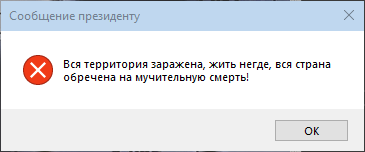
*Рис. 7 Сообщение при понижении показателей экономической сферы до нуля*



*Рис. 8 Сообщение при понижении показателей военной сферы до нуля*



*Рис. 9 Сообщение при понижении показателей социальной сферы до нуля*



*Рис. 10 Сообщение при понижении показателей социальной сферы до нуля*

Табл. 1

*Используемые элементы управления в форме*

|  |  |  |
| --- | --- | --- |
| Элемент управления | класс | Описание |
| label1-13 | Label | Навигационная строка |
| numericUpDown1-3 | NumericUpDown | Элемент для ввода чисел |
| comboBox1-5 | ComboBox | Элемент для выбора определенных ответов |
| textBox3-8 | TextBox | Элемент для записи текста |
| button1-5 | Button | Кнопки выполнения определенных функций |
| tabControl1-2 | TabControl | Позволяет разделить форму на разделы |
| progressBar1-8 | ProgressBar | Отображение числового значение шкалой |

**3.Реализация и тестирование**

***Реализация***

Программа разработана на языке C# при помощи форм **Windows Form (.NET FRAMEWORK)**.

**Описание компьютера:**

Операционная система: Windows 10 (x64)

ОЗУ: 16 ГБ

Процессор: Intel Core i5-2500

Видеокарта: Radeon R7 200 Series

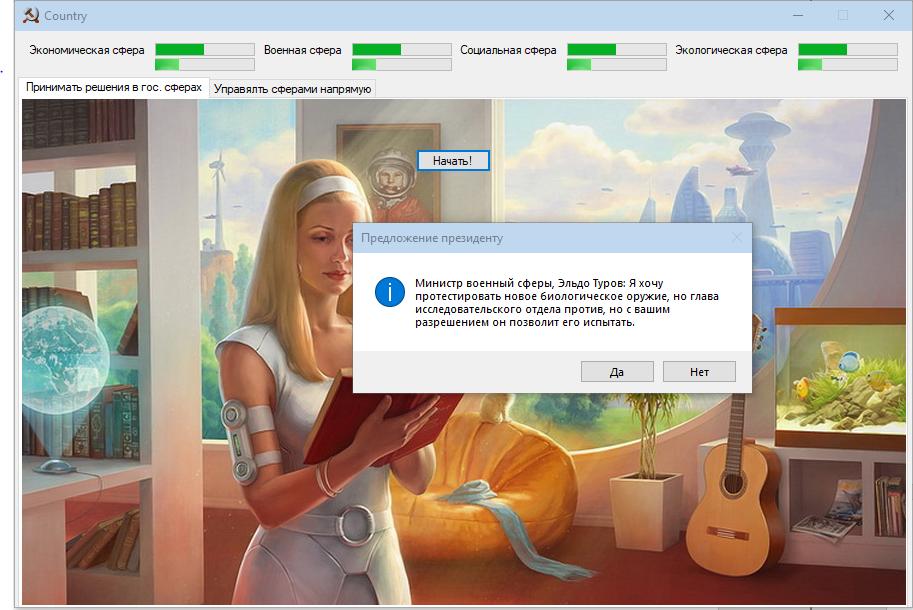
***Тестирование***

**Объект тестирования:** Country.exe

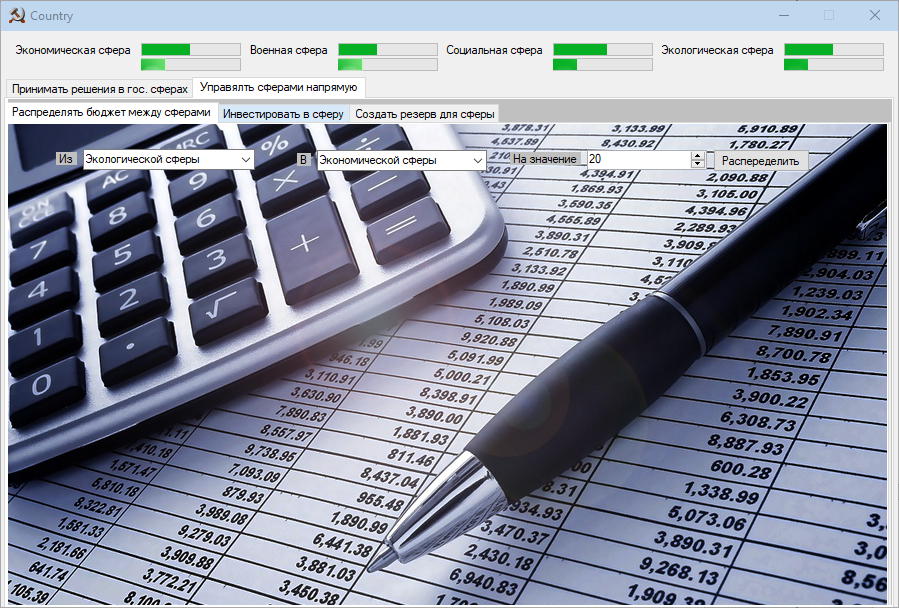
**Цель тестирования:** Проверка общей работоспособности программы.

**Метод тестирование:** функциональное тестирование.

Тестирование всех функций и элементов формы приведены в тестированиях, указанных ниже на Рис. 11-14



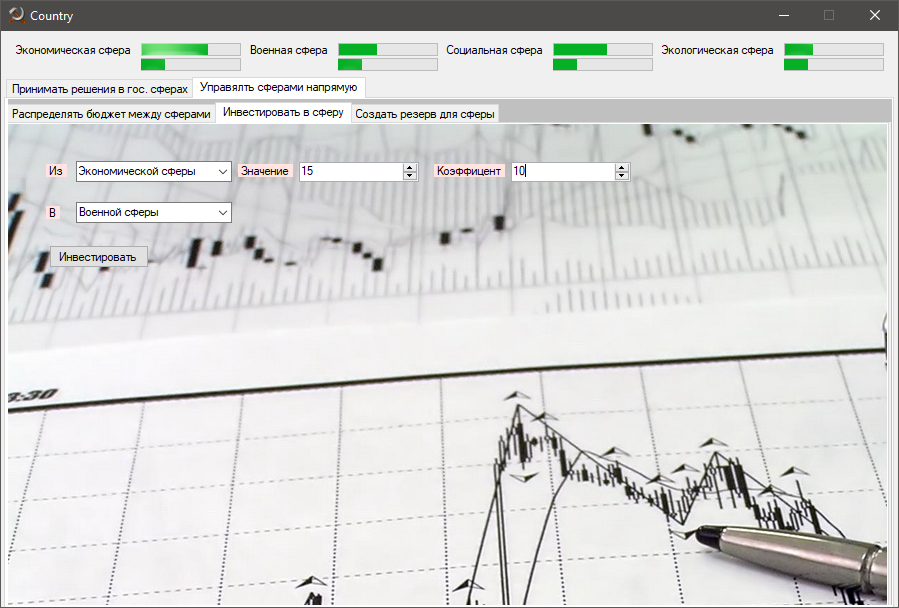
*Рис. 11 Тестирование. Принятие решений в государственных сферах*



*Рис. 12 Тестирование. Распределение бюджета между сферами*



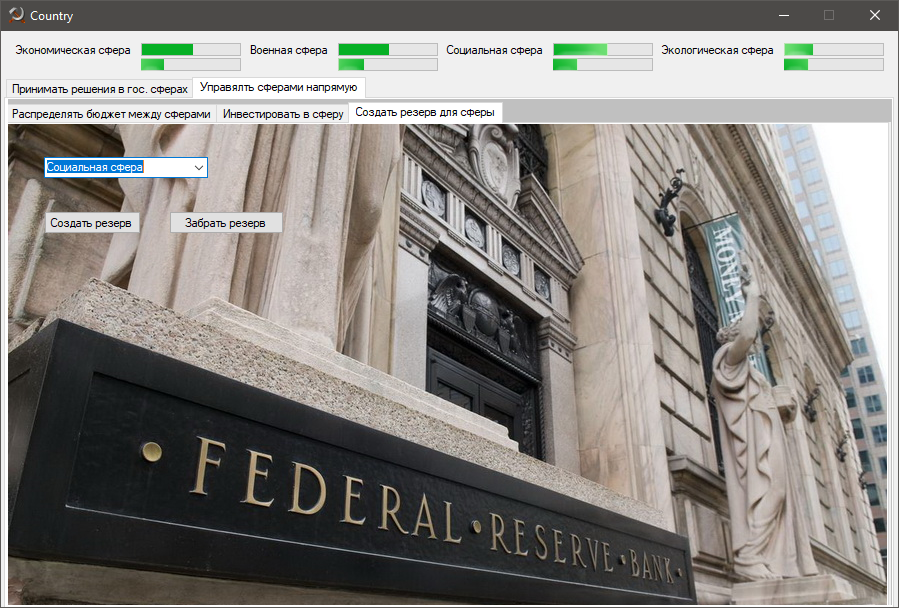
*Рис. 13 Тестирование. Изменение после нажатия кнопки «Распределить»*



*Рис. 14 Тестирование. Инвестирование в сферу*



*Рис. 15 Тестирование. Изменение после нажатия кнопки «Инвестировать»*



*Рис. 16 Тестирование. Создание резерва для сферы*



*Рис. 17 Тестирование. Изменение после нажатия кнопки «Создать резерв»*



*Рис. 18 Тестирование. Изменение после нажатия кнопки «Забрать резерв»*

**Код будет представлен ниже в разделе «Листинг кода».**

**«Листинг кода»**

**Абстрактный класс «Person»**

namespace IndividualWork

{

abstract class Person

{

public Person(string fio) { this.fio = fio; }

protected string fio;

public abstract string greeting();

}

}

**Интерфейс «Ruling»**

namespace IndividualWork

{

interface Ruling

{

void ruling(int [] massiv);

}

}

**Класс «Ministr»**

namespace IndividualWork

{

class Ministr : Person

{

public Ministr(List<Accident> list, string fio):base(fio) {

this.list = list;

}

public override string greeting() {//Представление министра

return "Министр " + fio;

}

public List<Accident> list;

}

}

**Класс «Accident»**

namespace IndividualWork

{

class Accident

{

public Accident(string situation, int [,]counts)

{

this.situation = situation;

this.counts = counts;

}

public int [,] counts = new int[3,4];

public string situation;

}

}

**Класс «Sphere»**

namespace IndividualWork

{

class Sphere

{

private int \_count;

public int count { get { return \_count; } set { \_count = value; } }

private int \_effefficiency;

public int effefficiency { get { return \_effefficiency; } set { \_effefficiency = value; } }

public Sphere()

{

count = 50;

effefficiency = 100;

reserve = false;

}

private bool reserve;

public void setReserve()//Создать резерв

{

if (!reserve && (count > 50))

{

reserve = true;

count -= 25;

effefficiency += 10;

}

}

public void getReserve()//Отдать резерв

{

if (reserve)

{

reserve = false;

count += 25;

effefficiency -= 10;

}

}

public bool endGame()//Проверка на выход полей за границы

{

if (count == 0 || effefficiency == 0)

return true;

return false;

}

public void outOfRange()//Изменение значений сферы при их выходе за пределы

{

if (count < 0)

{

count = 0;

}

else if (count > 100)

{

count = 100;

}

if (effefficiency < 0)

effefficiency = 0;

else if (effefficiency > 400)

effefficiency = 400;

}

}

}

**Класс «Country»**

namespace IndividualWork

{

class Country: Ruling

{

public Sphere economicalSphere { get { return \_economicalSphere; } set { this.\_economicalSphere = value; } }

public Sphere militarySphere { get { return \_militarySphere; } set { this.\_militarySphere = value; } }

public Sphere socialSphere { get { return \_socialSphere; } set { this.\_socialSphere = value; } }

public Sphere ecologicalSphere { get { return \_ecologicalSphere; } set { this.\_ecologicalSphere = value; } }

private Sphere \_economicalSphere;

private Sphere \_militarySphere;

private Sphere \_socialSphere;

private Sphere \_ecologicalSphere;

public string nameOfCountry;

public Country(Sphere economicalSphere, Sphere militarySphere, Sphere socialSphere, Sphere ecologicalSphere, string nameOfCountry)

{

this.economicalSphere = economicalSphere;

this.ecologicalSphere = ecologicalSphere;

this.socialSphere = socialSphere;

this.militarySphere = militarySphere;

this.nameOfCountry = nameOfCountry;

}

public void ruling(int[] massiv)

{

switch (massiv[0])

{

case 0:

economicalSphere.count -= massiv[2];

economicalSphere.effefficiency -= massiv[3];

break;

case 1:

militarySphere.count -= massiv[2];

militarySphere.effefficiency -= massiv[3];

break;

case 2:

socialSphere.count -= massiv[2];

socialSphere.effefficiency -= massiv[3];

break;

case 3:

ecologicalSphere.count -= massiv[2];

ecologicalSphere.effefficiency -= massiv[3];

break;

}

switch (massiv[1])

{

case 0:

economicalSphere.count += Convert.ToInt32(Convert.ToDouble(massiv[2]) \* 0.8);

economicalSphere.effefficiency += Convert.ToInt32(Convert.ToDouble(massiv[3]) \* 0.8);

break;

case 1:

militarySphere.count += Convert.ToInt32(Convert.ToDouble(massiv[2]) \* 0.8);

militarySphere.effefficiency += Convert.ToInt32(Convert.ToDouble(massiv[3]) \* 0.8);

break;

case 2:

socialSphere.count += Convert.ToInt32(Convert.ToDouble(massiv[2]) \* 0.8);

socialSphere.effefficiency += Convert.ToInt32(Convert.ToDouble(massiv[3]) \* 0.8);

break;

case 3:

ecologicalSphere.count += Convert.ToInt32(Convert.ToDouble(massiv[2]) \* 0.8);

ecologicalSphere.effefficiency += Convert.ToInt32(Convert.ToDouble(massiv[3]) \* 0.8);

break;

}

}

public void changeCountSphere(int[,] massiv, int choice)//Изменение значений сферы в зависимости от выбора пользователя 0 - да, 1 - нет

{

if (choice==0)

economicalSphere.effefficiency += massiv[2, 0];

else

economicalSphere.effefficiency -= massiv[2, 0];

if (choice == 0)

militarySphere.effefficiency += massiv[2, 1];

else

militarySphere.effefficiency -= massiv[2, 1];

if (choice == 0)

socialSphere.effefficiency += massiv[2, 2];

else

socialSphere.effefficiency -= massiv[2, 2];

if (choice == 0)

ecologicalSphere.effefficiency += massiv[2, 2];

else

ecologicalSphere.effefficiency -= massiv[2, 3];

economicalSphere.count += Convert.ToInt32((Convert.ToDouble(massiv [choice, 0]) \* Convert.ToDouble(economicalSphere.effefficiency))/100) ;

militarySphere.count += Convert.ToInt32((Convert.ToDouble(massiv[choice, 1]) \* Convert.ToDouble(militarySphere.effefficiency)) / 100);

socialSphere.count += Convert.ToInt32((Convert.ToDouble(massiv[choice, 2]) \* Convert.ToDouble(socialSphere.effefficiency)) / 100);

ecologicalSphere.count += Convert.ToInt32((Convert.ToDouble(massiv[choice, 3]) \* Convert.ToDouble(ecologicalSphere.effefficiency)) / 100);

}

}

}

**Класс «Parlament»**

namespace IndividualWork

{

class Parlament: Country

{

public Ministr economicalMinistr;

public Ministr militaryMinistr;

public Ministr socialMinistr;

public Ministr ecologicalMinistr;

public Parlament(Sphere economicalSphere, Sphere militarySphere, Sphere socialSphere, Sphere ecologicalSphere, string nameOfCountry, Ministr economicalMinistr, Ministr militaryMinistr, Ministr socialMinistr, Ministr ecologicalMinistr): base(economicalSphere, militarySphere, socialSphere, ecologicalSphere, nameOfCountry)

{

this.economicalMinistr = economicalMinistr;

this.ecologicalMinistr = ecologicalMinistr;

this.militaryMinistr = militaryMinistr;

this.socialMinistr = socialMinistr;

}

}

}

**Класс «President»**

namespace IndividualWork

{

class President:Ministr, Ruling

{

Parlament parlament;

public President(List<Accident> list, Parlament parlament, string fio) : base(list, fio)

{

this.parlament = parlament;

}

public override string greeting()//Представление президента

{

return "Президент государства " + parlament.nameOfCountry;

}

public void ruling(int [] massiv) //Управление странной (изменение полей сферы)

{

switch (massiv[0])

{

case 0:

parlament.economicalSphere.count -= massiv[2];

break;

case 1:

parlament.militarySphere.count -= massiv[2];

break;

case 2:

parlament.socialSphere.count -= massiv[2];

break;

case 3:

parlament.ecologicalSphere.count -= massiv[2];

break;

}

switch (massiv[1])

{

case 0:

parlament.economicalSphere.count += Convert.ToInt32(Convert.ToDouble(massiv[2])\*0.9);

break;

case 1:

parlament.militarySphere.count += Convert.ToInt32(Convert.ToDouble(massiv[2]) \* 0.9);

break;

case 2:

parlament.socialSphere.count += Convert.ToInt32(Convert.ToDouble(massiv[2]) \* 0.9);

break;

case 3:

parlament.ecologicalSphere.count += Convert.ToInt32(Convert.ToDouble(massiv[2]) \* 0.9);

break;

}

}

public void offer(int num)//Создание обращение к презеденту в связи с проблемой

{

switch (num / 4) {

case 0://Обращение министра экономики

if (MessageBox.Show(

parlament.economicalMinistr.greeting() + parlament.economicalMinistr.list[num % 4].situation,

"Предложение президенту",

MessageBoxButtons.YesNo,

MessageBoxIcon.Information) == DialogResult.Yes)

{

parlament.changeCountSphere(parlament.economicalMinistr.list[num % 4].counts, 0);

}

else

{

parlament.changeCountSphere(parlament.economicalMinistr.list[num % 4].counts, 1);

}

break;

case 1://Обращение военного министра

if (MessageBox.Show(

parlament.militaryMinistr.greeting() + parlament.militaryMinistr.list[num % 4].situation,

"Предложение президенту",

MessageBoxButtons.YesNo,

MessageBoxIcon.Information) == DialogResult.Yes)

{

parlament.changeCountSphere(parlament.militaryMinistr.list[num % 4].counts, 0);

}

else

{

parlament.changeCountSphere(parlament.militaryMinistr.list[num % 4].counts, 1);

}

break;

case 2://Обращение социального министра

if (MessageBox.Show(

parlament.socialMinistr.greeting() + parlament.socialMinistr.list[num % 4].situation,

"Предложение президенту",

MessageBoxButtons.YesNo,

MessageBoxIcon.Information) == DialogResult.Yes)

{

parlament.changeCountSphere(parlament.socialMinistr.list[num % 4].counts, 0);

}

else

{

parlament.changeCountSphere(parlament.socialMinistr.list[num % 4].counts, 1);

}

break;

case 3://Обращение экологического министра

if (MessageBox.Show(

parlament.ecologicalMinistr.greeting() + parlament.ecologicalMinistr.list[num % 4].situation,

"Предложение президенту",

MessageBoxButtons.YesNo,

MessageBoxIcon.Information) == DialogResult.Yes)

{

parlament.changeCountSphere(parlament.ecologicalMinistr.list[num % 4].counts, 0);

}

else

{

parlament.changeCountSphere(parlament.ecologicalMinistr.list[num % 4].counts, 1);

}

break;

case 4://Решение собственных проблем

if (MessageBox.Show(

greeting() + list[num % 4].situation,

"Предложение президенту",

MessageBoxButtons.YesNo,

MessageBoxIcon.Information) == DialogResult.Yes)

{

parlament.changeCountSphere(list[num % 4].counts, 0);

}

else

{

parlament.changeCountSphere(list[num % 4].counts, 1);

}

break;

}

}

}

}

**Форма «Form1»**

public partial class Form1 : Form

{

//Создание всех объектов класса для запуска программы

Sphere economic = new Sphere();

Accident economicalSphere1;

Accident economicalSphere2;

Accident economicalSphere3;

Accident economicalSphere4;

List<Accident> listEconomical = new List<Accident>();

Ministr economicalMinistr;

Sphere military = new Sphere();

Accident militarySphere1;

Accident militarySphere2;

Accident militarySphere3;

Accident militarySphere4;

List<Accident> listMilitary = new List<Accident>();

Ministr militaryMinistr;

Sphere social = new Sphere();

Accident socialSphere1;

Accident socialSphere2;

Accident socialSphere3;

Accident socialSphere4;

List<Accident> listSocial = new List<Accident>();

Ministr socialMinistr;

Sphere ecological = new Sphere();

Accident ecologicalSphere1;

Accident ecologicalSphere2;

Accident ecologicalSphere3;

Accident ecologicalSphere4;

List<Accident> listEcological = new List<Accident>();

Ministr ecologicalMinistr;

Country country;

Parlament parlament;

Accident presidentSphere1;

Accident presidentSphere2;

Accident presidentSphere3;

Accident presidentSphere4;

List<Accident> listPresidetn = new List<Accident>();

President president;

List<int> massiv = new List<int>();

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e) //Инициализация объектов классов при создании формы

{

economic = new Sphere();

int[,] ecns1 = { { 10, 0, -5, 0 }, { -10, 0, 5, 0 }, { 10, 0, 0, 0 } };

economicalSphere1 = new Accident("Благосостояние народа улучшилось, предлагаю немного поднять налоги.", ecns1);

int[,] ecns2 = { { 25, 0, 0, 0 }, { 0, 10, 10, 10 }, { 0, 0, 0, 0 } };

economicalSphere2 = new Accident("Дела налаживаются, войны заканчиваются, предлагаю отложить деньги на потом.", ecns2);

int[,] ecns3 = { { 10, 0, 0, -5 }, { 25, 0, 0, 0 }, { 20, 0, 0, 0 } };

economicalSphere3 = new Accident("Было найдено большое месторождение нефти. Предлагаю построить больше заводов по ее переработке.", ecns3);

int[,] ecns4 = { { -10, 0, 10, 0 }, { 5, 0, -10, 0 }, { 10, 0, 10, 0 } };

economicalSphere4 = new Accident("Благосостояние народа ухудшается, нужно ввести пособия для малоимущих.", ecns4);

listEconomical.Add(economicalSphere1);

listEconomical.Add(economicalSphere2);

listEconomical.Add(economicalSphere3);

listEconomical.Add(economicalSphere4);

economicalMinistr = new Ministr(listEconomical, "экономической сферы, Петр Древов: ");

military = new Sphere();

int[,] ms1 = { { 0, -20, -10, 0 }, { 0, 5, 5, 0 }, { 0, 0, 0, 0 } };

militarySphere1 = new Accident("Соседние страны инвестируют в вооружение. Нанесем превентный удар!", ms1);

int[,] ms2 = { { -5, 10, 0, 0 }, { 0, -10, 5, 0 }, { 0, 0, 10, 0 } };

militarySphere2 = new Accident("У нас маленькая и слабая армия. Нам нужны новобранцы.", ms2);

int[,] ms3 = { { 0, 0, 0, -100 }, { 0, -10, 5, 0 }, { 0, 0, 0, 0 } };

militarySphere3 = new Accident("Я хочу протестировать новое биологическое оружие, но глава исследовательского отдела против, но с вашим разрешением он позволит его испытать.", ms3);

int[,] ms4 = { { -10, 20, 0, 0 }, { 5, -10, 0, 0 }, { 5, 0, 10, 0 } };

militarySphere4 = new Accident("Отдел исследований не справляется с работой. Им нужно больше ресурсов для разработки оружия!", ms4);

listMilitary.Add(militarySphere1);

listMilitary.Add(militarySphere2);

listMilitary.Add(militarySphere3);

listMilitary.Add(militarySphere4);

militaryMinistr = new Ministr(listMilitary, "военный сферы, Эльдо Туров: ");

social = new Sphere();

int[,] ss1 = { { -5, 0, 10, 0 }, { -5, 0, -5, 0 }, { 0, 0, 5, 0 } };

socialSphere1 = new Accident("Люди несчастливы, они требуют больше развлечений!", ss1);

int[,] ss2 = { { -5, 0, 10, 0 }, { 5, 0, -5, 0 }, { 0, 0, 10, 0 } };

socialSphere2 = new Accident("Некоторые законы устарели. Нужно их пересмотреть.", ss2);

int[,] ss3 = { { -10, 0, 10, 5 }, { 5, 0, -10, 0 }, { 5, 0, 10, 0 } };

socialSphere3 = new Accident("Хотелось бы, чтобы государство лучше финансировало образование.", ss3);

int[,] ss4 = { { -10, 0, 10, 0 }, { 5, 0, -20, 0 }, { 5, 0, 10, 0 } };

socialSphere4 = new Accident("Люди на заводах отказываются работать и требуют повышение.", ss4);

listSocial.Add(socialSphere1);

listSocial.Add(socialSphere2);

listSocial.Add(socialSphere3);

listSocial.Add(socialSphere4);

socialMinistr = new Ministr(listSocial, "социальный сферы, Александр Строгов: ");

ecological = new Sphere();

int[,] ecls1 = { { -10, 0, 5, 10 }, { 5, 0, -5, -10 }, { 0, 0, 0, 10 } };

ecologicalSphere1 = new Accident("Соседние страны инвестируют в экологию, нам стоит поступить также.", ecls1);

int[,] ecls2 = { { -5, 0, 5, 10 }, { 10, 0, -5, -20 }, { 0, 0, 0, 10 } };

ecologicalSphere2 = new Accident("Было обнаружено, что одна влиятельная компания сливает токсичные отходы в реку. Надо с этим разобраться!", ecls2);

int[,] ecls3 = { { -10, 0, 0, 10 }, { 5, 0, 0, -10 }, { 0, 0, 0, 10 } };

ecologicalSphere3 = new Accident("Новая болезнь у животных поставила под угрозу животноводство. Многие фермы испытывают неудобство. Закрыть их на карантин?", ecls3);

int[,] ecls4 = { { -10, -10, 10, 10 }, { 5, 0, -10, -10 }, { 0, 0, 0, 10 } };

ecologicalSphere4 = new Accident("В некоторых городах продают продукты выращенные на зараженных территориях. Это очень опасно!", ecls4);

listEcological.Add(ecologicalSphere1);

listEcological.Add(ecologicalSphere2);

listEcological.Add(ecologicalSphere3);

listEcological.Add(ecologicalSphere4);

ecologicalMinistr = new Ministr(listEcological, "экологической сферы, Николай Грязнов: ");

country = new Country(economic, military, social, ecological, "Московия");

parlament = new Parlament(economic, military, social, ecological, "Московия", economicalMinistr, militaryMinistr, socialMinistr, ecologicalMinistr);

int[,] pres1 = { { -10, 5, 5, 5 }, { 10, 0, 0, 0 }, { 0, 0, 0, 0 } };

presidentSphere1 = new Accident(". На этой неделе у вас день рождение, стоит ли праздновать?", pres1);

int[,] pres2 = { { 0, -30, -30, 0 }, { 0, 10, 10, 0 }, { 0, 0, 0, 0 } };

presidentSphere2 = new Accident(". Садовник с вашей дачи попросился в генералы, сказал что в школе был старостой, назначить его в генералы?", pres2);

int[,] pres3 = { { -10, -5, -5, 0 }, { 5, 0, -30, 0 }, { 0, 0, 0, 0 } };

presidentSphere3 = new Accident(". В северных регионах страны произошло землетрясение, направить туда спасательные отряды?", pres3);

int[,] pres4 = { { 0, 0, -100, 0 }, { 0, 5, 5, 0 }, { 0, 0, 0, 0 } };

presidentSphere4 = new Accident(". Вам пришло анонимное письмо с угрозами и требованием вашей отставки. Уйти с поста президента?", pres4);

listPresidetn.Add(presidentSphere1);

listPresidetn.Add(presidentSphere2);

listPresidetn.Add(presidentSphere3);

listPresidetn.Add(presidentSphere4);

president = new President(listPresidetn, parlament, "В. В. Прутин");

for (int i = 0; i < 20; i++)

{

massiv.Add(i);

}

updateProgressBar();

}

private void updateProgressBar() //Обновление ProgressBar'ов и проверка выхода значения полей за границы

{

parlament.economicalSphere.outOfRange();

progressBar1.Value = parlament.economicalSphere.count;

progressBar5.Value = parlament.economicalSphere.effefficiency;

parlament.militarySphere.outOfRange();

progressBar2.Value = parlament.militarySphere.count;

progressBar6.Value = parlament.militarySphere.effefficiency;

parlament.socialSphere.outOfRange();

progressBar3.Value = parlament.socialSphere.count;

progressBar7.Value = parlament.socialSphere.effefficiency;

parlament.ecologicalSphere.outOfRange();

progressBar4.Value = parlament.ecologicalSphere.count;

progressBar8.Value = parlament.ecologicalSphere.effefficiency;

isOut();

}

private void isOut()//Проверка выхода значений полей за границы и сообщения соответствующие выходу значения опреденной сферы.

{

if (parlament.economicalSphere.endGame())//При выходе значний экономической сферы

{

MessageBox.Show(

"Вашу голову продали лидерам других стран из-за ваших многочисленных долгов!",

"Сообщение президенту",

MessageBoxButtons.OK,

MessageBoxIcon.Error);

this.Close();

}

if (parlament.militarySphere.endGame())//При выходе значний военной сферы

{

MessageBox.Show(

"На вас напали враги, через неделю вас и генерала окружили враги, бежать некуда!",

"Сообщение президенту",

MessageBoxButtons.OK,

MessageBoxIcon.Error);

this.Close();

}

if (parlament.socialSphere.endGame())//При выходе значний социальной сферы

{

MessageBox.Show(

"Народ, недовольный вашими решениями, вышел на улицу и штурмует белый дом. После успешнего переворота Наковальный приходит к власти и ведет государственные дела под командованием Госдепа.",

"Сообщение президенту",

MessageBoxButtons.OK,

MessageBoxIcon.Error);

this.Close();

}

if (parlament.ecologicalSphere.endGame())//При выходе значний экологической сферы

{

MessageBox.Show(

"Вся территория заражена, жить негде, вся страна обречена на мучительную смерть!",

"Сообщение президенту",

MessageBoxButtons.OK,

MessageBoxIcon.Error);

this.Close();

}

}

private void button1\_Click(object sender, EventArgs e)//Решение проблем в гос. сферах и вывод сообщение при полном из решении

{

if (massiv.Count == 0)

{//вывод сообщение при полном из решении

MessageBox.Show(

"Вы решили большинство проблем страны и удержали в ней власть, отныне дела будут налаживаться и вы приведете страну к процветанию",

"Поздравление о победе",

MessageBoxButtons.OK,

MessageBoxIcon.Information

);

Application.Exit();

}

else

{//Решение проблем в гос. сферах

Random rand = new Random();//выбор случайного числа для уникальных проблем сферы

int buff = rand.Next(massiv.Count);

president.offer(massiv[buff]);

massiv.RemoveAt(buff);

updateProgressBar();

}

}

private void button2\_Click(object sender, EventArgs e)//Распределение бюджета между сферами

{

int [] massiv = new int [3];

massiv[0] = comboBox1.SelectedIndex;//1-я сфера

massiv[1] = comboBox2.SelectedIndex;//2-я сфера

massiv[2] = Convert.ToInt32(numericUpDown1.Value);//Значение, на которое будем менять

president.ruling(massiv);//Изменение

updateProgressBar();

comboBox1.SelectedIndex = -1;//Обновление элементов формы

comboBox2.SelectedIndex = -1;

numericUpDown1.Value = 0;

}

private void button3\_Click(object sender, EventArgs e)//Инвестирование в сферы

{

int[] massiv = new int[4];

massiv[0] = comboBox3.SelectedIndex;//1-я сфера

massiv[1] = comboBox4.SelectedIndex;//2-я сфера

massiv[2] = Convert.ToInt32(numericUpDown2.Value);//Значение показателя, на которое будем менять

massiv[3] = Convert.ToInt32(numericUpDown3.Value);//Значение коэффициента, на которое будем менять

country.ruling(massiv);

updateProgressBar();

comboBox3.SelectedIndex = -1;//Обновление элементов формы

comboBox4.SelectedIndex = -1;

numericUpDown2.Value = 0;

numericUpDown3.Value = 0;

}

private void button4\_Click(object sender, EventArgs e)//Создание резерва для выбранной сферы

{

switch (comboBox5.SelectedIndex)

{

case 0:

economic.setReserve();

break;

case 1:

military.setReserve();

break;

case 2:

social.setReserve();

break;

case 3:

ecological.setReserve();

break;

}

updateProgressBar();

}

private void button5\_Click(object sender, EventArgs e)//Забрать резерв выбранной формы

{

switch (comboBox5.SelectedIndex)

{

case 0:

economic.getReserve();

break;

case 1:

military.getReserve();

break;

case 2:

social.getReserve();

break;

case 3:

ecological.getReserve();

break;

}

updateProgressBar();

}

}