# Практическая работа

## Создание проекта с использованием компонентов для работы с текстом

## 1. Цель работы. Получение навыков составления программ в среде Visual Studio.Net

## 2. Литература. Т.А.Павловская “С#. Программирование на языке высокого уровня”, И.Г.Семакин, А.П.Шестаков «Основы программирования», Т.А.Павловская, Ю.А.Щупак «С/С++. Структурное программирование. Практикум», Е.Л.Шиманович «С/С++ в примерах и задачах»

## 3. Подготовка к работе. Повторить конспект лекций по темам «Основные принципы ООП».

## 4. Перечень оборудования. Интегрированная среда Visual Studio.Net.

## 5. Задание.

1. Научиться писать приложения в интегрированной среде Visual Studio.Net.

***6. Теоретический материал***

**1. Button**

Элемент Button представляет обычную кнопку:

<Button x:Name="button1" Width="60" Height="30" Background="LightGray" />

Чтобы связать кнопку с обработчиком события нажатия, нам надо определить в самой кнопке атрибут Click. А значением этого атрибута будет название обработчика в коде C#. А затем в самом коде C# определить этот обработчик.

<Button x:Name="button1" Width="60" Height="30" Content="Нажать" Click="Button\_Click" />

И обработчик в коде C# генерируется автоматически:

private void Button\_Click(object sender, RoutedEventArgs e)

{

    MessageBox.Show("Кнопка нажата");

}

Кнопка имеет такие свойства как **IsDefault** и **IsCancel**, которые принимают значения true и false.

Если свойство IsDefault установлено в true, то при нажатии клавиши Enter будет вызываться обработчик нажатия этой кнопки.

Аналогично если свойство IsCancel будет установлено в true, то при нажатии на клавишу Esc будет вызываться обработчик нажатия этой кнопки.

 <StackPanel>

        <Button x:Name="acceptButton" Content="ОК" IsDefault="True" Click="acceptButton\_Click" />

        <Button x:Name="escButton" Content="Выход" IsCancel="True" Click="escButton\_Click" />

    </StackPanel>

 public partial class MainWindow : Window

    {

        public MainWindow()

        {

            InitializeComponent();

        }

        private void acceptButton\_Click(object sender, RoutedEventArgs e)

        {

            MessageBox.Show("Действие выполнено");

        }

        private void escButton\_Click(object sender, RoutedEventArgs e)

        {

            this.Close(); // закрытие окна

        }

    }

Теперь при нажатии на клавишу Enter будет отображаться сообщение, а при нажатии на Esc будет происходить выход из приложения и закрытие окна.

1. Класс **ComboBox** представляет выпадающий список.

**Событие SelectionChanged и обработка выбора объекта**

Обрабатывая событие **SelectionChanged**, мы можем динамически получать выделенный элемент. Для управления выбранными объектами класс ComboBox предоставляет следующие свойства:

* **SelectedItem**: выбранный объект списка
* **SelectedValue**: выбранное значение
* **SelectedIndex**: индекс выбранного объекта в списке

**3. DatePicker** - текстовое поле для ввода даты с выпадающим календарем после ввода.

Свойства:

|  |  |
| --- | --- |
| **DisplayDateStart** и **DisplayDateEnd** | Задают соответственно начальную и конечную дату диапазона, который будет отображаться в календаре. |
| **IsTodayHighlighted** | Отмечает, будет ли выделена текущая дата |
| **SelectedDate** | Задает выделенную дату |
| **FirstDayOfWeek** | Задает первый день недели |

**DatePicker**

Кроме выше названных свойств элемент DatePicker определяет еще ряд свойств, среди которых **IsDropDownOpen** (показывает, будет ли связанный с элементом всплывающий календарь оставаться открытым после выбора даты) и **SelectedDateFormat** (формат даты, принимает значение Short - формат в виде дд.мм.гггг или мм.дд.гггг и значение Long - с полными названиями месяцев):

<DatePicker Name="datePicker1" IsDropDownOpen="True" SelectedDateFormat="Short" />

Чтобы использовать в программе выбор даты пользователем, мы можем обработать событие SelectedDateChanged в коде c#:

private void calendar\_SelectedDatesChanged(object sender, SelectionChangedEventArgs e)

{

    DateTime? selectedDate = datePicker1.SelectedDate;

    MessageBox.Show(selectedDate.Value.Date.ToShortDateString());

}

***7. Порядок выполнения работы***.

**6.1** Компьютерный клуб "Фортуна" предоставляет машинное время с доступом в Интернет и к информационным ресурсам пользователям – физическим лицам. Доступ предоставляется после прохождения процедуры авторизации пользователя на портале клуба. Для получения пары логин/пароль пользователю необходимо ввести свое ИМЯ и ДАТУ РОЖДЕНИЯ.

Пользователь может вводить ИМЯ либо на английском, либо на русском языке. ДАТА РОЖДЕНИЯ вводится в формате ДД.ММ.ГГГГ.

Требуется разработать программный модуль, который будет генерировать пару ЛОГИН/ПАРОЛЬ на основании введенных пользователем данных по следующим требованиям:

ЛОГИН:

− каждый символ имени пользователя заменяется на номер

позиции символа в национальном алфавите;

− в конце добавляется число, соответствующее сумме цифр

даты рождения (Д+Д+М+М+Г+Г+Г+Г).

ПАРОЛЬ:

− длина пароля равна 10 символов;

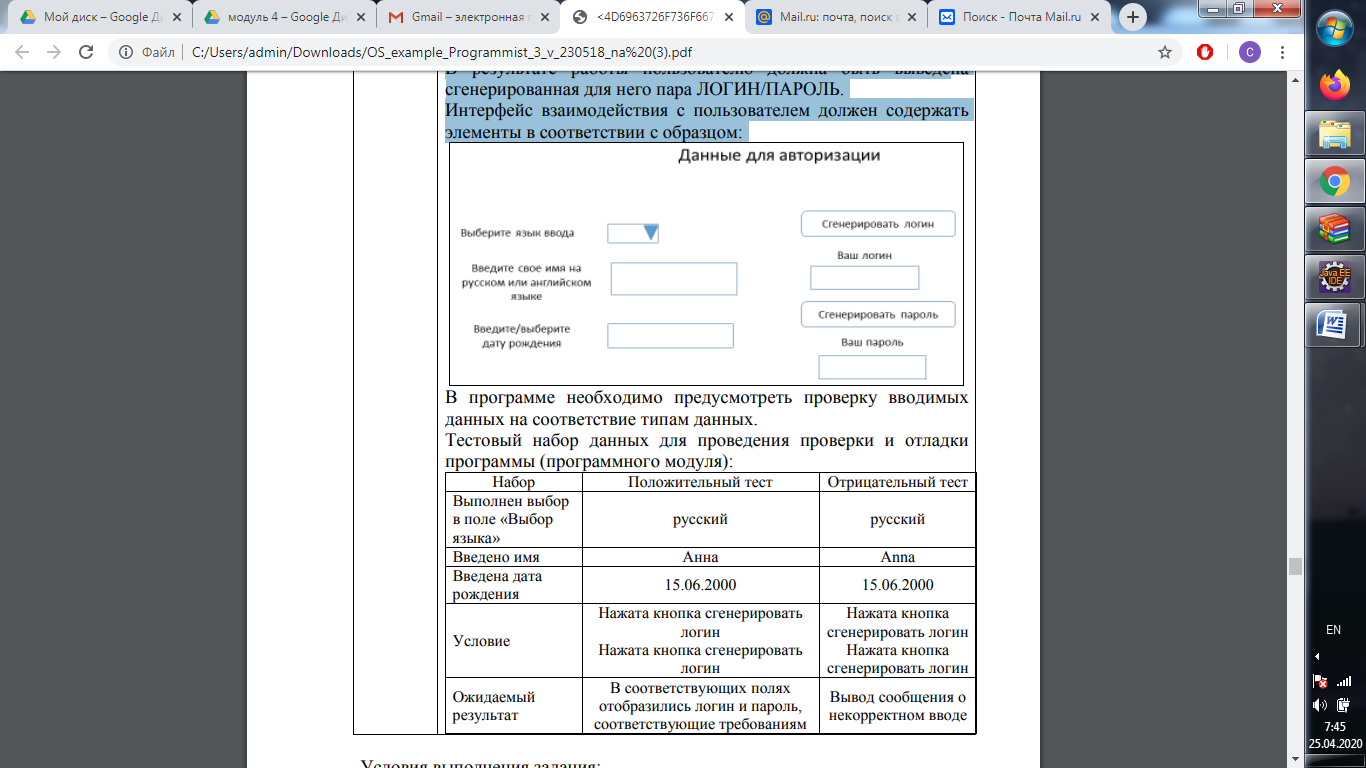
− в пароле случайным образом должен быть использован один раз любой из следующих символов:

-;+\_="[-@#$%^&?\*\*)(!];

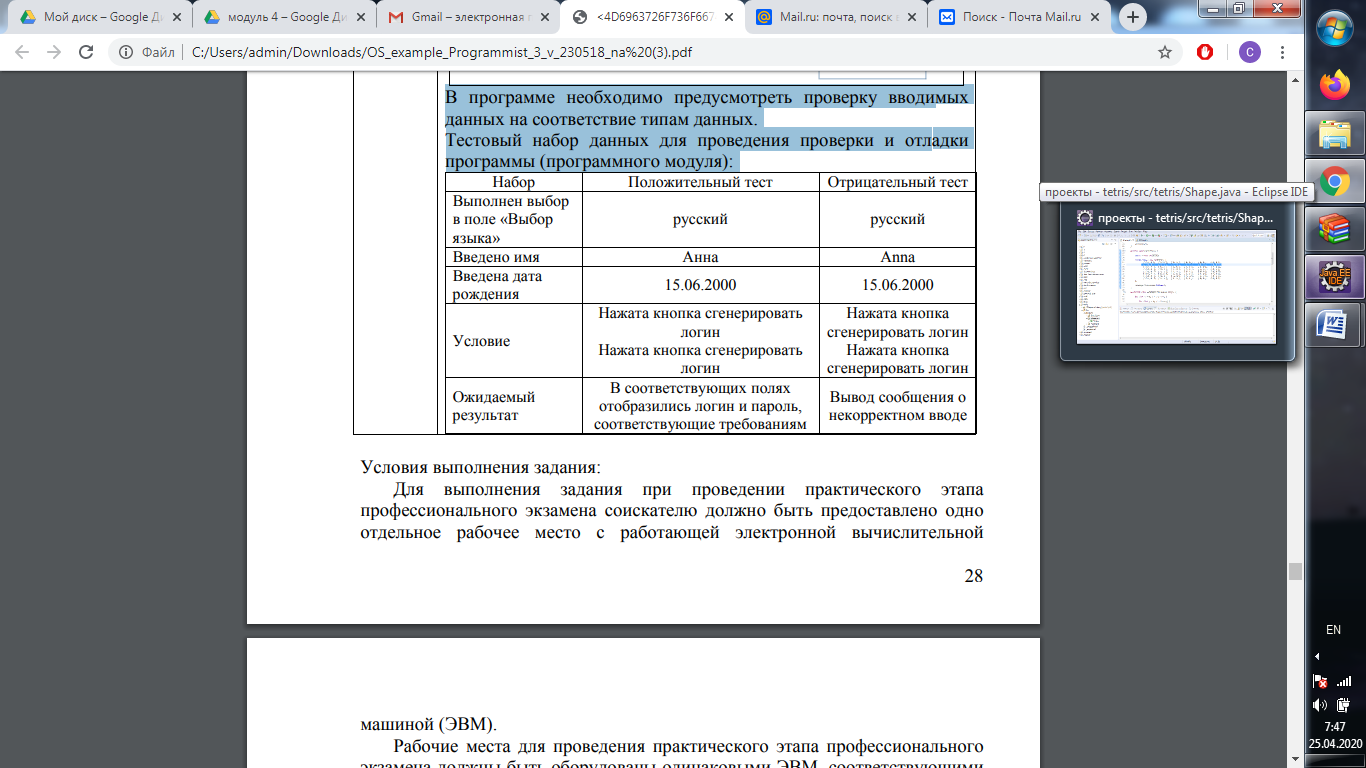
− пароль должен содержать не более пяти случайных цифр;

− любые две цифры в пароле подряд недопустимы;

− пароль должен содержать хотя бы одну заглавную букву, остальные прописные буквы английского алфавита. В результате работы пользователю должна быть выведена сгенерированная для него пара ЛОГИН/ПАРОЛЬ. Интерфейс взаимодействия с пользователем должен содержать элементы в соответствии с образцом:



В программе необходимо предусмотреть проверку вводимых данных на соответствие типам данных. Тестовый набор данных для проведения проверки и отладки программы (программного модуля):



***Содержание отчета.***

7.1. Название и цель практической работы.

7.2. Составленные тексты программ задач с ответами в рабочей тетради.

## 8. Приложение.