

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

ФГБОУ ВО

Московский авиационный институт

(национальный исследовательский университет)

Кафедра 304

Вычислительные машины, системы и сети

Отчет по лабораторной работе №2

по учебной дисциплине «Конструирование ПО»

Выполнил студент группы МЗО-311Б-21:

Давыдов А.П.

Принял:

Грабовский М. Н.

Москва 2023

**Оглавление**

Задание 1:.....3

Задание 2:.....3

    Задача 2.1.....3

    Задача 2.2.....4

## Задание 1:

{Работа остановлена: Вопрос по ТЗ.}

{В примере уже сделан стек, и уже сделан приём. Как следствие, я не знаю что ещё добавить. Прошу уточнить задание}

## Задание 2:

### Задача 2.1

namespace WinForms\_Task2

```
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        int space = 0;

        // Памятка для тех, кто сталкивается с этим впервые:
        /*
        * Это Event или EventHandler (Событие) C#'а.
        * Так язык может поддерживать функции с инверсией зависимости.
        *
        * Точнее говоря, есть какой-то источник, какая-то функция, что вызывает
        Событие.
        * Эта функция не знает ничего об функциях объектов, что подписаны на события.
        * Тем не менее, она в одну сторону, как бы вслепую, кидает данные о том, что
        случилась функция.
        * А дальше её ловят те, кто на это событие подписан.
        *
        * Следующие две функции – как раз такие подписанные.
        */

        private void textBox1_KeyPress(object? sender, KeyPressEventArgs e)
        {
            if (sender is not TextBox textBox1)
                return;

            if (Char.IsLetter(e.KeyChar) || (e.KeyChar == ' '))
                if (textBox1.SelectionStart == 0)
                {
                    if (e.KeyChar == ' ') e.Handled = true;
                    else e.KeyChar = Char.ToUpper(e.KeyChar);
                }
                else if (e.KeyChar == ' ') if (space == 0) space = 1;
                else e.Handled = true;
                else if (space > 0) { e.KeyChar = Char.ToUpper(e.KeyChar); space++; }
                else { }
            else e.Handled = true;
        }

        private void textBox1_KeyUp(object? sender, KeyEventArgs e)
        {
            if (sender is not TextBox textBox1)
                return;

            if (space == 2)
```

```

        {
            textBox1.AppendText(".");
            textBox1.SelectionStart = textBox1.Text.Length;
            textBox1.SelectionLength = 0;
        }
        if (space == 3)
        {
            textBox1.AppendText(".");
            textBox1.SelectionStart = 0;
            textBox1.SelectionLength = textBox1.Text.Length;
            space = 0;
        }
    };
}

private void Form1_Load(object sender, EventArgs e)
{
    TextBox used = new()
    {
        Width = 200,
        Height = 50,
        Location = new Point(100, 100)
    };
    Controls.Add(used);
    used.KeyPress += textBox1_KeyPress;
    used.KeyUp += textBox1_KeyUp;
}
}
}

```

## Задача 2.2

```
namespace WinForms_Task2
```

```

{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        int space = 0;
        bool makeLetterUppcase = false;

        // Памятка для тех, кто сталкивается с этим впервые:
        /*
         * Это Event или EventHandler (Событие) C#'а.
         * Так язык может поддерживать функции с инверсией зависимости.
         *
         * Точнее говоря, есть какой-то источник, какая-то функция, что вызывает
Событие.
         * Эта функция не знает ничего об функциях объектов, что подписаны на события.
         * Тем не менее, она в одну сторону, как бы вслепую, кидает данные о том, что
случилась функция.
         * А дальше её ловят те, кто на это событие подписан.
         *
         * Следующие две функции – как раз такие подписанные.
         */

        private void textBox1_KeyPress(object? sender, KeyPressEventArgs e)
        {
            if (sender is not TextBox textBox1)
                return;

            if (Char.IsLetter(e.KeyChar) || (e.KeyChar == ' ')) // Либо введённый символ
пробел, либо – буква

```

```

    {
        if (textBox1.SelectionStart == 0) // SelectionStart - это выбранные
            символы, которые тут используются для определения состояний из KeyUp
            // Selection этот нужен, чтобы после
            ввода делать сразу новый, стирая всю выбор
        {
            if (e.KeyChar == ' ') // Мы в Selection, то-есть ввелось всё.
                Следующий введённый пробел просто стирает всё
                e.Handled = true; // Пропускаем этот пробел, хотя стираем им
                всё

            else // Ввод ещё не закончен,
                e.KeyChar = Char.ToUpper(e.KeyChar);
        }
        else if (makeLetterUppcase)
        {
            e.KeyChar = Char.ToUpper(e.KeyChar);
            makeLetterUppcase = false; // Есть смысл декомпозировать в
            функцию... Хотя это надо будет вообще всё переписать.
        }
        else if (e.KeyChar == ' ') // Текущий символ - пробел
        {
            if (space == 0) // Это первый пробел
                space++;
            else
                e.Handled = true; // Пропускаем этот пробел
        }
        else if (space > 0) // Уже был пробел, а значит следующие символы -
            инициалы
        {
            e.KeyChar = Char.ToUpper(e.KeyChar);
            space++; // У инициалов только одна буква, а потому прибавляем
            пробел
        }
    }
    else if (e.KeyChar == '-')
    {
        makeLetterUppcase = true;
    }
    else
        e.Handled = true; // Пропускаем этот символ
}

private void textBox1_KeyUp(object? sender, KeyEventArgs e)
{
    if (sender is not TextBox textBox1)
        return;

    if (space == 2)
    {
        textBox1.AppendText(".");
        textBox1.SelectionStart = textBox1.Text.Length;
        textBox1.SelectionLength = 0;
    }
    if (space == 3)
    {
        textBox1.AppendText(".");
        textBox1.SelectionStart = 0;
        textBox1.SelectionLength = textBox1.Text.Length;
        space = 0;
    }
};

private void Form1_Load(object sender, EventArgs e)
{
    TextBox used = new()

```

```
        {
            Width = 200,
            Height = 50,
            Location = new Point(100, 100)
        };
        Controls.Add(used);
        used.KeyPress += textBox1_KeyPress;
        used.KeyUp += textBox1_KeyUp;
    }
}
```