

МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Автоматизированные системы управления»

# ВИЗУАЛЬНЫЕ СРЕДСТВА РАЗРАБОТКИ ПРОГРАММНЫХ ПРИЛОЖЕНИЙ

*Методические рекомендации к лабораторным работам для студентов  
специальности 1-40 05 01 «Информационные системы и технологии  
(по направлениям)» дневной и заочной форм обучения*



Могилев 2022

УДК 004.4  
ББК 32.973.26-018.2  
В42

Рекомендовано к изданию  
учебно-методическим отделом  
Белорусско-Российского университета

Одобрено кафедрой «Автоматизированные системы управления»  
«8» ноября 2021 г., протокол № 5

Составитель ст. преподаватель Л. А. Тимашкова

Рецензент канд. техн. наук, доц. И. В. Лесковец

Методические рекомендации к лабораторным работам по дисциплине «Визуальные средства разработки программных приложений» предназначены для студентов дневной и заочной форм обучения.

Учебно-методическое издание

## ВИЗУАЛЬНЫЕ СРЕДСТВА РАЗРАБОТКИ ПРОГРАММНЫХ ПРИЛОЖЕНИЙ

Ответственный за выпуск	А. И. Якимов
Корректор	Т. А. Рыжикова
Компьютерная верстка	Е. В. Ковалевская

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.  
Печать трафаретная. Усл. печ. л. . Уч.-изд. л. . Тираж 26 экз. Заказ №

Издатель и полиграфическое исполнение:  
Межгосударственное образовательное учреждение высшего образования  
«Белорусско-Российский университет».  
Свидетельство о государственной регистрации издателя,  
изготовителя, распространителя печатных изданий  
№ 1/156 от 07.03.2019.  
Пр-т Мира, 43, 212022, г. Могилев.

© Белорусско-Российский  
университет, 2022

## Содержание

Введение .....	5
1 Лабораторная работа № 1. Знакомство со средой визуальной разработки приложений .....	6
2 Лабораторная работа № 2. Использование компонентов и настройка их свойств. Обработка событий .....	8
3 Лабораторная работа № 3. Программирование разветвляющихся алгоритмов .....	9
4 Лабораторная работа № 4. Обработка одномерных массивов. Программирование циклических алгоритмов .....	12
5 Лабораторная работа № 5. Обработка двумерных массивов.....	15
6 Лабораторная работа № 6. Работа со строками .....	16
7 Лабораторная работа № 7. Разработка форм в Windows Forms .....	17
8 Лабораторная работа № 8. Работа с элементами управления. Панели и контейнеры.....	18
9 Лабораторная работа № 9. Создание меню и вкладок. Многооконный интерфейс .....	21
10 Лабораторная работа № 10. Классы, свойства, индексы .....	24
11 Лабораторная работа № 11. Наследование. Перегрузка методов и операторов. Полиморфизм .....	25
12 Лабораторная работа № 12. Исключительные ситуации. Обработка исключений .....	26
13 Лабораторная работа № 13. Делегаты и события .....	27
14 Лабораторная работа № 14. Интерфейсы .....	28
15 Лабораторная работа № 15. Обобщения. Классы-коллекции.....	28
16 Лабораторная работа № 16. Работа со строками. Использование регулярных выражений.....	29
17 Лабораторная работа № 17. Использование LINQ для работы с данными .....	30
18 Лабораторная работа № 18. Классы для работы с файлами. Сериализация .....	31
19 Лабораторная работа № 19. Отображение массивов и баз данных в таблицах.....	31
20 Лабораторная работа № 20. Построение графиков и диаграмм.....	32
21 Лабораторная работа № 21. Рисование в Windows Forms .....	36
22 Лабораторная работа № 22. Анимация с помощью таймера .....	37

23 Лабораторная работа № 23. Диспетчеры компоновки .....	38
24 Лабораторная работа № 24. Анимация в WPF-приложениях.....	43
25 Лабораторная работа № 25. Триггеры в WPF-приложениях.....	44
26 Лабораторная работа № 26. Использование фигур и кистей в WPF-приложениях.....	45
Список литературы .....	47

## Введение

Цель методических рекомендаций к выполнению лабораторных работ по дисциплине «Визуальные средства разработки программных приложений» заключается в овладении студентами практическими навыками разработки приложений на языке программирования C# в среде Microsoft Visual Studio и создания пользовательского интерфейса Windows-приложений.

Цель изучения дисциплины – изучение современных технологий разработки алгоритмов и программ с использованием методов объектно-ориентированного программирования, разработки визуальных интерфейсов, методов тестирования, отладки и решения задач, подготовка студентов к использованию современных технологий и программных средств как профессионального инструмента для решения научных и практических задач.

Для приобретения практических навыков программирования и использования компьютера студенту необходимо самостоятельно разработать алгоритмы будущих приложений, разработать для них GUI-интерфейс, отладить и получить решения предусмотренных в лабораторных работах задач.

Методические рекомендации предназначены для выполнения лабораторных работ в компьютерном классе, а также для получения практических навыков разработки Windows-приложений с использованием интегрированной среды Visual Studio на современном языке программирования C#. Выполнение рассмотренных примеров позволит студентам выработать практические навыки в создании таких приложений.

# 1 Лабораторная работа № 1. Знакомство со средой визуальной разработки приложений

**Цель работы:** изучить интегрированную среду Visual Studio; освоить простейшую структуру программы на языке C#; получить навыки создания Windows-приложения.

## *Порядок выполнения работы*

1 Разработать приложение с GUI-интерфейсом.

### **Задание 1**

Разработать программу, которая производит арифметические действия с двумя введенными числами. По нажатию кнопки «Очистить» поля для ввода и результат удаляются.

Создайте новый проект Windows Forms.

Задайте заголовок окна «Лабораторная работа 1».

Разместите на форме три компонента **Label**, три компонента **TextBox** и три компонента **Button**. Поменяйте для них свойства **Text** и положение так, чтобы получился внешний вид формы, аналогичный примеру (рисунок 1.1).

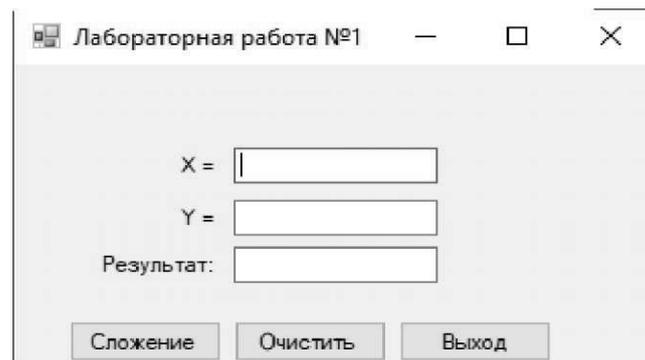


Рисунок 1.1 – Внешний вид формы

В зависимости от варианта, выданного преподавателем, программа будет выполнять сложение/вычитание/умножение/деление/возведение в степень. Соответствующее название дайте первой кнопке.

С кнопкой **Button1** свяжите обработчик события **Click**. В теле обработчика события объявите три переменные. Две переменные инициализируются числами, введенными в первые два текстовых поля, а в третьей хранится результат арифметического действия и выводится в третье текстовое поле.

Пример преобразования текста из **TextBox1** в вещественное число:

```
x = double.Parse(TextBox1.Text);
```

Пример вывода значения переменной в **TextBox1**:

```
TextBox1.Text = x.ToString();
```

По нажатии кнопки «Очистить» обработчик события должен всем трем текстовым полям присваивать пустые строки.

По нажатии кнопки «выход» обработчик события должен закрывать форму. Для этого существует функция **Close()**.

Протестируйте приложение.

2 Разработать приложение с GUI-интерфейсом.

### Задание 2

Написать программу для расчета по двум формулам согласно своему варианту (таблица 1.1). Создать интерфейс для ввода переменных и вывода результатов.

Таблица 1.1 – Варианты к заданию 2

Вариант	Формула 1	Формула 2
1	$z_1 = \frac{\sin 4\alpha}{1 + \cos 4\alpha} \cdot \frac{\cos 2\alpha}{1 + \cos 2\alpha}$	$z_2 = \operatorname{ctg}\left(\frac{3}{2}\pi - \alpha\right)$
2	$z_1 = \frac{1 - 2\sin^2 \alpha}{1 + \sin 2\alpha}$	$z_2 = \frac{1 - \operatorname{tg}\alpha}{1 + \operatorname{tg}\alpha}$
3	$z_1 = \frac{\sin\left(\frac{\pi}{2} + 3\alpha\right)}{1 - \sin(3\alpha - \pi)}$	$z_2 = \operatorname{ctg}\left(\frac{5}{4}\pi + \frac{3}{2}\alpha\right)$
4	$z_1 = (\cos \alpha - \cos \beta)^2 - (\sin \alpha - \sin \beta)^2$	$z_2 = -4 \sin^2 \frac{\alpha - \beta}{2} \cdot \cos(\alpha + \beta)$
5	$z_1 = \cos^4 x + \sin^2 y + \frac{1}{4} \sin^2 2x - 1$	$z_2 = \sin(y + x) \cdot \sin(y - x)$
6	$z_1 = \cos^2\left(\frac{3}{8}\pi - \frac{\alpha}{4}\right) - \cos^2\left(\frac{11}{8}\pi + \frac{\alpha}{4}\right)$	$z_2 = \frac{\sqrt{2}}{2} \sin \frac{\alpha}{2}$
7	$z_1 = \cos \alpha + \cos 2\alpha + \cos 6\alpha + \cos 7\alpha$	$z_2 = 4 \cos \frac{\alpha}{2} \cdot \cos \frac{5}{2}\alpha \cdot \cos 4\alpha$
8	$z_1 = 1 - \frac{1}{4} \sin^2 2\alpha + \cos 2\alpha$	$z_2 = \cos^2 \alpha + \cos^4 \alpha$
9	$z_1 = \frac{\sin 2\alpha + \sin 5\alpha - \sin 3\alpha}{\cos \alpha - \cos 3\alpha + \cos 5\alpha}$	$z_2 = \operatorname{tg} 3\alpha$
10	$z_1 = \frac{\sin 2\alpha + \sin 5\alpha - \sin 3\alpha}{\cos \alpha + 1 - 2 \sin^2 2\alpha}$	$z_2 = 2 \sin \alpha$

3 В качестве отчета представить работающее программное обеспечение.

4 Защитить лабораторную работу.

### Контрольные вопросы

- 1 Как создать приложение Windows Forms?
- 2 Что такое элементы управления?
- 3 Перечислите способы добавления элемента управления на форму.
- 4 В чём заключается различие между значимыми и ссылочными типами данных?
- 5 Допустимо ли использование величин разных типов в арифметическом выражении? Для чего нужно приведение типов и как оно реализуется?

## 2 Лабораторная работа № 2. Использование компонентов и настройка их свойств. Обработка событий

**Цель работы:** изучить методы построения форм Windows и добавления элементов управления; получить навыки по настройке форм; изучить способы использования элементов управления и получить навыки по обработке событий.

### *Порядок выполнения работы*

1 Написать программу с визуальным интерфейсом согласно варианту. Окно программы должно иметь название и пиктограмму.

2 В качестве отчета представить работающее программное обеспечение.

3 Защитить лабораторную работу.

### *Варианты заданий*

1 Добавьте на экран кнопки (красный, синий, зеленый, желтый, белый), по нажатию на которые будет меняться фоновый цвет экрана.

2 Программа загадывает число от 1 до 3, а пользователь пытается его угадать, нажимая на нужную кнопку с цифрой.

3 Добавьте на экран три кнопки и текстовое поле. По нажатию на каждую из кнопок текст, написанный на кнопке, попадает в текстовое поле. Добавьте кнопку для очистки текстового поля.

4 Добавьте на экран текстовое поле и две кнопки. Пусть в текстовом поле отображается количество нажатий на первую кнопку, а вторая сбрасывает подсчет.

5 Добавьте текстовое поле и две кнопки «Next» и «Back», по нажатию на которые текстовое поле циклично выводит сообщения.

6 Добавьте на экран две кнопки. Нажатие на первую кнопку меняет цвет второй кнопки на случайный, нажатие на вторую, делает то же с первой кнопкой.

7 Добавьте на экран три кнопки. Нажатие на одну из них меняет тексты на двух других кнопках между собой.

8 Добавьте на экран кнопки (красный, синий, зеленый) и текстовое поле, по нажатию на кнопки будет меняться цвет текста.

9 Программа загадывает цвет, а пользователь пытается его угадать, нажимая на кнопку нужного цвета. Достаточно четырех цветов.

10 Сделайте «клавиатуру». Добавьте текстовое поле и несколько кнопок с буквами и пробелом. В текстовом поле будет отображаться набранный текст.

### *Контрольные вопросы*

1 Какие элементы управления Вы знаете?

2 Какие свойства для настройки форм наиболее часто используются?

3 Как добавить обработчик события?

4 Как задается название окна?

### 3 Лабораторная работа № 3. Программирование разветвляющихся алгоритмов

**Цель работы:** изучить интегрированную среду, освоить простейшие структуры программы на языке C#; закрепить навыки в организации ввода-вывода с использованием визуального интерфейса; повторить базовые типы данных и математические функции языка C#; получить навыки в программировании разветвляющихся алгоритмов.

#### *Порядок выполнения работы*

1 Написать программу для вычисления значения функции с GUI-интерфейсом согласно своему варианту с использованием оператора выбора **if**. Протестировать программу при различных значениях аргументов.

2 Написать программу с GUI-интерфейсом согласно своему варианту на использование оператора выбора **switch**. Протестировать программу при различных исходных данных.

3 В качестве отчета представить работающее программное обеспечение.

4 Защитить лабораторную работу.

#### *Варианты заданий*

1 Задачи на использование оператора выбора **if**.

$$1 \quad y = \begin{cases} \sin x, & x \leq 0; \\ \operatorname{arctg} x, & 0 < x \leq \pi/4; \\ \log_2 x, & \pi/4 < x \leq 32; \\ 1/x, & \text{otherwise.} \end{cases}$$

$$2 \quad z = \begin{cases} \ln|x|, & x < -\pi; \\ \sin x + \cos 2x, & -\pi \leq x < \pi; \\ x^3 + 1, & \pi \leq x < 10; \\ \frac{x+1}{x^2+8}, & 10 \leq x < 100; \\ \ln x, & \text{otherwise.} \end{cases}$$

$$3 \quad z = \begin{cases} \operatorname{arctg} \frac{x}{y}, & y \neq 0 \quad \& \quad |x| > |y|; \\ \operatorname{arcsin} \frac{x}{y}, & y \neq 0 \quad \& \quad |x| \leq |y|; \\ 0, & \text{otherwise.} \end{cases}$$

$$4 \quad z = \begin{cases} \frac{1}{x} + \frac{1}{y}, & x < -10 \quad \& \quad y < -5; \\ \frac{x-y}{x+y}, & -10 \leq x < 0 \quad \& \quad -5 \leq y < 0; \\ \frac{\sin x}{\cos y}, & 0 \leq x < 2\pi \quad \& \quad 0 \leq y < \frac{\pi}{2}; \\ \ln(x^2 + y^2), & \text{otherwise.} \end{cases}$$

$$5 \quad z = \begin{cases} \sin \frac{x+y}{y}, & y \neq 0 \quad \& \quad |x| > |y|; \\ \arccos \frac{x}{y}, & y \neq 0 \quad \& \quad |x| \leq |y|; \\ \pi, & \text{otherwise.} \end{cases}$$

$$6 \quad y = \begin{cases} 0, & x \leq 0; \\ \frac{1}{x}, & 0 < x \leq 1; \\ x^2, & 1 < x \leq 4; \\ 14 + \log_2 x, & \text{otherwise.} \end{cases}$$

$$7 \quad z = \begin{cases} \frac{2}{x} - \frac{4}{y}, & x < -20 \quad \& \quad y < -10; \\ \frac{x-y-2}{x+y}, & -20 \leq x < 0 \quad \& \quad -10 \leq y < 0; \\ \frac{\sin x + \cos x}{\cos y}, & 1 < x \leq 4x < -20 \quad \& \quad y < -10; \\ \log_4(x^2 + y^2), & \text{otherwise.} \end{cases}$$

$$8 \quad y = \begin{cases} \cos x, & x \leq 0; \\ \arcsin x, & 0 < x \leq \pi/2; \\ \log_4 x, & \pi/2 < x \leq 64; \\ 1/x^2, & \text{otherwise.} \end{cases}$$

$$9 \quad y = \begin{cases} 0, & x < 0; \\ 1/(x+1), & 0 \leq x < 1; \\ 8+3x^3, & 1 \leq x < 4; \\ 62 + \log_8 x, & \text{otherwise.} \end{cases}$$

$$10 \quad z = \begin{cases} (x-y)(x+y), & x < 0 \quad \& \quad y \leq 0; \\ \operatorname{arctg} \frac{x}{y}, & 0 \leq x < 10 \quad \& \quad 0 < y \leq 8; \\ 0, & \text{otherwise.} \end{cases}$$

## 2 Задачи на использование оператора выбора **switch**.

1 По заданному виду арифметической операции (сложение, вычитание, умножение и деление) и двум операндам вывести на экран результат применения данной операции к операндам.

2 Написать алгоритм, позволяющий получить словесное наименование школьных оценок.

3 Написать алгоритм, который по номеру дня недели – целому числу от 1 до 7 будет выдавать в качестве результата количество пар в Вашей группе в соответствующий день.

4 Написать алгоритм нахождения числа дней в месяце.

5 По номеру дня недели вывести его название.

6 В зависимости от того введена ли открытая скобка или закрытая, напечатать «открытая круглая скобка» или «закрытая фигурная скобка» (учитывать круглые, квадратные, фигурные скобки).

7 В зависимости от введённого символа L, S, V программа должна вычислять длину окружности, площадь круга, объём цилиндра.

8 Написать программу, которая по введённому числу из промежутка 1...12 определяет пору года.

9 Написать программу, которая по введённому числу из промежутка 0...24 определяет время суток.

10 Написать программу преобразования цифр в слова.

### ***Контрольные вопросы***

- 1 Опишите синтаксис и работу полной, сокращённой и вложенной форм оператора **if**.
- 2 Нарисуйте блок-схемы различных форм оператора **if**.
- 3 С помощью каких операций формируется условие-выражение в операторе **if**?
- 4 Для чего предназначены логические выражения? В чём заключается различие между логическими и арифметическими выражениями?
- 5 К какому типу данных принадлежит результат работы операций сравнения и логических операций?
- 6 Объясните назначение, синтаксис, работу оператора **switch**.

## **4 Лабораторная работа № 4. Обработка одномерных массивов. Программирование циклических алгоритмов**

**Цель работы:** изучить интегрированную среду; получить практические навыки в работе с циклическими структурами языка C# при работе с одномерными массивами.

### ***Порядок выполнения работы***

- 1 Написать программу для решения задачи с GUI-интерфейсом согласно своему варианту, с использованием циклического оператора с предусловием. Вывести на печать все значения на каждом шаге итераций. Протестировать программу при различных значениях аргументов.
- 2 Написать программу с GUI-интерфейсом согласно своему варианту на работу с одномерными массивами. Массив задавать с помощью генератора случайных чисел. Для задания количества элементов в массиве использовать список множественного выбора. Протестировать программу при различных исходных данных.
- 3 В качестве отчета представить работающее программное обеспечение.
- 4 Защитить лабораторную работу.

### ***Варианты заданий***

- 1 Задачи на использование оператора цикла **do...while**.

1 Вычислить приближенное значение бесконечной суммы  $1 + \frac{1}{2} + \frac{1}{4} + \dots + \frac{1}{2^n} \dots$  с точностью до  $\varepsilon = 0,0001$ .

2 Вычислить приближенное значение бесконечной суммы  $1 - \frac{1}{2} + \frac{1}{4} - \frac{1}{8} \dots \pm \frac{1}{2^n} \mp \dots$  с точностью до  $\varepsilon = 0,05$ .

3 Вычислить приближенное значение бесконечной суммы  $\frac{1}{1 \cdot 3} + \frac{1}{3 \cdot 5} + \dots + \frac{1}{(2n-1)(2n+1)} + \dots$  с точностью до  $\varepsilon = 0,000005$ .

4 Вычислить приближенное значение бесконечной суммы  $\frac{1}{1 \cdot 3} + \frac{1}{7 \cdot 9} + \dots + \frac{1}{(4n-1)(4n+1)} + \dots$  с точностью до  $\varepsilon = 0,0001$ .

5 Вычислить приближенное значение бесконечной суммы  $\frac{1}{1^2} + \frac{1}{3^2} + \dots + \frac{1}{(2n+1)^2} + \dots$  с точностью до  $\varepsilon = 0,001$ .

6 Вычислить приближенное значение бесконечной суммы  $1 + \frac{1}{2^4} + \frac{1}{3^4} + \dots + \frac{1}{n^4} + \dots$  с точностью до  $\varepsilon = 0,0005$ .

7 Вычислить приближенное значение бесконечной суммы  $1 - \frac{1}{2^4} + \frac{1}{3^4} - \dots \pm \frac{1}{n^4} \mp \dots$  с точностью до  $\varepsilon = 0,0005$ .

8 Вычислить приближенное значение бесконечной суммы  $1 + \frac{1}{3^3} + \dots + \frac{1}{(2n+1)^3} + \dots$  с точностью до  $\varepsilon = 0,0001$ .

9 Вычислить приближенное значение бесконечной суммы  $\frac{1}{1 \cdot 4} + \frac{1}{4 \cdot 7} + \dots + \frac{1}{(3n-2)(3n+1)} + \dots$  с точностью до  $\varepsilon = 0,005$ .

10 Вычислить приближенное значение бесконечной суммы  $\frac{1}{1^2} + \frac{1}{3^2} + \dots \pm \frac{1}{(2n+1)^2} \mp \dots$  с точностью до  $\varepsilon = 0,005$ .

## 2 Задачи на работу с одномерными массивами.

1 В одномерном массиве, состоящем из  $n$  вещественных элементов, вычислить:

а) сумму отрицательных элементов массива;

б) произведение элементов массива, расположенных между максимальным и минимальным элементами.

Упорядочить элементы массива по возрастанию.

2 В одномерном массиве, состоящем из  $n$  вещественных элементов, вычислить:

а) сумму положительных элементов массива;

б) произведение элементов массива, расположенных между максимальным по модулю и минимальным по модулю элементами.

Упорядочить элементы массива по убыванию.

3 В одномерном массиве, состоящем из  $n$  целых элементов, вычислить:

а) произведение элементов массива с четными номерами;

б) сумму элементов массива, расположенных между первым и последним нулевыми элементами.

Преобразовать массив таким образом, чтобы сначала располагались все положительные элементы, а потом – все отрицательные (элементы, равные 0, считать положительными).

4 В одномерном массиве, состоящем из  $n$  вещественных элементов, вычислить:

а) сумму элементов массива с нечетными номерами;

б) сумму элементов массива, расположенных между первым и последним отрицательными элементами.

Сжать массив, удалив из него все элементы, модуль которых не превышает 1. Освободившиеся в конце массива элементы заполнить нулями.

5 В одномерном массиве, состоящем из  $n$  вещественных элементов, вычислить:

а) максимальный элемент массива;

б) сумму элементов массива, расположенных до последнего положительного элемента.

Сжать массив, удалив из него все элементы, модуль которых находится в интервале  $[a, b]$ . Освободившиеся в конце массива элементы заполнить нулями.

6 В одномерном массиве, состоящем из  $n$  вещественных элементов, вычислить:

а) минимальный элемент массива;

б) сумму элементов массива, расположенных между первым и последним положительными элементами.

Преобразовать массив таким образом, чтобы сначала располагались все элементы, равные нулю, а потом – все остальные.

7 В одномерном массиве, состоящем из  $n$  целых элементов, вычислить:

а) номер максимального элемента массива;

б) произведение элементов массива, расположенных между первым и вторым нулевыми элементами.

Преобразовать массив таким образом, чтобы в первой его половине располагались элементы, стоявшие в нечетных позициях, а во второй половине – элементы, стоявшие в четных позициях.

8 В одномерном массиве, состоящем из  $n$  вещественных элементов, вычислить:

а) номер минимального элемента массива;

б) сумму элементов массива, расположенных между первым и вторым отрицательными элементами.

Преобразовать массив таким образом, чтобы сначала располагались все элементы, модуль которых не превышает 1, а потом – все остальные.

9 В одномерном массиве, состоящем из  $n$  вещественных элементов, вычислить:

а) максимальный по модулю элемент массива;

б) сумму элементов массива, расположенных между первым и вторым положительными элементами.

Преобразовать массив таким образом, чтобы элементы, равные нулю, располагались после всех остальных.

10 В одномерном массиве, состоящем из  $n$  целых элементов, вычислить:

а) минимальный по модулю элемент массива;

б) сумму модулей элементов массива, расположенных после первого элемента, равного нулю.

Изменить порядок следования элементов в массиве на обратный.

### ***Контрольные вопросы***

1 Когда рекомендуется использовать операторы цикла **while**, **do...while**, **for**?

2 Где и когда используется оператор **continue**?

3 Назовите назначение оператора **break**. Где его можно применять?

4 К какому типу данных относится массив?

5 Назовите способы объявления одномерного массива.

6 Назовите способы инициализации одномерного массива.

7 Можно ли изменить размер массива после его создания?

8 Допустимо ли динамическое определение размера массива?

9 Что произойдет с программой, если индекс массива выйдет за объявленные границы?

10 Где размещается (в стеке или управляемой куче) ссылка на массив?

11 Чему равно свойство **Length** для одномерного массива?

## **5 Лабораторная работа № 5. Обработка двумерных массивов**

**Цель работы:** изучить интегрированную среду; получить практические навыки при работе с двумерными массивами.

### ***Порядок выполнения работы***

1 Написать программу с GUI-интерфейсом согласно своему варианту на работу с двумерными массивами. Массив задавать с помощью генератора случайных чисел. Для задания количества элементов в массиве использовать список множественного выбора. Протестировать программу при различных исходных данных.

2 В качестве отчета представить работающее программное обеспечение.

3 Защитить лабораторную работу.

### ***Варианты заданий***

1 Дана целочисленная прямоугольная матрица. Определить количество строк, не содержащих ни одного нулевого элемента.

2 Дана целочисленная прямоугольная матрица. Определить количество столбцов, не содержащих ни одного нулевого элемента.

3 Дана целочисленная квадратная матрица. Определить произведение элементов в тех строках, которые не содержат отрицательных элементов.

4 Дана целочисленная квадратная матрица. Определить сумму элементов в тех столбцах, которые не содержат отрицательных элементов.

5 Дана целочисленная матрица размером  $5 \times 4$ . Определить количество строк, содержащих больше одного нулевого элемента.

6 Найти наибольший элемент матрицы  $5 \times 3$  и номер строки и столбца, в котором он находится.

7 Вычислить сумму элементов каждой строки матрицы  $4 \times 3$ , определить наименьшее значение этих сумм и номер соответствующей строки.

8 Найти наибольшие элементы каждой строки матрицы  $X(4 \times 5)$  и записать их в массив  $Y$ .

9 Найти наибольший элемент главной диагонали матрицы  $4 \times 4$  и вывести на экран все строку, в которой он находится.

10 В массиве  $M[6 \times 4]$  все числа различны. Найти минимальные элементы каждой строки и среди них выбрать максимальное.

### ***Контрольные вопросы***

1 К какому типу данных относится двумерный массив?

2 Как организовать ввод/вывод массива размером  $N \times M$  элементов?

3 Объясните основные правила организации вложенных циклов, перечислите способы выхода из внутреннего цикла.

4 Чему равно свойство **Length** для двумерного массива?

5 Какие методы сортировки массива Вам известны?

6 Чем отличается принцип сортировки по неубыванию (невозрастанию) от сортировки по возрастанию (убыванию)?

## **6 Лабораторная работа № 6. Работа со строками**

**Цель работы:** изучить интегрированную среду; получить практические навыки при работе со строчными функциями.

### ***Порядок выполнения работы***

1 Написать программу согласно варианту.

2 В качестве отчета представить работающее программное обеспечение.

3 Защитить лабораторную работу.

### ***Варианты заданий***

1 Дана строка символов. Подсчитать количество слов, которые начинаются и заканчиваются одной и той же буквой.

2 Дана строка символов. Группы символов, разделенные пробелами (одним или несколькими) и не содержащие пробелов внутри себя, будем называть словами. Подсчитать количество слов в данной строке длиной в пять символов.

3 Дана строка символов. Удалить из данной строки символов все вхождения подстроки, введенной пользователем.

4 Дана строка символов. Группы символов, разделенные пробелами (одним или несколькими) и не содержащие пробелов внутри себя, будем называть словами. Найти длину самого короткого слова.

5 Дана строка символов. Определить, сколько раз в ней встречается слово «и» (в любом регистре).

6 Дана строка символов. Удалить из нее все пробелы. После этого определить, является ли она палиндромом, т. е. одинаково пишется как с начала, так и с конца.

7 Найти в строке указанную подстроку и заменить ее на новую. Строку, ее подстроку для замены и новую подстроку вводит пользователь.

8 Вводится строка слов, разделенных пробелами. Найти самое длинное слово и вывести его на экран. Случай, когда самых длинных слов может быть несколько, не обрабатывать.

9 Дана ненормированная строка, у которой могут быть пробелы в начале, в конце и между словами более одного пробела. Привести ее к нормированному виду, т. е. удалить все пробелы в начале и конце, а между словами оставить только один пробел.

10 Пользователь вводит неопределенное количество слов. Программа должна соединить их в одно предложение и в конце поставить точку.

### ***Контрольные вопросы***

1 Чем являются строки в C#?

2 Какого типа строки в C#?

3 Какая информация может быть представлена с помощью строк? Каковы основные правила их описания?

4 Можно ли массив символов инициализировать строкой символов?

5 Если строка – это последовательность символов, может ли строка быть короче одного символа?

6 Строка состоит из двух слов, каждое из пяти букв. Сколько символов в строке?

7 Какие существуют операции над строками?

## **7 Лабораторная работа № 7. Разработка форм в Windows Forms**

**Цель работы:** изучить методы построения форм Windows; получить навыки по настройке форм, созданию непрямых и наследуемых (производных) форм.

### ***Порядок выполнения работы***

1 Создать простое приложение, отслеживающее события мыши, которые происходят у конкретного элемента управления и изменяют соответствующие

свойства формы, согласно своему варианту. Добавить кнопку на форму, по нажатию на которую форма становится треугольной, при повторном нажатии возвращается к прежнему состоянию.

2 В качестве отчета представить работающее программное обеспечение.

3 Защитить лабораторную работу.

### *Варианты заданий*

1 Событие **Click** элемента управления **Button** изменяет **Location** формы.

2 Событие мыши **DoubleClick** элемента управления **ImageBox** изменяет **FormBorderStyle** формы.

3 Событие **GotFocus** элемента управления **TextBox** изменяет **Opacity** формы.

4 Событие мыши **MouseUp** элемента управления **Label** изменяет **BackgroundImage** формы.

5 Событие **LostFocus** элемента управления **ListBox** изменяет **Size** формы.

6 Событие **CheckedChanged** элемента управления **RadioButton** изменяет **MaximumSize** формы.

7 Событие мыши **MouseDoubleClick** элемента управления **MonthCalendar** изменяет **FormBorderStyle** формы.

8 Событие мыши **SelectedIndexChanged** элемента управления **ListBox** изменяет **Location** формы.

9 Событие мыши **MouseEnter** элемента управления **CheckBox** изменяет **Opacity** формы.

10 Событие мыши **MouseLeave** элемента управления **ImageList** изменяет **Size** формы.

### *Контрольные вопросы*

1 Как создать прямоугольную форму?

2 Какие свойства формы влияют на ее положение на экране?

3 Какие свойства формы могут сделать ее недоступной для пользователя?

4 Что такое наследуемые формы и как их создавать?

## **8 Лабораторная работа № 8. Работа с элементами управления. Панели и контейнеры**

**Цель работы:** закрепить практические навыки по работе с элементами управления; получить практические навыки использования контейнеров и методов компоновки с помощью панелей.

### *Порядок выполнения работы*

1 Создать приложение, в котором используются свойства кнопок. На форме определить несколько кнопок; при щелчке на одной кнопке какие-то кнопки становятся невидимыми, другие недоступны, меняется цвет, размер и т. д. (минимум пять свойств).

2 Создать приложение с использованием панелей, имеющее внешний вид, соответствующий приведенному на рисунках по вариантам (рисунки 8.1–8.5)

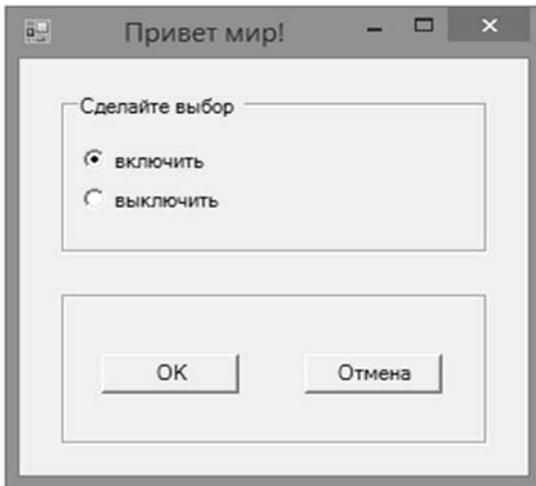


Рисунок 8.1 – Внешний вид формы для вариантов 1 и 6

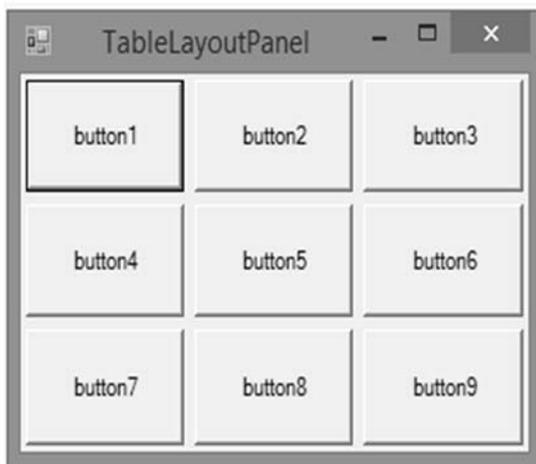


Рисунок 8.2 – Внешний вид формы для вариантов 2 и 7

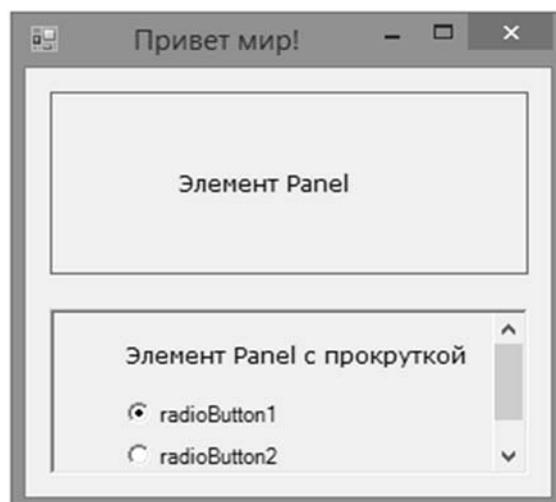


Рисунок 8.3 – Внешний вид формы для вариантов 3 и 8

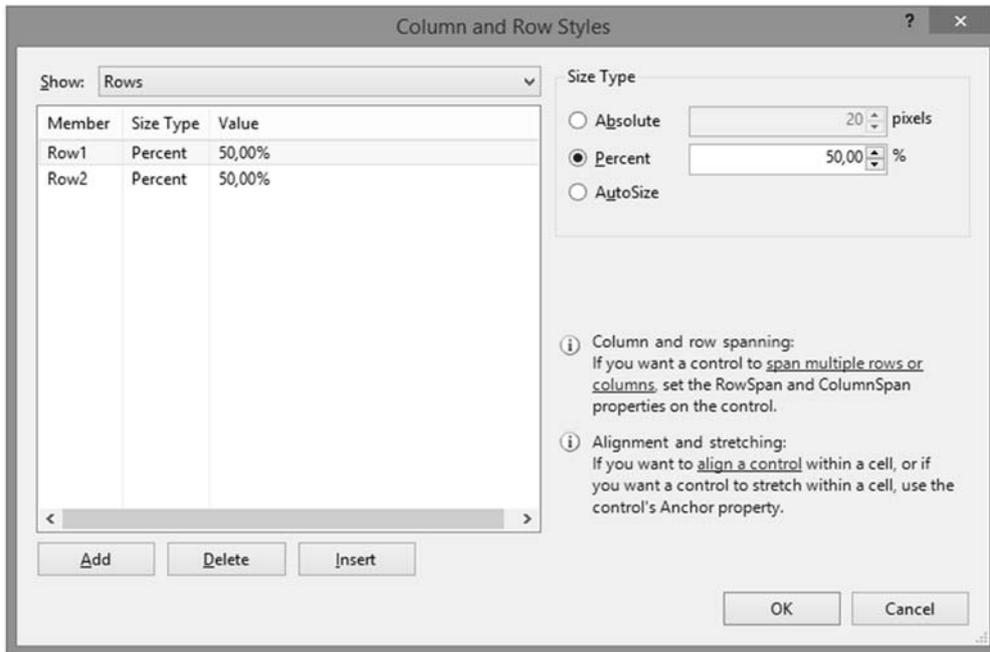


Рисунок 8.4 – Внешний вид формы для вариантов 4 и 9

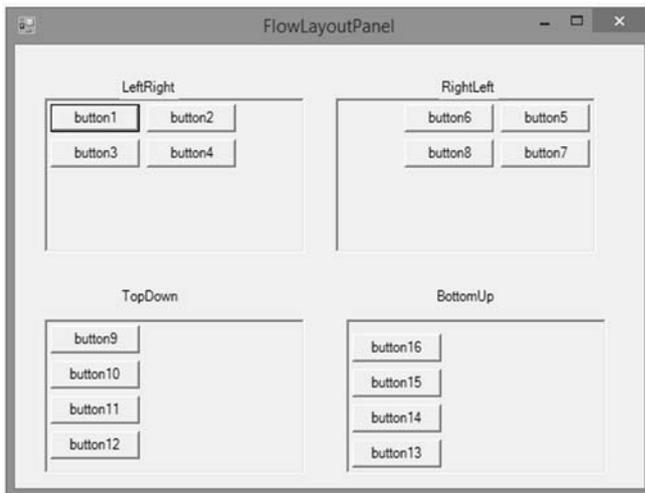


Рисунок 8.5 – Внешний вид формы для вариантов 5 и 10

- 3 В качестве отчета представить работающее программное обеспечение.
- 4 Защитить лабораторную работу.

### ***Контрольные вопросы***

- 1 Какие инструменты для компоновки элементов управления существуют в Windows Forms?
- 2 С помощью каких свойств можно управлять положением дочерних элементов внутри панели?
- 3 Какие особенности в табличной компоновке элементов управления с помощью панели **TableLayoutPanel**?
- 4 Какая панель требует жесткого позиционирования своих дочерних элементов?
- 5 Как **FlowLayoutPanel** управляет положением своих дочерних элементов?

## 9 Лабораторная работа № 9. Создание меню и вкладок. Многооконный интерфейс

**Цель работы:** изучить способы создания меню и вкладок; закрепить практические навыки по их настройке; получить практические навыки создания нескольких форм, передачи параметров и перехода между ними.

### *Порядок выполнения работы*

- 1 Написать программу согласно варианту.
- 2 В качестве отчета представить работающее программное обеспечение.
- 3 Защитить лабораторную работу.

### *Варианты заданий*

- 1 Создать меню с командами **Input**, **Calc** и **Exit**.

При выборе команды **Input** открывается диалоговое окно, содержащее:

- три поля типа **TextBox** для ввода длин трех сторон треугольника;
- группу из двух флажков (Периметр и Площадь) типа **CheckBox**;
- кнопку типа **Button**.

Обеспечить возможность:

- ввода длин трех сторон треугольника;
- выбора режима с помощью флажков: подсчет периметра и/или площади треугольника.

При выборе команды **Calc** открывается диалоговое окно с результатами.

При выборе команды **Exit** приложение завершается.

- 2 Создать меню с командами **Size**, **Color**, **Paint**, **Quit**.

Команда **Paint** недоступна. При выборе команды **Quit** приложение завершается. При выборе команды **Size** открывается диалоговое окно, содержащее:

- два поля типа **TextBox** для ввода длин сторон прямоугольника;
- группу из трех флажков (**Red**, **Green**, **Blue**) типа **CheckBox**;
- кнопку типа **Button**.

Обеспечить возможность:

- ввода длин сторон прямоугольника в пикселях в поля ввода;
- выбора его цвета с помощью флажков.

После задания параметров команда **Paint** становится доступной.

При выборе команды **Paint** в главном окне приложения выводится прямоугольник заданного размера и сочетания цветов или выдается сообщение, если введенные размеры превышают размер окна.

- 3 Создать меню с командами **Input**, **Work**, **Exit**.

При выборе команды **Exit** приложение завершает работу. При выборе команды **Input** открывается диалоговое окно, содержащее:

- три поля ввода типа **TextBox** с метками **Radius**, **Height**, **Density**;
- группу из двух флажков (**Volume**, **Mass**) типа **Checkbox**;
- кнопку типа **Button**.

Обеспечить возможность:

- ввода радиуса, высоты и плотности конуса;
- выбора режима с помощью флажков: подсчет объема и/или массы конуса.

При выборе команды **Work** открывается окно сообщений с результатами.

4 Создать меню с командами **Input, Calc, Draw, Exit**.

При выборе команды **Exit** приложение завершает работу. При выборе команды **Input** открывается диалоговое окно, содержащее:

- поле ввода типа **TextBox** с меткой **Radius**;
- группу из двух флажков (**Square, Length**) типа **CheckBox**;
- кнопку типа **Button**.

Обеспечить возможность:

- ввода радиуса окружности;
- выбора режима с помощью флажков: подсчет площади круга (**Square**) и/или длины окружности (**Length**).

При выборе команды **Calc** открывается окно сообщений с результатами.

При выборе команды **Draw** в центре главного окна выводится круг введенного радиуса или выдается сообщение, что рисование невозможно (если диаметр превышает размеры рабочей области).

5 Создать меню с командами **Input, Calc, About**.

При выборе команды **About** открывается окно с информацией о разработчике. При выборе команды **Input** открывается диалоговое окно, содержащее:

- три поля ввода типа **TextBox** с метками **Number 1, Number 2, Number 3**;

- группу из двух флажков (**Summ, Least multiple**) типа **CheckBox**;
- кнопку типа **Button**.

Обеспечить возможность:

- ввода трех чисел;
- выбора режима вычислений с помощью флажков: подсчет суммы трех чисел (**Summ**) и/или наименьшего общего кратного двух первых чисел (**Least multiple**).

При выборе команды **Calc** открывается диалоговое окно с результатами.

6 Создать меню с командами **Input, Calc, Quit**.

Команда **Calc** недоступна. При выборе команды **Quit** приложение завершается. При выборе команды **Input** открывается диалоговое окно, содержащее:

- два поля ввода типа **TextBox** с метками **Number 1, Number 2**;
- группу из трех флажков (**Summa, Max divisor, Multiply**) типа **CheckBox**;
- кнопку типа **Button**.

Обеспечить возможность:

- ввода двух чисел;
- выбора режима вычислений с помощью флажков (суммы, наибольшего общего делителя и произведения двух чисел).

При выборе команды **Calc** открывается окно сообщений с результатами.

7 Создать меню с командами **Begin, Help, About**.

При выборе команды **About** открывается окно с информацией о разработчике. При выборе команды **Begin** открывается диалоговое окно, содержащее:

- поле ввода типа **TextBox** с меткой **input**;
- метку типа **Label** для вывода результата;
- группу из трех переключателей (2, 8, 16) типа **RadioButton**;
- две кнопки типа **Button** – **Do** и **OK**.

Обеспечить возможность:

- ввода числа в десятичной системе в поле **input**;

– выбора режима преобразования с помощью переключателей: перевод в двоичную, восьмеричную или шестнадцатеричную систему счисления.

При щелчке на кнопке **Do** должен появляться результат перевода.

8 Создать меню с командами **Input color, Change, Exit, Help**.

При выборе команды **Exit** приложение завершает работу. При выборе команды **Input color** открывается диалоговое окно, содержащее:

- три поля ввода типа **TextBox** с метками **Red, Green, Blue**;
- группу из двух флажков (**Left, Right**) типа **CheckBox**;
- кнопку типа **Button**.

Обеспечить возможность ввода **RGB**-составляющих цвета. При выборе команды **Change** цвет главного окна изменяется на заданный (левая, правая или обе половины окна в зависимости от установки флажков).

9 Создать меню с командами **Input size, Choose, Change, Exit**.

При выборе команды **Exit** приложение завершает работу. Команда **Change** недоступна. При выборе команды **Input size** открывается диалоговое окно, содержащее:

- два поля ввода типа **TextBox** с метками **Size x, Size y**;
- кнопку типа **Button**.

При выборе команды **Choose** открывается диалоговое окно, содержащее:

- группу из двух переключателей (**Increase, Decrease**) типа **RadioButton**;
- кнопку типа **Button**.

Обеспечить возможность ввода значений в поля **Size x** и **Size y**. Значения интерпретируются как количество пикселей, на которое надо изменить размеры главного окна (увеличить или уменьшить в зависимости от положения переключателей).

После ввода значений команда **Change** становится доступной. При выборе этой команды размеры главного окна увеличиваются или уменьшаются на введенное количество пикселей.

10 Создать меню с командами **Begin, Work, About**.

При выборе команды **About** открывается окно с информацией о разработчике. При выборе команды **Begin** открывается диалоговое окно, содержащее:

- поле ввода типа **TextBox** с меткой **Input word**;
- группу из двух переключателей (**Upper case, Lower case**) типа **RadioButton**;
- кнопку типа **Button**.

Обеспечить возможность:

- ввода слова
- выбора режима перевода в верхний или нижний регистр в зависимости от положения переключателей.

При выборе команды **Work** открывается диалоговое окно с результатом перевода.

### ***Контрольные вопросы***

- 1 С помощью каких элементов управления создается меню?
- 2 Перечислите наиболее важные свойства компонентов меню.
- 3 Как создаются вкладки?
- 4 Какие есть способы передачи данных между формами? Какие наиболее правильные с точки зрения объектно-ориентированного программирования?

## 10 Лабораторная работа № 10. Классы, свойства, индексаторы

**Цель работы:** получить практические навыки по разработке программ с использованием классов.

### *Порядок выполнения работы*

1 Создать класс согласно своему варианту. Создать соответствующий тип-перечисление (**enum**).

Класс должен содержать:

- 4–5 скрытых полей;
- одно из полей должно быть типа перечисление;
- на каждое поле должно быть свойство;
- хотя бы одно свойство должно быть доступно только для чтения;
- конструктор по умолчанию, два конструктора с параметрами;
- индексатор типа **int** (только с методом **get**) с одним параметром типа перечисление; значение индексатора равно 1, если значение поля содержащего перечисление совпадает со значением индекса, –1 – если не совпадает, и 0 – если такого значения нет вообще.

2 Протестировать все свойства и методы в классе **Main**.

3 В качестве отчета представить работающее программное обеспечение.

4 Защитить лабораторную работу.

### *Варианты заданий*

- 1 Телефон.
- 2 Человек.
- 3 Животное.
- 4 Транспортное средство.
- 5 Текст.
- 6 Жилое здание.
- 7 Человек.
- 8 Лекарственное средство.
- 9 Компьютер.
- 10 Товар.

### *Контрольные вопросы*

1 Какие базовые концепции положены в основу объектно-ориентированного программирования? В чем они заключаются?

2 Дайте определение терминам «класс» и «объект». Как соотносятся эти понятия между собой?

3 Что такое инкапсуляция? В чем она заключается и как используется при работе с классами?

4 Что такое конструктор класса и для чего он используется в программе? Что такое конструктор копирования и конструктор по умолчанию?

5 В чём отличия свойств от полей?

6 Объясните принцип работы индексатора.

## 11 Лабораторная работа № 11. Наследование. Перегрузка методов и операторов. Полиморфизм

**Цель работы:** получить навыки использования механизмов наследования и полиморфизма классов; закрепить навыки перегрузки методов и операторов.

### *Порядок выполнения работы*

1 Построить иерархию классов предметной области согласно варианту (таблица 11.1).

Таблица 11.1 – Классы предметной области для построения иерархии

Номер варианта	Класс родитель	Производный класс
1	Телефон	Смартфон, стационарный
2	Человек	Студент, преподаватель
3	Животное	Млекопитающее, птица
4	Транспортное средство	Грузовик, легковой автомобиль
5	Текст	Статья, реклама
6	Жилое здание	Коттедж, многоквартирный дом
7	Человек	Пациент, врач
8	Лекарственное средство	Таблетки, мазь
9	Компьютер	Ноутбук, десктоп
10	Товар	Продукт, домашняя техника

Для этого необходимо:

- в классах описать конструкторы с параметрами и конструкторы по умолчанию, свойства для установки и получения значений полей классов, методы для описания поведения объектов. Каждый из создаваемых классов должен иметь не менее трёх методов, свойств, конструкторов;

- для каждого созданного класса переопределить методы **Equals()** и **ToString()**. Метод **Equals** переопределить так, чтобы объекты считались равными, если равны значения полей объектов, а не ссылки объектов;

- для классов перегрузить операцию сложения;

- для каждого созданного класса переопределить один и тот же метод родительского класса;

- создать массив из объектов обоих производных классов и продемонстрировать полиморфизм в действии.

2 В качестве отчета представить работающее программное обеспечение.

3 Защитить лабораторную работу.

### ***Контрольные вопросы***

- 1 Каким образом реализуется перегрузка конструкторов?
- 2 Можно ли перегрузкой операторов отменить очередность выполнения операции?
- 3 Какие операции требуют, чтобы левый операнд был объектом класса?
- 4 В чем заключена основная задача наследования?
- 5 Как осуществляется доступ к элементам производных и базовых классов?
- 6 Что такое полиморфизм? В чем он заключается и как используется при работе с классами?

## **12 Лабораторная работа № 12. Исключительные ситуации. Обработка исключений**

**Цель работы:** получить навыки в разработке программ с использованием обработки исключительных ситуаций.

### ***Порядок выполнения работы***

1 Для выполнения лабораторной работы использовать классы, разработанные в лабораторной работе № 11.

Во всех заданиях реализуемые методы должны генерировать подходящие исключения. Обработку исключений нужно выполнять методом **Main()**, который должен демонстрировать обработку всех перехватываемых исключений.

Методы, реализуемые в заданиях, обязаны выполнять проверку передаваемых параметров и генерировать исключение в случае ошибочных ситуаций.

Все методы реализуются в пяти вариантах:

- 1) без спецификации исключений;
- 2) со спецификацией **throw()**;
- 3) с конкретной спецификацией с подходящим стандартным исключением (использовать методы класса **Exception**);
- 4) спецификация с собственным реализованным исключением. Собственное исключение должно быть реализовано как наследник от стандартного исключения.
- 5) использования вложения блоков **try**. Должно выводиться сообщение об отсутствии обработчика исключения, в случае непредвиденной ошибки.

В конце каждой последовательности операторов **try/catch** использовать блок **finally**.

2 В качестве отчета представить работающее программное обеспечение.

3 Защитить лабораторную работу.

### ***Контрольные вопросы***

- 1 Какие ключевые слова используются при работе с механизмом исключений?
- 2 Для чего применяется ключевое слово **catch**?

- 3 Для чего применяется ключевое слово **finally**?
- 4 Для чего применяется ключевое слово **throw**?
- 5 Как отследить все стандартные исключительные ситуации?

### 13 Лабораторная работа № 13. Делегаты и события

**Цель работы:** научиться разрабатывать программы с использованием делегатов и событий.

#### *Порядок выполнения работы*

1 В программе, разработанной в лабораторной работе № 12, требуется определить класс, содержащий типизированную коллекцию, который с помощью событий извещает об изменениях в коллекции.

Коллекция состоит из объектов ссылочных типов. Коллекция изменяется при удалении/добавлении элементов или при изменении одной из входящих в коллекцию ссылок, например, когда одной из ссылок присваивается новое значение. В этом случае в соответствующих методах или свойствах класса бросаются события.

При изменении данных объектов, ссылки на которые входят в коллекцию, значения самих ссылок не изменяются. Этот тип изменений не порождает событий.

Для событий, извещающих об изменениях в коллекции, определяется свой делегат. События регистрируются в специальных классах-слушателях.

Для этого определить класс **Journal**, который можно использовать для накопления информации об изменениях в коллекциях. Класс **Journal** хранит информацию в списке объектов. Каждый элемент списка содержит информацию об отдельном изменении, которое произошло в коллекции.

- 2 В качестве отчета представить работающее программное обеспечение.
- 3 Защитить лабораторную работу.

#### *Контрольные вопросы*

- 1 Назовите этапы создания и применения делегатов.
- 2 Как осуществляется вызов методов с помощью делегата?
- 3 В чём заключается механизм события?
- 4 Каков порядок создания пользовательского события?
- 5 Как используются методы класса в роли обработчика события?
- 6 Какой синтаксис должны иметь .NET-совместимые обработчики событий?

## 14 Лабораторная работа № 14. Интерфейсы

**Цель работы:** ознакомиться с основами объектного подхода в языке C#; приобрести практические навыки по работе с интерфейсами; закрепить навыки использования наиболее часто применяемых интерфейсов в языке C#.

### *Порядок выполнения работы*

- 1 В программе, разработанной в лабораторной работе № 13:
  - реализовать в классе-родителе интерфейс **System.Collections.IEnumerable** для перебора объектов по отдельному признаку;
  - определить вспомогательный класс, реализующий интерфейс **System.Collections.IEnumerator** для производного класса;
  - реализовать в классе-родителе интерфейс **IComparable** для выполнения сортировки по одному из признаков;
  - реализовать в классе-родителе интерфейс **IComparer** <«класс-родитель»> для выполнения сортировки по другому признаку;
  - реализовать во вспомогательном классе интерфейс **IComparer** <«класс-наследник»> для сортировки по признаку, принадлежащему классу-наследнику;
  - определить методы по поиску и по всем видам сортировок;
  - протестировать все методы в классе **Main**.
- 2 В качестве отчета представить работающее программное обеспечение.
- 3 Защитить лабораторную работу.

### *Контрольные вопросы*

- 1 Что такое интерфейс?
- 2 Какие элементы языка C# могут быть членами интерфейса?
- 3 Сколько интерфейсов может наследовать класс?
- 4 Где должны быть реализованы методы интерфейса?
- 5 Можно ли реализовать множественный интерфейс?
- 6 Как проявляется принцип полиморфизма при использовании интерфейсов?

## 15 Лабораторная работа № 15. Обобщения. Классы-коллекции

**Цель работы:** научиться разрабатывать обобщённые классы, коллекции, итераторы и применять их в программах.

### *Порядок выполнения работы*

- 1 На базе программы, разработанной в предыдущей лабораторной работе, определить вместо массива объектов новый класс **Collection**, содержащий следующее:
  - закрытое поле типа **List<T>** – список и **Dictionary<TKey, TValue>** – словарь. В качестве параметра типа <T> для **List<T>** использовать базовый тип в

созданной иерархии классов. В коллекции **Dictionary**<TKey, TValue> в качестве параметра типа <TKey> использовать наименование производного класса, а в качестве <TValue> – значение одного из полей производного класса;

- конструктор с параметрами и конструктор по умолчанию для инициализации полей;
- методы для просмотра, добавления, удаления элементов в коллекцию;
- перегруженную версию виртуального метода **ToString()** для формирования строки с информацией об элементах коллекции;
- метод, выполняющий сортировку коллекции по заданному условию;
- итератор для поиска элементов коллекции, удовлетворяющих заданному условию;
- метод поиска в коллекции-словаре по ключу и по значению.

В методе **Main()** создать экземпляр класса **Collection** и продемонстрировать всю функциональность этого класса.

2 В качестве отчета представить работающее программное обеспечение.

3 Защитить лабораторную работу.

### ***Контрольные вопросы***

1 Что понимают в языке C# под термином обобщение?

2 Как создать класс обобщенного типа? Сколько у него может быть параметров?

3 Можно ли определять обобщенные свойства, методы?

4 Что такое коллекции?

5 Опишите состав пространства имен **System.Collections** и дайте характеристику основных типов-коллекций.

6 Объясните назначение классов **List**<T> и **Dictionary**<TKey, TValue>. Назовите основные методы и свойства классов **List**<T> и **Dictionary**<TKey, TValue>.

7 Какие интерфейсы реализуются в классах **List**<T> и **Dictionary**<TKey, TValue>?

## **16 Лабораторная работа № 16. Работа со строками. Использование регулярных выражений**

**Цель работы:** получить практические навыки по программированию регулярных выражений в языке C#.

### ***Порядок выполнения работы***

1 В программе, разработанной в предыдущей лабораторной работе организовать валидацию вводимых данных при создании новых объектов с использованием регулярных выражений.

Организовать поиск данных по некоторому вводимому слову или сочетанию символов с использованием регулярных выражений.

- 2 В качестве отчета представить работающее программное обеспечение.
- 3 Защитить лабораторную работу.

### ***Контрольные вопросы***

- 1 Для чего предназначены регулярные выражения?
- 2 Перечислите основные действия, которые можно выполнять над строками с помощью регулярных выражений.
- 3 Из каких элементов состоит язык описания регулярных выражений?
- 4 Перечислите наиболее употребительные метасимволы.
- 5 Что такое мнимые метасимволы? Приведите примеры использования мнимых метасимволов.
- 6 Какую роль в регулярных выражениях выполняют повторители? Приведите примеры повторителей.
- 7 Перечислите основные методы регулярных выражений.

## **17 Лабораторная работа № 17. Использование LINQ для работы с данными**

**Цель работы:** ознакомиться с языком интегрированных запросов LINQ и его использованием для работы с данными.

### ***Порядок выполнения работы***

- 1 В программе, разработанной в лабораторной работе № 16, организовать поиск в коллекции объектов по различным полям с использованием языка интегрированных запросов.  
Организовать вывод результатов поиска на форме в виде текста.
- 2 В качестве отчета представить работающее программное обеспечение.
- 3 Защитить лабораторную работу.

### ***Контрольные вопросы***

- 1 Каковы структура и работа простого запроса LINQ?
- 2 Какой оператор используется для отбора данных, возвращаемых по запросу?
- 3 Как отсортировать результаты запроса LINQ?
- 4 Объясните назначение следующих операторов запроса LINQ: **group**, **into**, **let**, **join**.

## 18 Лабораторная работа № 18. Классы для работы с файлами. Сериализация

**Цель работы:** изучить особенности основных принципов и механизмов, используемых для сериализации и десериализации программных объектов; изучить особенности сохранения и восстановления состояния программных объектов различной сложности путем их сериализации и десериализации.

### *Порядок выполнения работы*

1 На основе программы, разработанной в предыдущей лабораторной работе, создать программную реализацию сохранения состояния объектов в файле и восстановления их из файла в бинарном и XML-форматах.

Для выбора имени файла используется класс **OpenFileDialog**, который открывает диалоговое окно с выбором файла. Ограничить выбор только файлами с расширением «.xml».

Реализовать тестирование процессов сериализации и десериализации с возможностью сравнения запоминаемых объектов и их восстановленных копий.

2 В качестве отчета представить работающее программное обеспечение.

3 Защитить лабораторную работу.

### *Контрольные вопросы*

- 1 Что такое сериализация/десериализация?
- 2 Как реализовать сериализацию и в чем ее значение?
- 3 Как узнать, можно ли сериализовать объект класса?
- 4 Сохраняется ли возможность сериализации при наследовании?
- 5 Сколько полей содержит интерфейс **Serializable**?

## 19 Лабораторная работа № 19. Отображение массивов и баз данных в таблицах

**Цель работы:** закрепить практические навыки по работе с массивами; закрепить навыки использования статических методов; получить практические навыки использования методов для обработки массивов.

### *Порядок выполнения работы*

1 В программе, созданной в предыдущих лабораторных работах осуществить вывод данных в таблице. Для этого:

- организовать вывод содержимого списка **List<T>** в **DataGridView**;
- добавить поля, дающие возможность для поиска, фильтрации и сортировки списка по различным критериям;
- результаты должны отображаться в той же таблице;
- предусмотреть возможность возврата к предыдущему шагу и сброса в начальное состояние.

2 В качестве отчета представить работающее программное обеспечение.

3 Защитить лабораторную работу.

### *Контрольные вопросы*

- 1 Как связать данные с элементом управления **DataGridView**?
- 2 Какие возможности предоставляет **DataGridView**?
- 3 Элемент управления **DataGridView** поддерживает привязку данных к открытым свойствам объектов, возвращаемых какими интерфейсами?
- 4 Можно ли использовать **DataGridView** в несвязанном режиме?
- 5 Какие еще существуют элементы управления для отображения данных в табличном виде и их отличия?

## **20 Лабораторная работа № 20. Построение графиков и диаграмм**

**Цель работы:** изучить возможности и закрепить практические навыки построения графических зависимостей.

### *Порядок выполнения работы*

- 1 Написать программу согласно варианту.
- 2 В качестве отчета представить работающее программное обеспечение.
- 3 Защитить лабораторную работу.

### *Варианты заданий*

1 Написать Windows-приложение, которое строит график функции и выделяет красным цветом линии на графике, значения которых выше некой заданной величины (числа-ограничителя).

Создать меню, с командами **Chart**, **Build**, **Clear**, **About**, **Quit**.

Команда **Quit** завершает работу приложения. При выборе команды **About** открывается окно с информацией о разработчике.

Команда **Chart** открывает диалоговое окно, содержащее:

- список для выбора цвета графика типа **TListBox**;
- поле типа **TextBox** для ввода числа-ограничителя;
- кнопку типа **Button**.

Обеспечить возможность выбора цвета. После щелчка на кнопке **ОК** в главном окне приложения строится график выбранной функции. Окно должно содержать заголовок графика, наименование и градацию осей. Изображение должно занимать все окно и масштабироваться при изменении размеров окна.

Команда **Clear** очищает окно.

2 Написать Windows-приложение, которое по заданным в файле исходным данным строит круг в прямоугольнике.

Создать меню с командами **Input data**, **Choose**, **Quit**.

При выборе команды **Input data** из файла читаются исходные данные (файл сформировать самостоятельно) – стороны прямоугольника.

По команде **Choose** открывается диалоговое окно, содержащее:

- списки для выбора цвета обеих фигур типа **TListBox**;
- кнопку типа **Button**.

Обеспечить возможность ввода цвета. Окно должно содержать заголовок. Изображение должно занимать все окно и масштабироваться при изменении размеров окна.

3 Написать Windows-приложение – графическую иллюстрацию графика квадратного уравнения:

$$y = ax^2 + c.$$

Создать меню с командами **Open, Root, Show, About, Exit**.

Команда **Exit** завершает работу приложения. Команда **About** открывает окно с информацией о разработчике. Для выбора файла исходных данных (команда **Open**) использовать объект **OpenFileDialog**. Исходные данные для уравнения – массивы значений коэффициентов  $a$  и  $c$  – сформировать самостоятельно.

При выборе команды **Root** выводится окно сообщений с корнем уравнения  $x$ . При выборе команды **Show** в главном окне приложения отображается график уравнения. Окно должно содержать заголовок. Изображение должно занимать все окно.

4 Написать Windows-приложение – графическую иллюстрацию нахождения точки пересечения двух графиков:

$$y = 6x, \quad y = 2x^2.$$

Создать меню с командами **Choose, Coordinates, Show, About, Exit**.

Команда **Exit** завершает работу приложения. Команда **About** открывает окно с информацией о разработчике.

При выборе команды **Choose** открывается диалоговое окно, содержащее:

- списки для выбора цветов графиков типа **TListBox**;
- кнопку типа **Button**.

При выборе команды **Coordinates** выводится окно сообщений с вычисленными координатами  $y, x$ . При выборе команды **Show** в главном окне приложения отображаются оба графика зависимости. Окно должно содержать заголовок. Изображение должно занимать все окно.

5 Написать Windows-приложение, которое выполняет строит квадрат.

Создать меню с командами **Choose, Show, About, Quit**.

Команда **About** открывает окно с информацией о разработчике.

Команда **Quit** завершает работу приложения.

При выборе команды **Choose** открывается диалоговое окно, содержащее:

- поле типа **TextBox** для ввода длины сторон квадрата;
- поле типа **TextBox** с меткой **Center** для ввода координат центра квадрата;
- кнопку типа **Button**.

При выборе команды **Show** в главном окне приложения отображается график уравнения. Окно должно содержать заголовок, наименование и градацию осей.

6 Написать Windows-приложение, которое строит две геометрические фигуры (круги).

Создать меню, с командами **Choose, Build, Clear, About, Quit**.

Команда **Quit** завершает работу приложения. При выборе команды **About** открывается окно с информацией о разработчике.

Команда **Choose** открывает диалоговое окно, содержащее:

- список для выбора цвета круга типа **TListBox**;
- два поля типа **TextBox** для ввода координат центра первого круга и второго круга;
- кнопку типа **Button**.

Обеспечить возможность выбора цвета. После щелчка на кнопке **OK** в главном окне приложения появляются два круга, центры которых соединены друг с другом синей прямой. Окно должно содержать заголовок. Изображение должно занимать все окно и масштабироваться при изменении размеров окна.

Команда **Clear** очищает окно.

7 Написать Windows-приложение, которое по заданным в файле исходным данным строит в круге прямоугольник.

Создать меню с командами **Input data, Choose, Quit**.

При выборе команды **Input data** из файла читаются исходные данные (файл сформировать самостоятельно) – стороны прямоугольника.

По команде **Choose** открывается диалоговое окно, содержащее:

- списки для выбора цвета обеих фигур типа **TListBox**;
- кнопку типа **Button**.

Обеспечить возможность ввода цвета. Окно должно содержать заголовок. Изображение должно занимать все окно и масштабироваться при изменении размеров окна.

8 Написать Windows-приложение, которое строит графики трех заданных функций.

Создать меню, с командами **Chart, Build, Clear, About, Quit**.

Команда **Quit** завершает работу приложения. При выборе команды **About** открывается окно с информацией о разработчике.

Команда **Chart** открывает диалоговое окно, содержащее:

- список для выбора цвета графика типа **TListBox**;
- список для выбора типа графика типа **TListBox**, содержащий три пункта:

$$y = 2x, y = 2x^2, y = 2x^3;$$

- кнопку типа **Button**.

Обеспечить возможность выбора цвета и вида графика. После щелчка на кнопке **OK** в главном окне приложения строится график выбранной функции.

Окно должно содержать заголовок графика, наименование и градацию осей. Изображение должно занимать все окно и масштабироваться при изменении размеров окна.

Команда **Clear** очищает окно.

9 Написать Windows-приложение, которое строит домик из двух фигур – прямоугольника и треугольника.

Создать меню, с командами **Choose, Build, Clear, About, Quit**.

Команда **Quit** завершает работу приложения. При выборе команды **About** открывается окно с информацией о разработчике.

Команда **Choose** открывает диалоговое окно, содержащее:

- список для выбора цвета фигур;
- два поля типа **TextBox** для ввода длины сторон квадрата;
- кнопку типа **Button**.

Обеспечить возможность выбора цвета. После щелчка на кнопке **OK** в главном окне приложения появляются треугольник и квадрат, образующие домик. Окно должно содержать заголовок. Изображение должно занимать все окно и масштабироваться при изменении размеров окна.

Команда **Clear** очищает окно.

10 Написать Windows-приложение, которое строит график функции и выделяет синим цветом линии на графике, значения которых ниже некой заданной величины (числа-ограничителя).

Создать меню с командами **Chart, Build, Clear, About, Quit**.

Команда **Quit** завершает работу приложения. При выборе команды **About** открывается окно с информацией о разработчике.

Команда **Chart** открывает диалоговое окно, содержащее:

- список для выбора цвета графика типа **TListBox**;
- поле типа **TextBox** для ввода числа-ограничителя;
- кнопку типа **Button**.

Обеспечить возможность выбора цвета. После щелчка на кнопке **OK** в главном окне приложения строится график выбранной функции. Окно должно содержать заголовок графика, наименование и градацию осей. Изображение должно занимать все окно и масштабироваться при изменении размеров окна.

Команда **Clear** очищает окно.

### ***Контрольные вопросы***

- 1 Можно ли в **ZedGraph** строить диаграммы?
- 2 Какой метод позволяет обновить данные об осях?
- 3 Как настраивать свойства осей?
- 4 Как можно изменить тип линии, например на пунктирные и штрихпунктирные?
- 5 Как выводить постоянно обновляемые данные?

## 21 Лабораторная работа № 21. Рисование в Windows Forms

**Цель работы:** освоить основы создания графических приложений в системе Windows с использованием технологий Windows Forms; закрепить практические навыки по работе с массивами; закрепить навыки использования статических методов; получить практические навыки использования методов для обработки массивов.

### *Порядок выполнения работы*

- 1 Создать согласно варианту приложение, в котором при нажатии клавиш (клавиши выбираются самостоятельно) происходят вызовы методов **RotateTransform**, **ScaleTransform**, **TranslateTransform**, а также различных свойств класса **Graphics**. Вызовы методов применить к графическим примитивам (**Image**, **Rectangle**, **Ellipse**, **Line** и пр.) и блокам текста. При каждом событии клавиатуры, изменять как положение графических объектов, так и их заливку.
- 2 В качестве отчета представить работающее программное обеспечение.
- 3 Защитить лабораторную работу.

### *Варианты заданий*

- 1 Image, Rectangle, Ellipse.
- 2 Rectangle, Ellipse, Polygon.
- 3 Ellipse, Image, Line.
- 4 Rectangle, Line, Text.
- 5 Polygon, Text, Image.
- 6 Rectangle, Line, Text.
- 7 Image, Rectangle, Text.
- 8 Text, Rectangle, Polygon.
- 9 Image, Polygon, Text.
- 10 Ellipse, Line, Text.

### *Контрольные вопросы*

- 1 Какой класс отвечает за создание графических изображений?
- 2 Какой набор классов называется интерфейсом управляемого класса для GDI+?
- 3 Каким образом происходит заливка фигур и настройка их внешнего вида?
- 4 Как работает трансформация для графических примитивов?

## 22 Лабораторная работа № 22. Анимация с помощью таймера

**Цель работы:** закрепить практические навыки по созданию графических приложений в системе Windows с использованием технологий Windows Forms; отработать навыки по созданию простейшей анимации на основе таймера.

### *Порядок выполнения работы*

1 Нарисовать на форме окружность с заливкой. Каждые  $t$  секунд цвет окружности должен изменяться; при этом название цвета отображается в нижней части формы. Использовать объект **Timer**.

Цвет выбирается случайным образом (использовать объект **Random()**).

2 Напишите приложение, согласно варианту, которое выполняет анимацию геометрической фигуры.

Создать меню с командами **Choose, Animate, Stop, Quit**.

Команда **Quit** завершает работу приложения.

При выборе команды **Choose** открывается диалоговое окно, содержащее:

- группу переключателей (**squared, triangle, ellipse**) типа **RadioButton** – выбор используемой фигуры;
- список для выбора цвета фигуры типа **TListBox**;
- список для выбора цвета фона типа **TListBox**;
- поле типа **TextBox** с меткой **Speed** для ввода скорости движения объекта.

По команде **Animate** объект начинает перемещаться с заданной скоростью, по команде **Stop** – прекращает движение.

3 В качестве отчета представить работающее программное обеспечение.

4 Защитить лабораторную работу.

### *Варианты заданий*

1 Написать Windows-приложение, которое выполняет анимацию геометрической фигуры по диагонали окна вверх слева направо.

2 Написать Windows-приложение, которое выполняет анимацию геометрической фигуры по горизонтали окна вправо.

3 Написать Windows-приложение, которое выполняет анимацию геометрической фигуры по вертикали окна вверх.

4 Написать Windows-приложение, которое выполняет анимацию геометрической фигуры по диагонали окна вниз слева направо.

5 Написать Windows-приложение, которое выполняет анимацию геометрической фигуры по горизонтали окна влево.

6 Написать Windows-приложение, которое выполняет анимацию геометрической фигуры по диагонали окна вниз справа налево.

7 Написать Windows-приложение, которое выполняет анимацию геометрической фигуры по вертикали окна вниз.

8 Написать Windows-приложение, которое выполняет анимацию геометрической фигуры по диагонали окна вверх справа налево.

9 Написать Windows-приложение, которое выполняет анимацию геометрической фигуры по диагонали окна вверх слева направо.

10 Написать Windows-приложение, которое выполняет анимацию геометрической фигуры по горизонтали окна вправо.

### ***Контрольные вопросы***

1 В чем заключается принцип анимации геометрических фигур?

2 Какие функции, позволяющие установить в приложении требуемое количество программных таймеров, предоставляет Windows прикладному программисту?

3 Чем отличаются мультимедийные таймеры от обычных?

4 Как задать однократный интервал времени?

5 Как можно измерять интервалы времени?

## **23 Лабораторная работа № 23. Диспетчеры компоновки**

**Цель работы:** получить практические навыки по созданию интерфейса для WPF-приложений; изучить способы компоновки и трансформации объектов.

### ***Порядок выполнения работы***

1 Разработать приложение WPF в соответствии с предложенным на рисунках 23.1–23.10 графическим интерфейсом и с аналогичным поведением при изменении размеров окна согласно варианту. При создании интерфейса стараться избегать явного задания размеров.

2 В качестве отчета представить работающее программное обеспечение.

3 Защитить лабораторную работу.

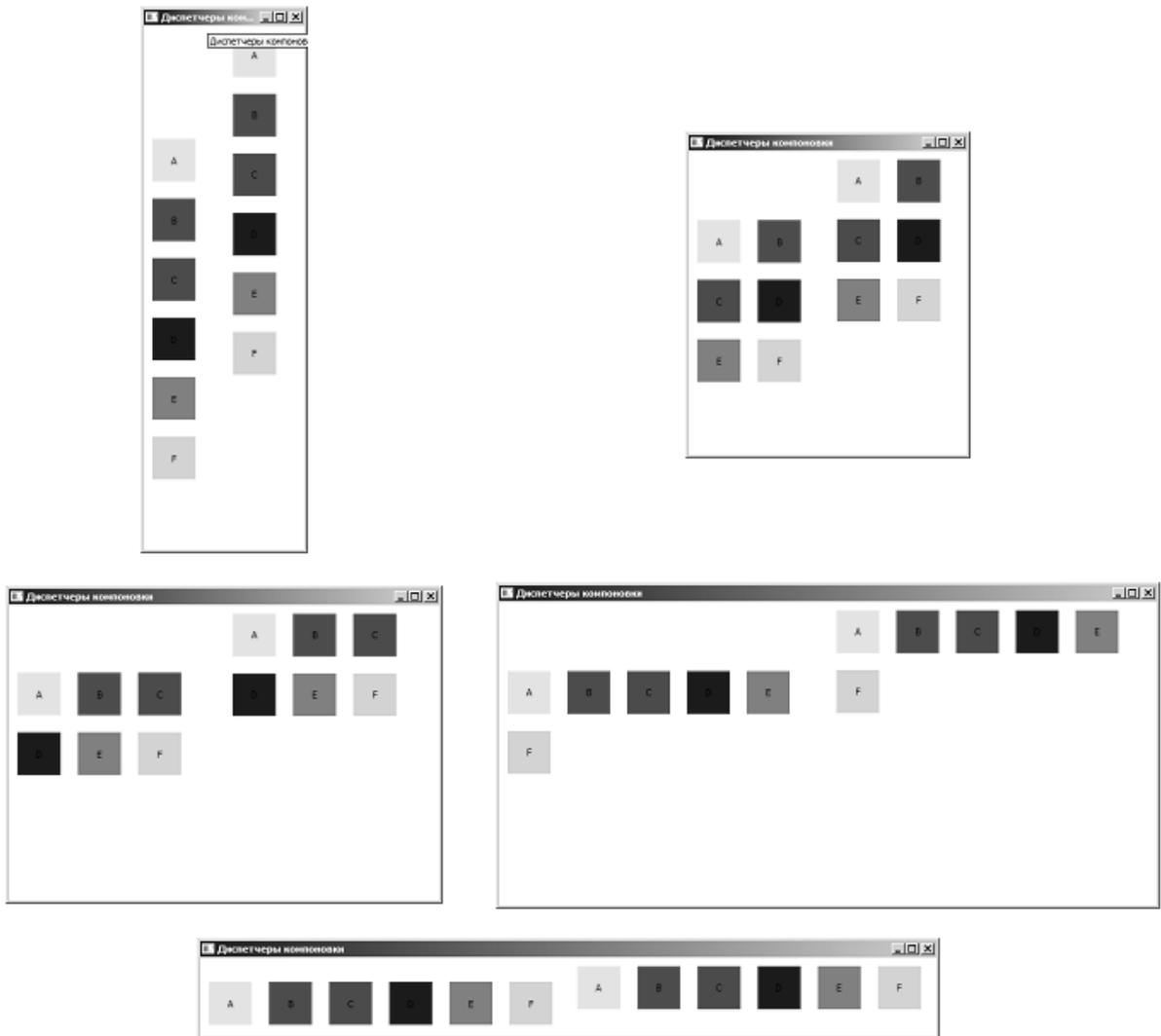


Рисунок 23.1 – Внешний вид приложения для варианта 1

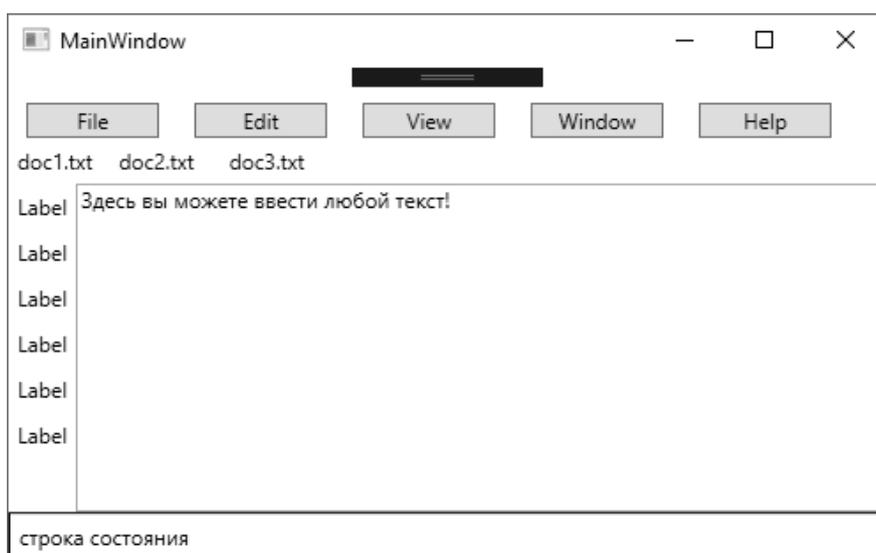


Рисунок 23.2 – Внешний вид приложения для варианта 2



Рисунок 23.3 – Внешний вид приложения для варианта 3

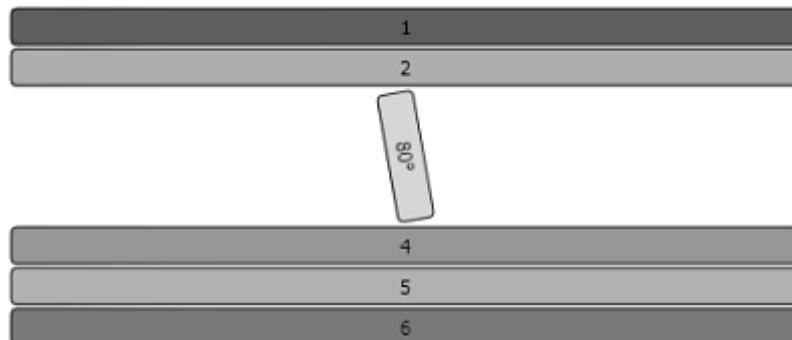


Рисунок 23.4 – Внешний вид приложения для варианта 4

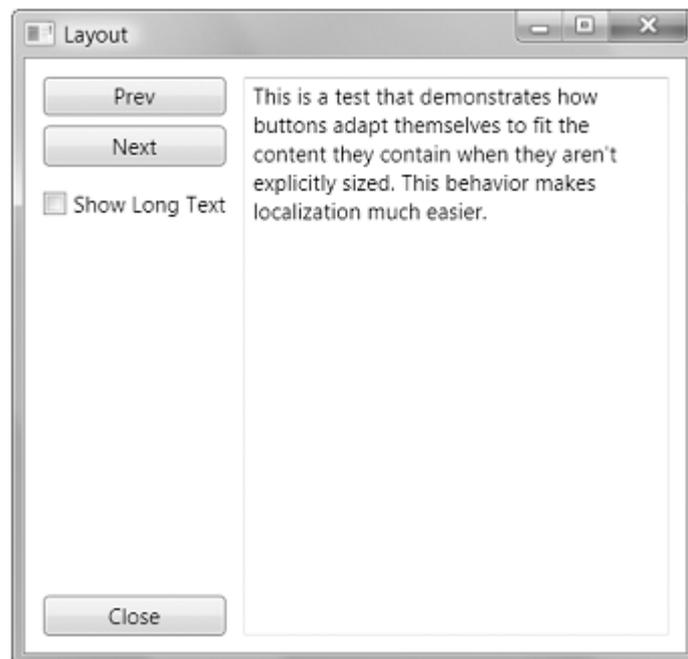


Рисунок 23.5 – Внешний вид приложения для варианта 5

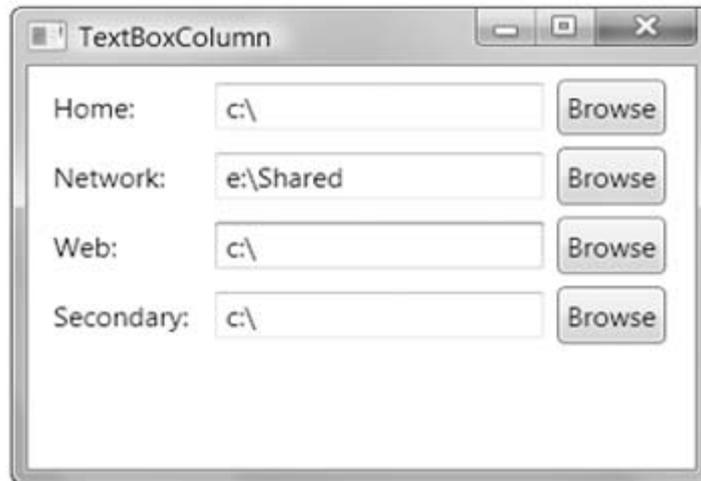


Рисунок 23.6 – Внешний вид приложения для варианта 6

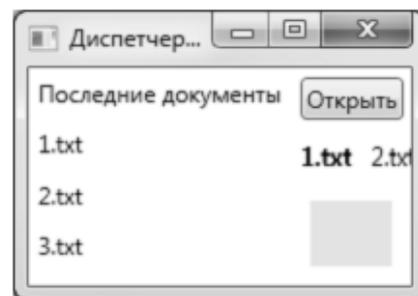
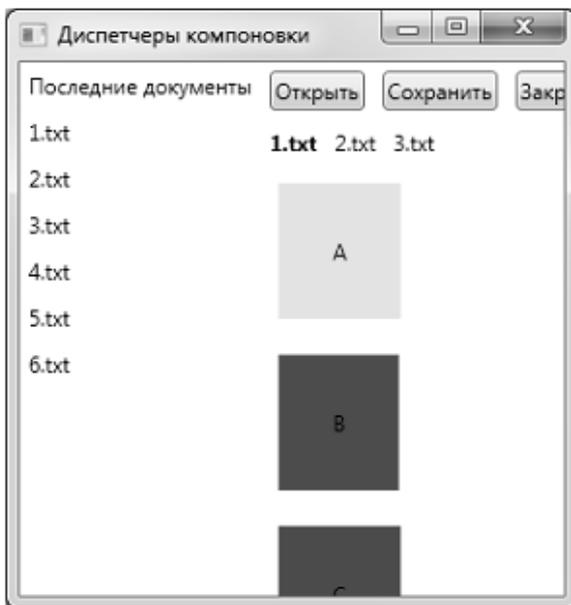
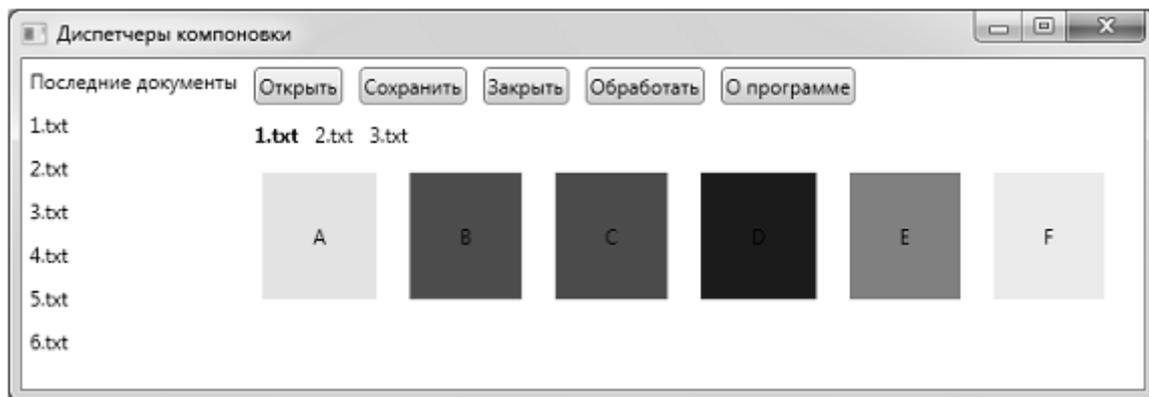


Рисунок 23.7 – Внешний вид приложения для варианта 7

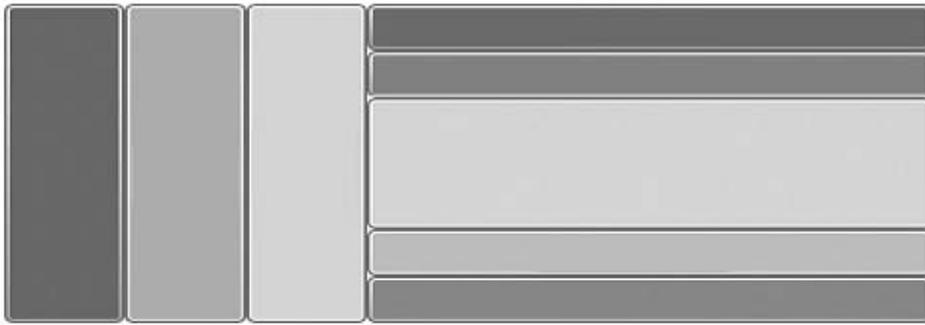


Рисунок 23.8 – Внешний вид приложения для варианта 8

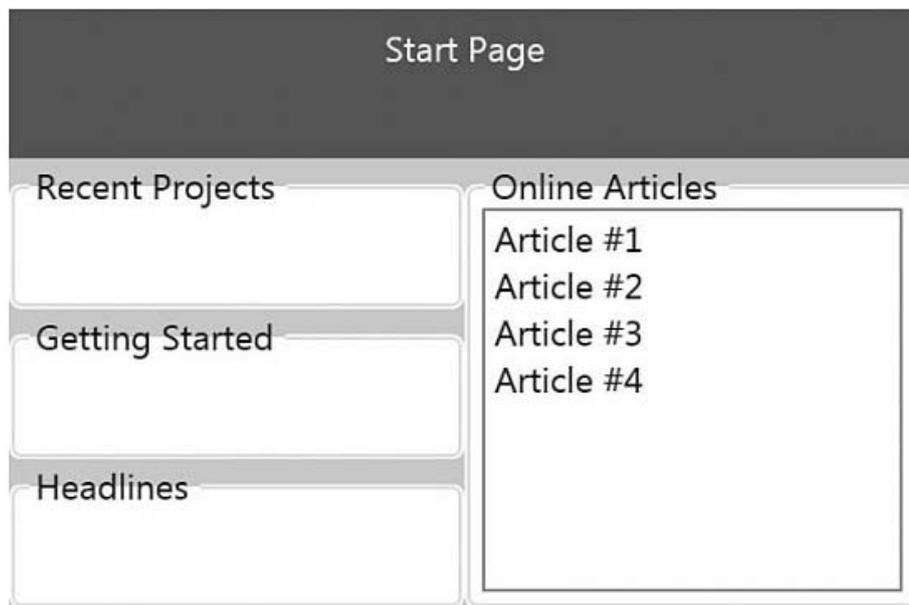


Рисунок 23.9 – Внешний вид приложения для варианта 9

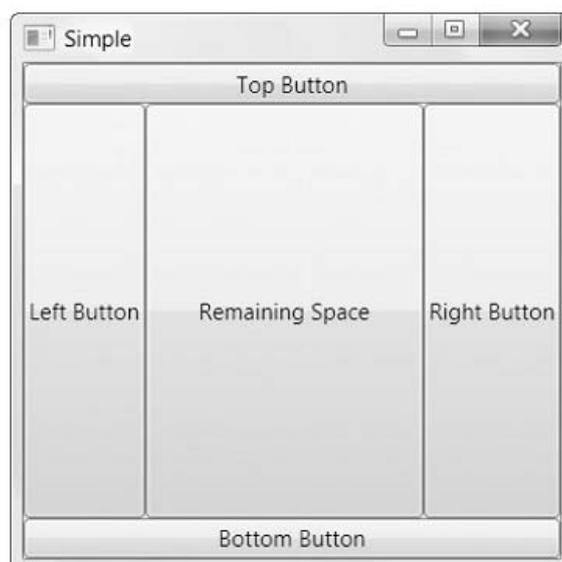


Рисунок 23.10 – Внешний вид приложения для варианта 10

### ***Контрольные вопросы***

- 1 Каким образом можно задавать размеры элементы управления?
- 2 Почему не рекомендуется задание абсолютных размеров?
- 3 На чем основана компоновка элементов в WPF?
- 4 Какой контейнер компоновки позволяет располагать элементы управления вдоль своих границ?
- 5 Какие есть особенности в применения преобразований?

## **24 Лабораторная работа № 24. Анимация в WPF-приложениях**

**Цель работы:** закрепить практические навыки по работе со свойствами и анимацией в коде на основе свойств.

### ***Порядок выполнения работы***

- 1 Написать программу согласно варианту.
- 2 В качестве отчета представить работающее программное обеспечение.
- 3 Защитить лабораторную работу.

### ***Варианты заданий***

1 Создать WPF-приложение с одной страницей, на которой расположены **Label** и две кнопки. При нажатии на первую кнопку происходит увеличение ширины элемента **Label** на случайное количество единиц, а при нажатии на вторую – уменьшение на случайное количество единиц. Предусмотреть выход размеров кнопки за пределы экрана, а также ее «исчезновение».

2 Создать WPF-приложение с одной страницей, на которой имеется кнопка. При наведении на нее указателя мыши она «пульсирует» (циклично меняет свой размер).

3 Разработайте WPF-приложение «Убегающая кнопка»: при наведении курсора мыши на кнопку она смещается на некоторое расстояние от курсора в случайном направлении. Кнопка не должна «убежать» за пределы окна.

4 Создать WPF-приложение с одной страницей, на которой имеется кнопка. При наведении на нее курсора мыши кнопка непрерывно увеличивается до пределов окна, а при отведении курсора – уменьшается до прежних размеров.

5 Создать WPF-приложение с одной страницей, на которой имеется кнопка. При нажатии на кнопку она увеличивается, а затем опять уменьшается до прежних размеров. И так бесконечно, пока на нее не нажмешь еще раз.

6 Создать WPF-приложение с одной страницей, на которой имеется кнопка. При наведении на нее курсора мыши кнопка быстро становится очень маленькой, а затем медленно возвращается к прежним размерам.

7 Создать WPF-приложение с одной страницей, на которой имеется кнопка. При нажатии на кнопку она ускоренно растет до размеров окна, при следующем нажатии – с замедлением уменьшается до прежних размеров.

8 Создать WPF-приложение с одной страницей, на которой имеется кнопка. При нажатии на кнопку она с ускорением «убегает» в случайный угол окна приложения.

9 Создать WPF-приложение с одной страницей, на которой имеется кнопка. При нажатии на кнопку она меняет свою высоту или ширину случайным образом. Предусмотреть выход размеров кнопки за пределы экрана, а также ее «исчезновение». При двойном щелчке по ней, она восстанавливает свои первоначальные размеры.

10 Создать WPF-приложение с одной страницей, на которой имеется кнопка. При наведении на кнопку она «прячется» в край окна приложения и потом медленно возвращается на место.

### ***Контрольные вопросы***

- 1 Перечислите основные классы анимации.
- 2 Какое свойство анимации устанавливает задержку перед ее запуском?
- 3 От чего зависит выбор класса анимации?
- 4 Какие свойства позволяют наследовать свои значения от родительского элемента?
- 5 Какое свойство анимации определяет, что произойдет со свойством по ее завершении? Как по умолчанию ведет себя анимация по своему завершению?
- 6 Объясните назначение класса **DependencyProperty**. Зачем нужны свойства зависимостей?

## **25 Лабораторная работа № 25. Триггеры в WPF-приложениях**

**Цель работы:** закрепить навыки по созданию WPF-приложений с использованием триггеров.

### ***Порядок выполнения работы***

1 Разработать WPF-приложение с двумя элементами управления. При изменении на один из них у обоих элементов одновременно срабатывает триггер, но у каждого свой. Какие элементы управления расположить на форме и какие изменения с ними должны происходить по триггеру, смотреть в таблице согласно своему варианту (таблица 25.1).

2 Разработать WPF-приложение с двумя многострочными текстовыми полями, кнопками «Открыть», «Очистить», «Закрыть» и выпадающим списком для задания внешнего вида текстовых полей. Задать для текстовых полей одинаковый градиентный фон. Кнопка «Закрыть» должна быть доступна только в том случае, если в обоих текстовых полях нет текста. Задать для кнопок различный внешний вид при наведении курсора и при нажатии на них. Внешний вид текстовых полей (тип шрифта, размер шрифта, цвет шрифта) должен меняться в зависимости от значения, выбранного в выпадающем списке.

- 3 В качестве отчета представить работающее программное обеспечение.
- 4 Защитить лабораторную работу.

Таблица 25.1 – Элементы управления и их изменения

Номер варианта	Элемент управления 1	Элемент управления 2
1	Button – цвет	Label – шрифт
2	Button – размер	Button – цвет
3	Label – цвет	Label – шрифт
4	Button – поворот	TextBlock – шрифт
5	Button – шрифт	Rectangle – поворот
6	TextBlock – шрифт	Label – сдвиг
7	Rectangle – размер	TextBlock – поворот
8	Rectangle – цвет	Rectangle – размер
9	TextBlock – цвет	TextBlock – шрифт
10	Rectangle – поворот	Label – цвет

### ***Контрольные вопросы***

- 1 Какие существуют виды триггеров?
- 2 Для чего могут применяться триггеры?
- 3 В чем заключаются преимущества и недостатки применения триггеров?
- 4 Что произойдет, если сработают несколько триггеров с конфликтующими наборами элементов **Setter**?
- 5 Можно ли к одному и тому же элементу применять несколько триггеров?

## **26 Лабораторная работа № 26. Использование фигур и кистей в WPF-приложениях**

**Цель работы:** закрепить практические навыки по WPF-приложений; получить навыки по созданию графических фигур и использованию кистей.

### **Порядок выполнения работы**

1 Разработать WPF-приложение «Переключатели». На экране изображены три одинаковых переключателя, которые при изменении размеров окна пропорционально увеличиваются или уменьшаются. Для создания переключателей следует использовать графические примитивы и градиентную заливку.

При наведении курсора на переключатель он увеличивается (рисунок 26.1).

При каждом клике по переключателю он плавно поворачивается на 20 град по часовой стрелке (рисунок 26.2).

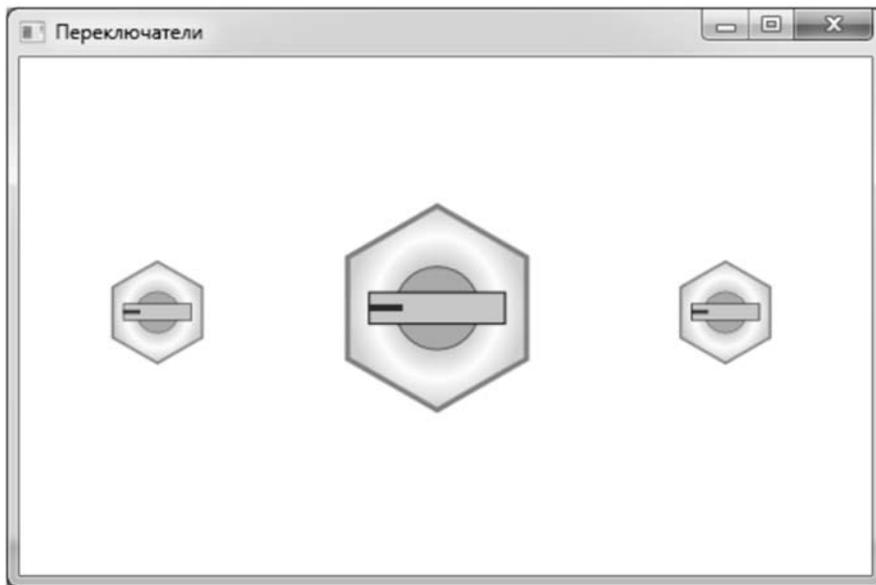


Рисунок 26.1 – Увеличение переключателя при наведении на него курсора.

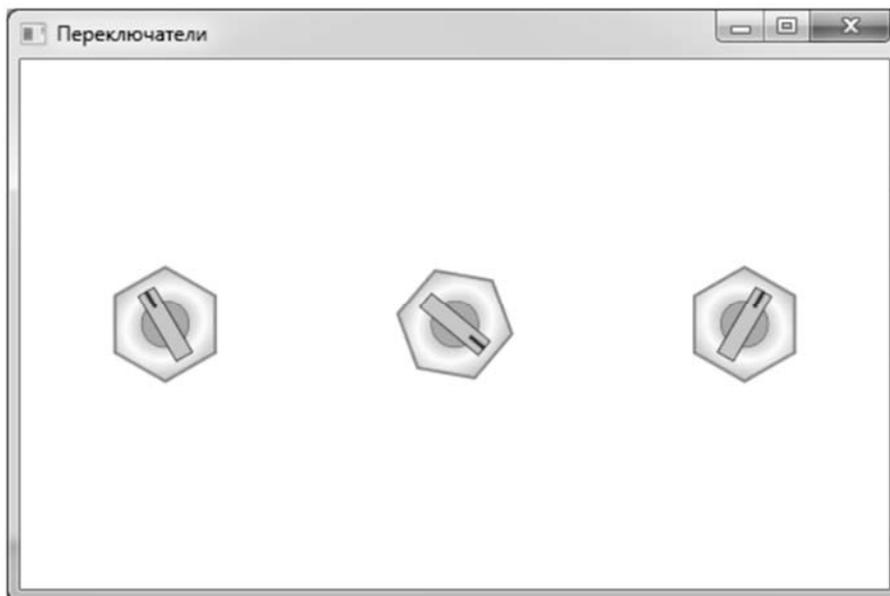


Рисунок 26.2 – Поворот переключателей после нажатия на них.

- 2 В качестве отчета представить работающее программное обеспечение.
- 3 Защитить лабораторную работу.

### ***Контрольные вопросы***

- 1 Перечислите известные Вам классы для создания двумерной графики.
- 2 Для чего применяется класса **Path**?
- 3 С помощью какого свойства можно создавать различные эффекты, связанные с прозрачностью?
- 4 Какие виды кистей существуют в WPF? Перечислите их основные классы.
- 5 Каким образом задается градиент?
- 6 Как создать сложную заливку для объектов классов двумерной графики?

## Список литературы

1 **Хорев, П. Б.** Объектно-ориентированное программирование с примерами на C#: учебное пособие / П. Б. Хорев. – Москва: ФОРУМ; ИНФРА-М, 2020. – 200 с.

2 **Гуриков, С. Р.** Введение в программирование на языке Visual C#: учебное пособие / С. Р. Гуриков. – Москва: ФОРУМ; ИНФРА-М, 2019. – 447 с.

3 **Бедердинова, О. И.** Программирование на языках высокого уровня: учебное пособие / О. И. Бедердинова, Т. А. Минеева, Ю. А. Водовозова. – Москва: ИНФРА-М, 2019. – 159 с.

4 **Рихтер, Д.** CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C# / Д. Рихтер. – 4-е изд. – Санкт-Петербург: Питер, 2016. – 896 с. : ил.

5 Приемы объектно-ориентированного проектирования. Паттерны проектирования / Э. Гамма [и др.]. – Санкт-Петербург: Питер, 2016. – 368 с.: ил.

6 **Шилдт, Г.** C# 4.0: полное руководство / Г. Шилдт. – Москва: Вильямс, 2015. – 1056 с.: ил.

7 **Мак-Дональд, М.** WPF 4.5: Windows Presentation Foundation в .NET 4.5 с примерами на C# 5.0 для профессионалов / М. Мак-Дональд. – Москва: Вильямс, 2013. – 1024 с.