Государственное образовательное учреждение  
Высшего профессионального образования  
САНКТ-ПЕТЕРБУРГСКИЙНАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

**Отчет**

**по лабораторной работе**

тема: Разработка структурной модели – диаграммы классов

по предмету: Разработка программного обеспечения инфокоммуникационных систем

**Выполнил** студент гр. К3420

Сахаров Владислав Вадимович

**Дата** 5 апреля 2021

**Проверил** Осипов Никита Алексеевич

Санкт-Петербург

2020 г.

Цель

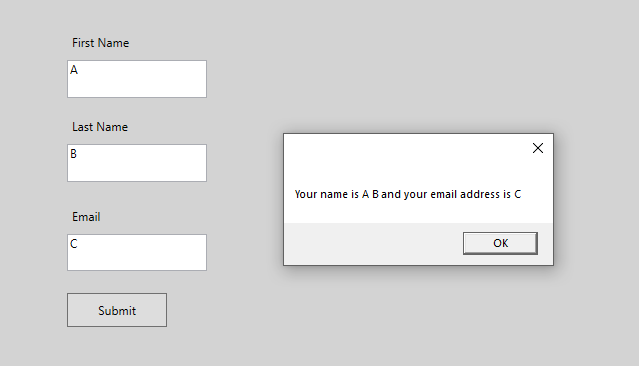
Выполнить ряд практических заданий для дальнейшего ознакомления со технологией WPF.

Решение

# Практическое занятие 4. Разработка пользовательского интерфейса

## Упражнение 1. Использование простых элементов управления.

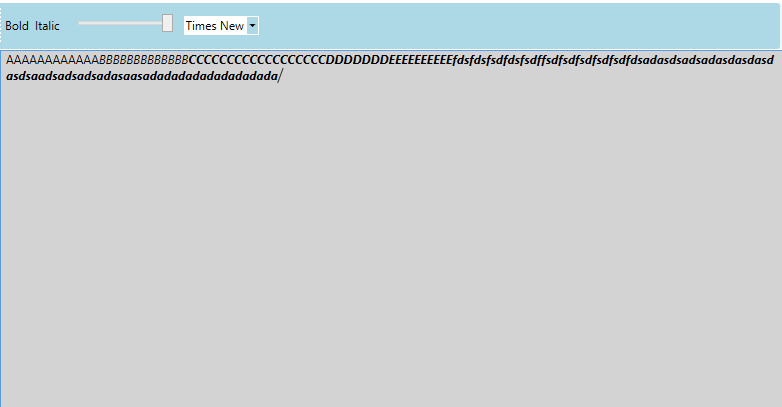
Создается проект *UserIn1*, в котором возможно переключение между текстовыми полями с помощью клавиши Tab и подчеркнутой буквы в надписи.



1. Вид приложения

## Упражнение 2. Размещение связанных элементов.

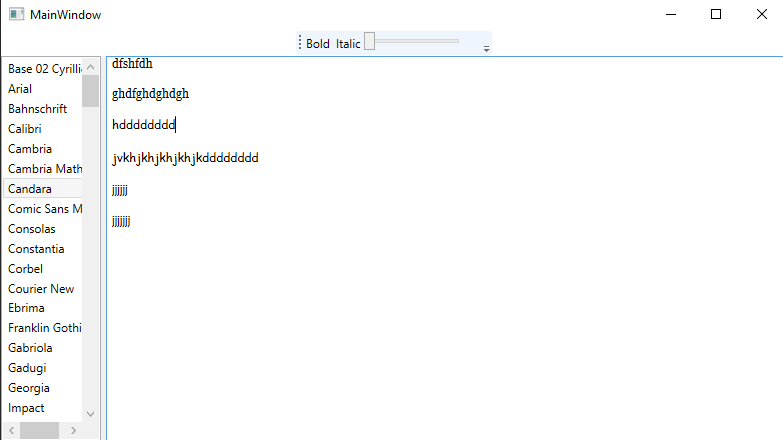
Выполняется создание проекта *UserIn2,* в котором используется *toolbar* для редактирования текста, введенного пользователем.



1. Вид проекта

## Упражнение 3. Работа с элементами управления раскладкой

Выполняется проект, дополняющий предыдущий возможностью определять, сколько в ширину будет текст, а сколько – список стилей.



1. Вид проекта

# Практическое занятие 5. Использование компонента BackgroundWorker

## Упражнение. Работа с компонентом BackgroundWorker

Создается проект, в котором используется BackgroundWorker, работающий как отдельный поток. Сам проект выполняет 50 циклов, после чего прекращает работу.

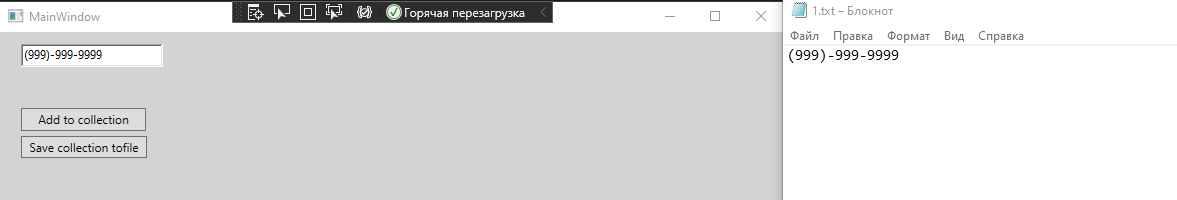


1. Преждевременная остановка работы

# Практическое занятие 6. Настройка интерфейса пользователя

## Упражнение 1. Использование элементов Windows Forms

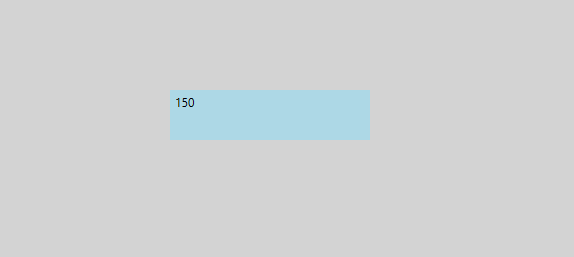
Выполняется создание проекта *UserForms*, в котором есть возможность дополнять список вводимыми в поле с маской номерами, а после – сохранить список в файл по выбору пользователя.



1. Пример работы

## Упражнение 2. Построение специального свойства зависимости

Создается проект *WpfUserControl* с анимацией изменения числа от 100 до 200 с помощью класса DependencyProperty. Данный класс позволяет устанавливать различные функции обратного вызова к свойству. В проекте используется две – проверка значения числа и выставление значения числа в ячейке при его изменении.

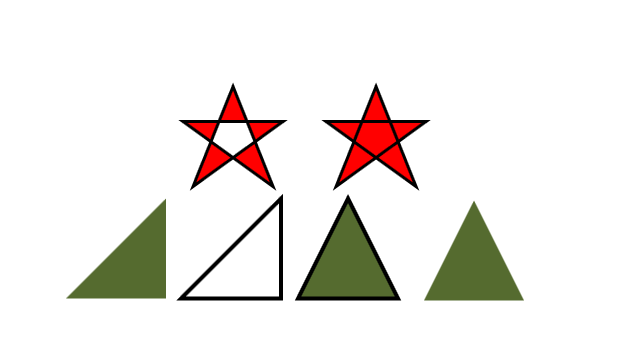


1. Вид приложения

# Практическое занятие 7. Графика в приложениях WPF

## Упражнение 1. Рисование замкнутой фигуры с помощью элемента "Многоугольник"

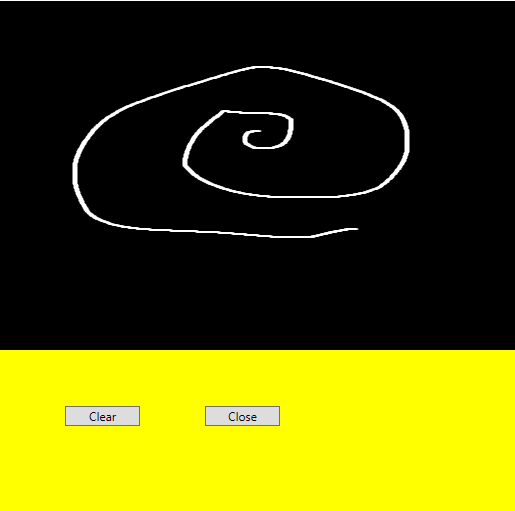
Создается проект *WpfApplicationPolygon*, в котором выполняется визуализация фигур различных форм средствами WPF.



1. Вид приложения

## Упражнение 2. Создание приложения рисования

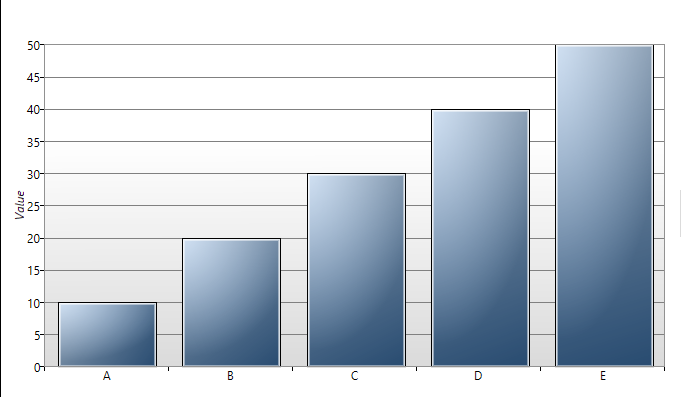
Создается проект *Ink Pad*, в котором используется InkCanvas, с помощью которого пользователь может рисовать.



1. Пример использования

## Упражнение 3. Создание диаграмм с использованием WPF Toolkit

Выполняется создание проекта *WpfApplicationChart*, в котором используется Chart для визуализации диаграммы, показывающей генерируемые данные классом *Model*.



1. Вид приложения.

Вывод

В ходе выполнения данной работы были изучены возможностью по предоставлению средств редактирования, смены фокуса комбинацией клавиш, использованию фонового процесса, масок ввода и анимации, а также средств визуализации, в том числе редактируемой в режиме реального времени.

Смена фокуса выполняет ту же роль, что и команды – позволяет не переключаться на мышь ради выполнения некоторых действий, сохраняя время пользователя.

BackgrounWorker – прекрасный инструмент для использования потоков без изучения их синтаксиса и особенностей.

Для отображения различной графики требуется установка дополнительных пакетов, которая полностью окупается выдаваемым функционалом – пользователь всегда рад наглядности данных в приложении.