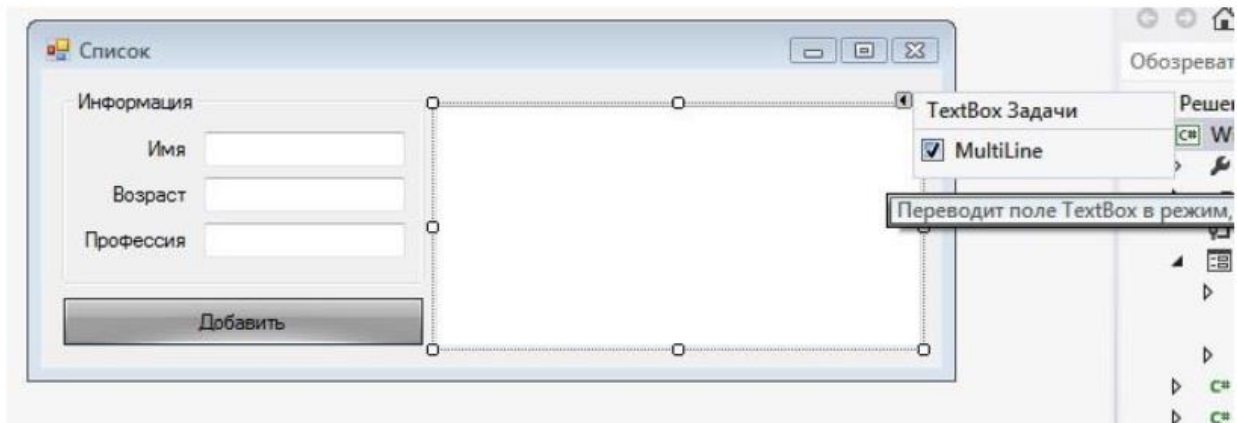


Разработка приложения Windows Forms с использованием классов

Пример

- 1) Создать приложение Windows Forms, ввести имя проекта.
- 2) В форме Form1 разместить четыре элемента TextBox, один элемент GroupBox, один элемент Button, три элемента Label.
- 3) Сгруппировать элементы так, как показано на рисунке, изменить значение поля Text элемента GroupBox1 на «Информация», сменить значение поля Name элементов TextBox1, TextBox2, TextBox3 на NameBox, AgeBox, ProfessionBox и ResultBox соответственно. Сменить значение поля Name элемента Button1 на StartButton
- 4) Поместить Элементы NameBox, AgeBox, ProfessionBox в GroupBox1, подписать их с помощью элементов Label1, Label2 и Label3(сменив значение поля Text элементов Label). Элемент ResultBox перевести в режим MultiLine (рисунок).
- 5) Значение поля Text элемента StartButton сменить на «Добавить».



Разработка класса Person

- 1) В окне Обозревателя Решений щелкнуть правой кнопкой по имени проекта, затем Добавить -> Класс. В появившемся окне ввести имя класса (Person) и нажать кнопку Добавить.
- 2) Объявить переменные name и profession типа string и переменную age типа integer:
`string name;`
`int age;`
`string profession;`
- 3) Объявить конструктор класса Person, который будет принимать значение поля name:
`public Person(string name)`
`{`
`this.name = name;`
`}`
- 4) Перегрузить этот конструктор для различных параметров, которые могут передаваться при создании экземпляра класса.

Пример конструктора, принимающего значения полей name и age:

```
public Person(string name, int age)
{
    this.name = name;
    this.age = age;
}
```

Пример конструктора, принимающего значения полей name и profession:

```
public Person(string name, string profession)
{
    this.name = name;
    this.profession = profession;
}
```

Пример конструктора, принимающего значения полей name, age и profession:

```
public Person(string name, int age, string profession)
{
    this.name = name;
    this.age = age;
    this.profession = profession;
}
```

```
}
```

Взаимосвязь класса с формой

1) Дважды щелкнуть по элементу StartButton, откроется код обработчика события нажатия кнопки

```
private void button1_Click(object sender, EventArgs e)
```

2) Объявить экземпляр класса Person, не вызывая при этом конструктор (необходимо, чтобы работать с объектом впоследствии, вызывая различные перегрузки конструктора для конкретной ситуации):

```
Person new_person;
```

3) Реализовать проверку на наличие текста в NameBox: если текста нет, то выводим сообщение пользователю в диалоговом окне MessageBox:

```
if (NameBox.Text == "")
```

```
{
```

```
    MessageBox.Show("Введите имя");
```

```
}
```

4) Реализовать конструкцию из условий, проверяющих либо наличие, либо отсутствие текста в элементах управления TextBox, в результате выполнения условий, явно вызывать одну из перегрузок конструктора класса Person:

```
if (AgeBox.Text != "")
```

```
{
```

```
    if (ProfessionBox.Text != "")
```

```
    {
```

```
        new_person = new Person(NameBox.Text, Convert.ToInt32(AgeBox.Text), ProfessionBox.Text);
```

```
    }
```

```
    else
```

```
    {
```

```
        new_person = new Person(NameBox.Text, Convert.ToInt32(AgeBox.Text));
```

```
    }
```

```
}
```

```
else
```

```
{
```

```
    if (ProfessionBox.Text != "")
```

```
    {
```

```
        new_person = new Person(NameBox.Text, 0, ProfessionBox.Text);
```

```
    }
```

```
    else
```

```
    {
```

```
        new_person = new Person(NameBox.Text);
```

```
    }
```

```
}
```

Вывод результатов

1) Для вывода результатов работы программы, необходимо использовать метод GetInformation() класса Person, задача которого будет выводить строку, в которой будет содержаться текущая информация об экземпляре класса: поля name, age, profession (рисунок 2).

```
public string GetInformation()
```

```
{
```

```
    string information;
```

```
    information = "Имя: " + this.name + "; Возраст: " + this.age.ToString() + "; Профессия: " +
```

```
this.profession;
```

```
    return information;
```

```
}
```

Синтаксис в коде:

```
ResultBox += new_person.GetInformation();
```

Список

Информация

Имя: Евгений Эдуардович

Возраст: 47

Профессия: Преподаватель

Добавить

Имя: Сергей; Возраст: 19; Профессия: Студент
Имя: Игорь; Возраст: 25; Профессия: Автомеханик
Имя: Евгений Эдуардович; Возраст: 47; Профессия: Преподаватель

Задание для выполнения работы

Каждый разрабатываемый класс должен содержать следующие элементы: скрытые поля, конструкторы с параметрами и без параметров, методы, свойства. Методы и свойства должны обеспечивать непротиворечивый, полный, минимальный и удобный интерфейс класса.

В программе должна выполняться проверка всех разработанных элементов класса.

Задание 1.

Описать класс «компьютер», содержащий сведения о модели, тактовой частоте, марке видеокарты, объеме жесткого диска, объеме оперативной памяти и стоимости. Предусмотреть инициализацию с проверкой допустимости значений полей. Описать свойства для получения состояния объекта.

Задание 2.

Описать класс «комната», содержащий сведения о длине и ширине комнаты, высоте потолков и количестве окон. Предусмотреть инициализацию с проверкой допустимости значений полей. Описать методы вычисления площади и объема комнаты и свойства для получения состояния объекта.

Описать класс «квартира», включающий адрес квартиры, количество комнат и массив элементов класса «комната». Предусмотреть инициализацию с проверкой допустимости значений поля «адрес квартиры» и количество комнат. Создать конструктор с параметрами для инициализации адреса квартиры, количества комнат и выделения памяти под массив комнат для этого количества комнат (`массив_комнат = new комната[количество комнат];`). Описать метод для добавления комнаты с параметрами: номер комнаты и ее характеристики: длина, ширина, высота, количество окон. Этот метод создает объект массива комнат с номером комнаты и вызывает конструктор класса «комната» (`массив_комнат[i] = new комната(длина, ширина, высота, количество окон);`). Описать свойства для получения состояния объекта.

Создать приложение Windows Forms, содержащее поля для ввода информации о квартире и комнатах, элемент DataGridView для отображения информации о квартирах, их комнатах и вычисленных площадях и объеме комнат, кнопки для добавления квартиры и добавления информации о комнате.