

Министерство образования Республики Беларусь
Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ
Факультет компьютерных систем и сетей
Кафедра информатики
ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовому проекту
на тему
Клавиатурный тренажёр C#

Студент гр. 953502

Слиж Максим Денисович

Руководитель

ассистент кафедры информатики Удовин Иван

Минск 2020

□ Введение

Выбор языка и среды разработки

Для написания этой курсовой работы выбран язык C#.

C# — компилируемый в CIL байт код, статически типизированный язык программирования общего назначения.

Поддерживает такие парадигмы программирования, как объектно-ориентированное программирование, процедурное программирование, обобщённое программирование. Язык имеет богатую стандартную библиотеку, которая включает в себя распространенные контейнеры и алгоритмы, ввод-вывод, регулярные выражения, поддержку многопоточности и другие возможности. C# сочетает свойства как высокоуровневых, так и низкоуровневых языков.

Интегрированная Среда Разработки (ИСР) – это среда, в которой есть все необходимое для проектирования, запуска и тестирования приложений, где все нацелено на облегчение процесса создания программ. ИСР интегрирует в себе редактор кодов, отладчик, инструментальные панели, редактор изображений, инструментарий баз данных, т.е. все, с чем приходится работать. Результатом является быстрая разработка сложных прикладных программ. Таким образом, IDE дает возможность получить EXE файл, не используя другие программы.

Для разработки программ на языке C# будет использована среда программирования Microsoft Visual Studio 2019 (ее английская аббревиатура – IDE: Integrated Development Environment), которая содержит в себе средства создания программы, ее компиляции, отладки и запуска на выполнение. В этой связи рассмотрим кратко структуру этой среды, а точнее ее интерфейс. Интерфейс – это аппарат, который позволяет удобно взаимодействовать пользователю со средой

Задание на курсовую работу: Разработать программу обучения работе с клавиатурой. Программа должна выдавать на экран буквы, цифры, слова и фразы, которые следует набрать на клавиатуре, и оценивать правильность и скорость набора. В программе надо предусмотреть три уровня подготовленности обучающегося. Предусмотреть сохранение статистики результатов обучающегося.

Круг пользователей приложения: пользователи, которым необходимо усвоить навыки работы с клавиатурой.

Информация, вводимая пользователем приложения: обучающие тексты, которые предлагает программа, в зависимости от выбранного уровня сложности. При старте игры есть возможность изменения обучающих текстов для каждого уровня подготовки.

(Изначально установлены значения:

Низкий уровень - русский алфавит

Средний уровень - отрывок песни L'one - Весь мир для нас

Высокий уровень - песня В.Цой - Кукушка)

Результаты, которые необходимо получить с помощью приложения: программа, которая позволит обучать пользователей работе с клавиатурой.

Требования к пользовательскому интерфейсу и приемы обработки ошибок пользователя: при вводе неверного текста пользователь будет уведомлен звуковым сигналом и обучение прекратится до тех пор, пока текст не будет набираться правильно.

Сохранение результатов: по окончании обучения ученику будет предложено сохранить результаты обучения, результаты хранятся в файле блокнота, им задается названием выбранного уровня.

- Описание принципа работы приложения в текстовой форме:

1. Запуск приложения
2. Выбор уровня сложности для подготовки
3. Редактирование обучающих текстов (для учителя)
4. Ввод имени обучающегося
5. Начало обучения
6. Набор текста в ходе работы программы
7. Сохранение результатов

Структура приложения

Таблица свойств объектов

EditForm.cs

Объект	Название	Другие свойства
richTextBox	richTextBox	-
Button	SaveButton	Text: Сохранить
	CancelButton	Text: Отмена

InputForm.cs

Объект	Название	Другие свойства
Label	SecondLabel	Text: Время:
	ErrorsLabel	Text: Ошибки:
	SpeedLabel	Text: Скорость набора:
	InputLabel	Text: Введите строку:
	TaskString	Text: (строка)
	InputString	Text: (ввод)
	LevelLabel	Text: Уровень:
	StringNumberLabel	Text: Строка:

-Таблица идентификаторов

Область видимости	Назначение	Идентификатор	Тип
InputForm.cs	Переменная имени	Name	string
	Переменная размера текста	InputSize	const int
	Переменная уровня	Level	int
	Переменная времени	Second	int
	Переменная ошибок	ErrorsCount	int
	Исправленный текст	CorrectChars	int
	Текущий текст	CurrentString	int
	Вводимые данные	Input	int
	Переменная строки	lines	int
EditForm.cs	Переменная уровня	Level	int
	Переменная текста	Path	string

3. Функциональное описание

- `InputForm_Shown()` – функция показа формы. При показе в коде устанавливаются счетчики ошибок вводимой информации и времени, при старте программы счетчики обнулены. Происходит проверка условий выбранного уровня, в зависимости от того, какой уровень выбран, программа решит, какой текст предложить для обучения. При записи каждого текста подсчитывается время и ошибки с помощью специальных операторов. Данные выводятся динамически по ходу обучения на форму.
 - `timer_Tick()` – функция времени. Позволяет считать время при каждом обучении.
 - `InputForm_KeyPress()` – функция нажатия клавиш в форме для обучения. Создается массив для хранения вводимых данных, вызывается функция подсчета времени, с помощью специальных расчетов производится вычисление количества вводимых букв в единицу времени. По окончании обучения по выбранному уровню приложение предлагает сохранить результаты или продолжить обучение.
- **Описание работы на примерах с представлением экранных форм**
- При запуске приложения необходимо указать своё имя.

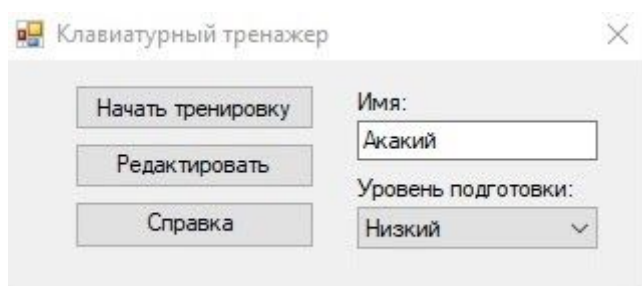


Рис.1. Запуск приложения

- Затем выбрать уровень сложности для обучения

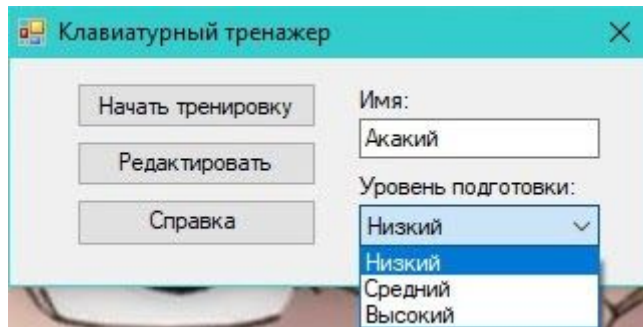


Рис. 2. Выбор уровня обучения

- По желанию можно изменять тексты для обучения

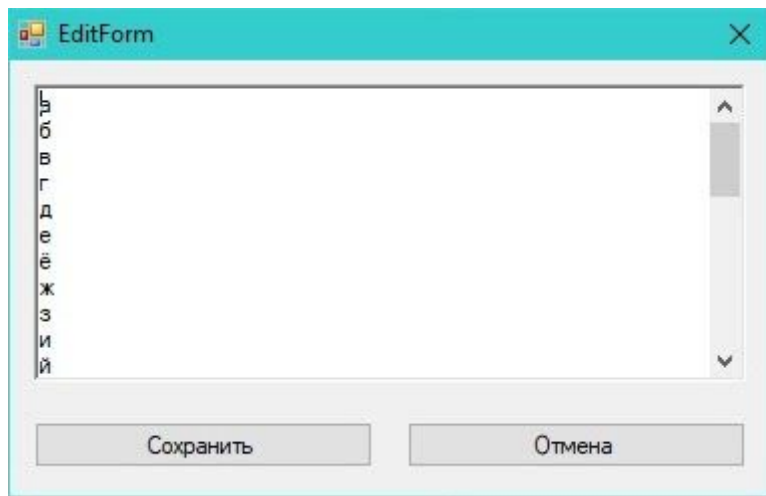
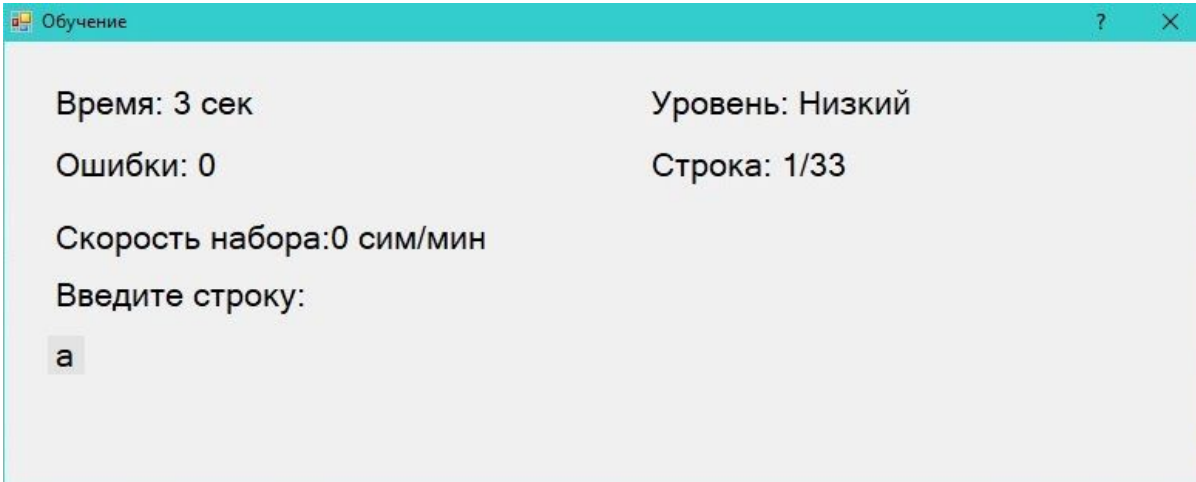


Рисунок 3. Изменение текстов для обучения

- После данных процедур необходимо нажать на кнопку «начать тренировку », перед Вами появится форма, после показа которой сразу необходимо набирать текст, в это же время будет считаться общее время набора текста и количество допущенных ошибок.

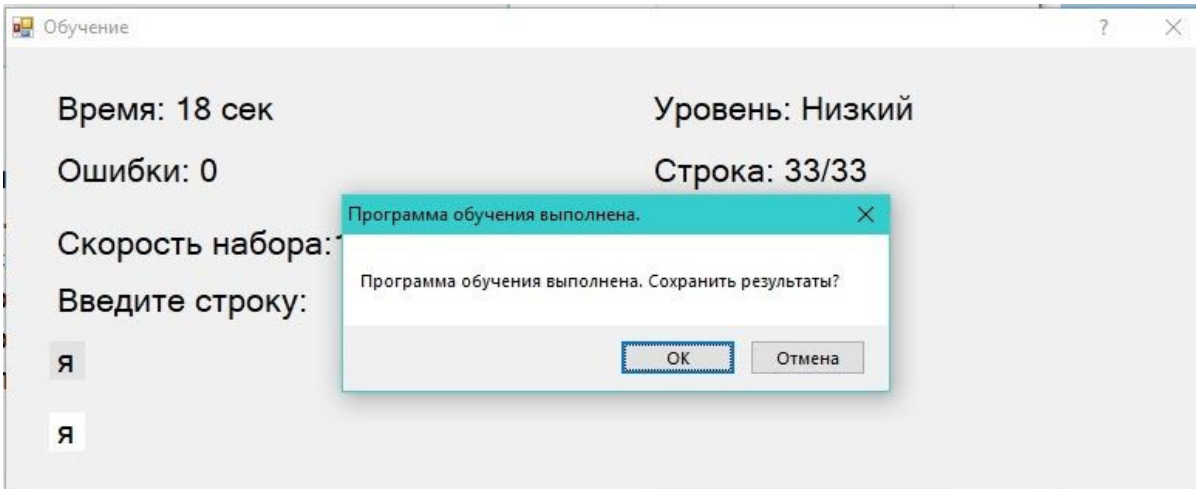


The screenshot shows a window titled 'Обучение' (Training). It displays the following information:

- Время: 3 сек (Time: 3 sec)
- Уровень: Низкий (Level: Low)
- Ошибки: 0 (Mistakes: 0)
- Строка: 1/33 (Line: 1/33)
- Скорость набора: 0 сим/мин (Typing speed: 0 symbols/min)
- Введите строку: (Enter the line:)
- Below the input field, the letter 'а' is visible.

Рис. 4. Обучение

- После завершения обучения программа предлагает сохранить результаты или не сохранять. Если пользователь сохраняет результаты, то они сохранятся в выбранном месте в файле блокнота с именем уровня, который обучающийся проходил. При отказе от сохранения программа примет начальное положение – на экране появится форма при запуске программы.



The screenshot shows the 'Обучение' window with updated statistics and a dialog box:

- Время: 18 сек (Time: 18 sec)
- Уровень: Низкий (Level: Low)
- Ошибки: 0 (Mistakes: 0)
- Строка: 33/33 (Line: 33/33)
- Скорость набора: (Typing speed:)
- Введите строку: (Enter the line:)
- Below the input field, the letter 'я' is visible.

A dialog box titled 'Программа обучения выполнена.' (Training program completed.) is overlaid on the window. It contains the text: 'Программа обучения выполнена. Сохранить результаты?' (Training program completed. Save results?). At the bottom of the dialog box are two buttons: 'ОК' (OK) and 'Отмена' (Cancel).

Рис. 5. Сохранение результатов

□ Заключение

Задание на курсовую работу успешно выполнено! Приложение функционирует исправно, в соответствии с требованиями по выполнению. В написании использовался язык программирования C# (WinForms).

Данная программа может стать отличным помощником людям, которые еще только изучают основы работы с компьютером. В перспективе программу можно дополнить подсказками и усложнить (например: сделать обучение по сочетаниям клавиш клавиатуры и так далее).

Приложение

Программный код

MainForm.cs

```
namespace Keyboard
{
    public partial class MainForm : Form
    {
        public InputForm input_form; // форма ввода текстов
        public EditForm edit_form; // форма изменения текстов
        public MainForm() // форма при запуске
        {
            InitializeComponent();
        }

        private void MainForm_Load(object sender, EventArgs e) // функция загрузки формы
        {
            input_form = new InputForm(); // определение новых форм
            edit_form = new EditForm();
            LevelComboBox.SelectedIndex = 0;
        }
    }
}
```

```
private void StartButton_Click(object sender, EventArgs e)//функция
нажатия кнопки нового обучения
```

```
{
    if (NameBox.Text == "")//если не введено имя
    {
        MessageBox.Show("Введите имя!");//уведомления
        return;
    }
    input_form.Level = LevelComboBox.SelectedIndex;//передача
индекса введенного уровня
    input_form.Name = NameBox.Text;//вывод текста
    input_form.ShowDialog();//установление диалога с формой
ввода
}
```

```
private void EditButton_Click(object sender, EventArgs e)//функция
нажатия кнопки для изменения текстов
```

```
{
    edit_form.Level = LevelComboBox.SelectedIndex;
    edit_form.ShowDialog();
}
}
```

EditForm.cs

```
namespace Keyboard
```

```
{
    public partial class EditForm : Form
    {
        public int Level;//определение переменной уровня
        public string Path;//текст
        public EditForm()//форма изменения текста
        {
            InitializeComponent();
        }
    }
}
```

```

    private void SaveButton_Click(object sender, EventArgs e)//ф-я
нажатия кнопки сохранения текстов
    {

        richTextBox.SaveFile(Path,
RichTextBoxStreamType.PlainText);//сохранение в элементе
управления
        this.Close();//закрытие после сохранения
    }

    private void EditForm_Shown(object sender, EventArgs e)//показ
формы изменения текстов
    {
        Path = Application.StartupPath;
        if (Level == 0)//если 0 уровень то файл сохранения будет с
названием "изи"
            Path += "\\easy.txt";
        if (Level == 1)//если 1 уровень то файл сохранения с
названием медиум
            Path += "\\medium.txt";
        if (Level == 2)//если 2 уровень то файл сохранения с
названием хард
            Path += "\\hard.txt";

        richTextBox.LoadFile(Path,
RichTextBoxStreamType.PlainText);//загрузка файлов
    }

    private void CancelButton_Click(object sender, EventArgs e)
    {
        this.Close();
    }
}

```

InputForm.Cs

```
namespace Keyboard
{
    public partial class InputForm : Form
    {
        //определение всех необходимых переменных
        public string Name;
        public const int InputSize = 65;
        public int Level;
        public int Second;
        public int ErrorsCount;
        public int CurrentChar;
        public int CorrectChars;
        public int CurrentString;

        public string Input;
        public string[] lines;
        public InputForm()
        {
            InitializeComponent();
        }

        private void InputForm_Shown(object sender, EventArgs e)
        {
            //установка счетчиков при запуске программы
            Second = 0;
            CurrentChar = 0;
            CurrentString = 0;
            ErrorsCount = 0;
            CorrectChars = 0;
            timer.Enabled = true; //таймер работает)

            //загрузка строк
            string Path = Application.StartupPath;
```

//вывод разного текста в зависимости от выбранной сложности (сложность проверяется в условиях)

```
if (Level == 0)
{
    Path += "\\easy.txt";
    LevelLabel.Text = "Уровень: Низкий";
}
if (Level == 1)
{
    Path += "\\medium.txt";
    LevelLabel.Text = "Уровень: Средний";
}
if (Level == 2)
{
    Path += "\\hard.txt";
    LevelLabel.Text = "Уровень: Высокий";
}
lines = System.IO.File.ReadAllLines(Path,
Encoding.GetEncoding(1251)); //чтение текстов

TaskString.Text = lines[CurrentString]; //запись в массив
InputString.Text = ""; //вывод
Input = "";

SecondLabel.Text = "Время: " + Second.ToString() + "
сек"; //вывод времени
ErrorsLabel.Text = "Ошибки: " +
ErrorsCount.ToString(); //ошибок
StringNumberLabel.Text = "Строка: " + (CurrentString +
1).ToString() + "/" + lines.Count().ToString(); //строки текста
SpeedLabel.Text = "Скорость набора: 0 сим/мин"; //скорости
}
```

```

private void timer_Tick(object sender, EventArgs e)//функция
таймера для подсчета показателей
{
    Second++;
    SecondLabel.Text = "Время: " + Second.ToString() + "
сек";//времени
    int InputSpeed = (int)((((float)CorrectChars / (float)Second) * 60);
    SpeedLabel.Text = "Скорость набора:" + InputSpeed.ToString()
+ " сим/мин"; //скорости набора текста
}

private void InputForm_KeyPress(object sender,
KeyPressEventArgs e)//функция ввода текста
{
    if (!timer.Enabled) return;//проверка условия работы таймера
    if (e.KeyChar.ToString() ==
lines[CurrentString][CurrentChar].ToString())
    {

        Input += e.KeyChar.ToString();
        //ВВОД текста
        InputString.Text = Input;
        TaskString.Text = lines[CurrentString];
        if (CurrentChar > InputSize / 2)
        {
            InputString.Text = Input.Substring(CurrentChar - InputSize /
2);
            TaskString.Text =
lines[CurrentString].Substring(CurrentChar - InputSize / 2);
        }
        //динамическое изменение показателей и вывод на экран
        CurrentChar++;
        CorrectChars++;
        int InputSpeed = (int) (((float)CorrectChars / (float)Second) *
60);

```

```

        SpeedLabel.Text = "Скорость набора:" +
InputSpeed.ToString() + " сим/мин";
    }
    else//при неверном наборе текста - звуковые уведомления
    {
        System.Media.SystemSounds.Beep.Play();
        ErrorsCount++;
    }
    if (CurrentChar == lines[CurrentString].Length)
    {

        CurrentChar = 0;
        CurrentString++;
        if (CurrentString == lines.Count())
        {
            timer.Enabled = false;
            if (MessageBox.Show("Программа обучения выполнена.
Сохранить результаты?", "Программа обучения выполнена.",
MessageBoxButtons.OKCancel) == DialogResult.OK)
                if (saveFileDialog.ShowDialog() == DialogResult.OK)
                {
                    string[] SaveLines = { "Имя обучающегося: " + Name,
LevelLabel.Text, SecondLabel.Text, ErrorsLabel.Text, SpeedLabel.Text
};

                    System.IO.File.WriteAllLines(saveFileDialog.FileName,
SaveLines);
                }
                this.Close();
            }
            else
            {
                TaskString.Text = lines[CurrentString];
                InputString.Text = "";
                Input = "";
            }
        }
    }

```

```

        StringNumberLabel.Text = "Строка: " + (CurrentString +
1).ToString() + "/" + lines.Count().ToString();
    }

    ErrorsLabel.Text = "Ошибки: " + ErrorsCount.ToString();
}

}
}

```

ЗАКЛЮЧЕНИЕ

В результате выполнения курсового проекта была разработана тестирующая программа по C#, проверяющая знания пользователя по данному языку программирования. В ходе его разработки были получены теоретические и практические знания по языку программирования C#, а также по принципам объектно-ориентированного программирования.

Список использованных источников

1. Герберт Шилдт C# 4.0 Полное руководство, 2010
2. Эндрю Троелсен, Филипп Джепикс Язык программирования C#7 и платформы .NET и .NET Core, 8-е издание, 2017
3. METANIT[Электронный ресурс] – METANIT:

<https://metanit.com>

4. .WIKIPEDIA [Электронный ресурс] – WIKIPEDIA:

<https://ru.wikipedia.org>

5. #SimpleCode[YouTube канал]

<https://www.youtube.com/channel/UCtLKO1Cb2GVNrbU7Fi0pM0w>

