How to create Typing Test.

Typing test in an app that measures a user’s typing speed and accuracy. The user sets desirable time for the test, chooses the level of difficulty and starts the test. The test stops in three cases: when time is over, when the phrase the user entered matches a testing phrase, when the user pressed the “Stop testing” button. Having finished the test the user sees the results: test status, time spent on the test, speed of typing and a number of words typed with mistakes.

Here is an instruction to create such an app, WinForms Framework, C#. It includes the next steps:

[1. Creating a TabControl and placing the necessary elements on the form](#_Toc81691853)

[2. Time setting](#_Toc81691854)

[3. Switching between the tabs](#_Toc81691855)

[4. Getting a test phrase](#_Toc81691856)

[5. Timer control](#_Toc81691857)

[6. Writing the method “Start”](#_Toc81691858)

[7. Writing the method “Stop”](#_Toc81691859)

[8. Writing the method “CompareTestPhrase”](#_Toc81691860)

[9. Counting statistics](#_Toc81691861)

[10. Displaying the results of the Typing test](#_Toc81691862)

[11. Starting Typing test again](#_Toc81691863)

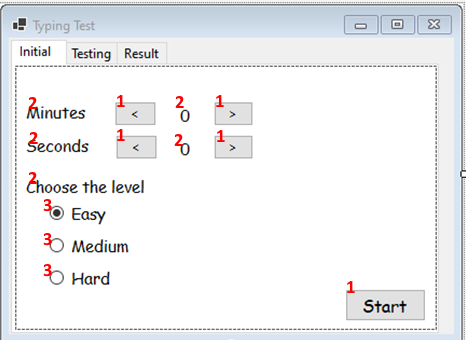
[12. Making flat buttons of tab pages invisible](#_Toc81691864)

You can get this project from GitHub <https://github.com/LenaPesochek/project/tree/main/TypingTest>

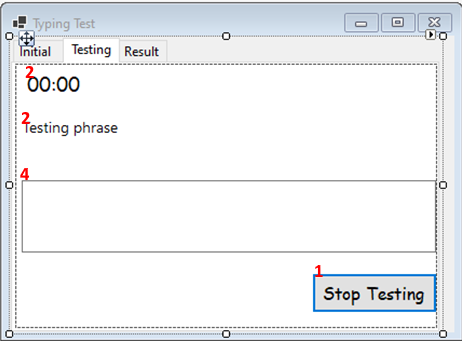
# Creating a TabControl and placing the necessary elements on the form

The TabControl consists of three items:

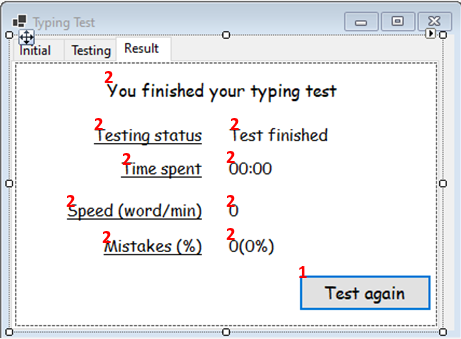
1. Input data (here we have buttons to set time for testing, radio buttons to choose the level and a button to start testing)



1. Testing (there are labels that display the time left for testing and a testing phrase, textbox to enter words and a button to stop testing)



1. Results output (on this tab the results of testing are displayed: testing status, time spent, speed and a number of mistakes)



To add the elements to the form you should drag them from the toolbar. The elements used are buttons(1), labels(2), radio buttons(3) and textbox(4).

Time setting and switching between the tabs

# Time setting

To set time for testing you should use event “button\_click”. For example, to increase the number of minutes you should write the next method:

private void MinuteIncreasingButton\_Click(object sender, EventArgs e) {

var minuteCount = int.Parse(MinuteLabel.Text);

minuteCount++;

MinuteLabel.Text = minuteCount.ToString();

}

Using the same logic write the methods for the other SetTimeButtons.

# Switching between the tabs

To set the necessary Tab active write the method “SelectTabByName”:

public void SelectTabByName(string name) {

TabControl.SelectTab(name);

}

# Getting a test phrase

The phrases you get for the typing test need to be stored in a separate file. Every line in the file contains difficulty and a phrase according to the mask. For example:

[Easy]Hello, world!

All phrases: <https://github.com/LenaPesochek/project/blob/main/TypingTest/TypingTest/Content/phrases.txt>

In order to get a test phrase according to the chosen difficulty you should write the next method:

string GetTestPhrase() {

var difficultyInPhrase = "[" + Difficulty.ToString() + "]";

var currentDifficultyPhrases = File.ReadLines(phrasesPath).Where(x => x.Contains(difficultyInPhrase));

var phrase = currentDifficultyPhrases.Skip(new Random().Next(0, currentDifficultyPhrases.Count() - 1)).First();

phrase = phrase.Replace(difficultyInPhrase, "");

return phrase;

}

# Timer control

* + - Creating a field Timer timer = new Timer();
    - In a Testing class constructor we add an event handler and set an interval in which the event is raised

public Testing(Form form) {

…;

timer.Tick += new EventHandler(TimerTick);

timer.Interval = 1000;

}

# Writing the method “Start”

To start testing you should write the method “Start” that runs the timer and displays the test phrase.

public void Start(int startValue, CurrentDifficulty difficulty) {

timerCurrentValue = startValue;

timerStartValue = startValue;

Difficulty = difficulty;

timer.Start();

TestingForm.SetTestingPhraseLabelText(TestPhrase);

}

# Writing the method “Stop”

To stop testing you should write the method “Stop” that stops the timer. This method will be called either if the input phrase matches the test phrase, or when pressing the “Stop testing” button, or when time is over. Parameter “forced”: Boolean (true, if the method is called by pressing the “Stop testing” button, otherwise false)

public void Stop() {

timer.Stop();

TestingForm.SelectTabByName("ResultTab");

}

# Writing the method “CompareTestPhrase”

To compare an input phrase to the test phrase you need to transform these phrases to arrays by the method “String.Split”.

public void CompareTestPhrase(string inputPhrase) {

InputPhrase = inputPhrase;

if(InputPhrase == "") {

MistakesCount = 0;

CorrectWordsCount = 0;

return;

}

var inputPhraseArray = GetSplitInputPhrase();

if(InputPhrase == TestPhrase) {

MistakesCount = 0;

CorrectWordsCount = inputPhraseArray.Length;

Stop(false);

return;

}

var testPhraseArray = GetSplitTestPhrase();

var mistakesCount = 0;

for(var i=0; i<inputPhraseArray.Length; i++) {

if(inputPhraseArray[i] != testPhraseArray[i])

mistakesCount++;

}

MistakesCount = mistakesCount;

CorrectWordsCount = inputPhraseArray.Length – MistakesCount;

}

Every time this method is called the value of the property “MistakesCount” will be overwritten. The “MistakesCount” represents a number of entered words which have a mistake or are unfinished.

# Counting statistics

* + A number of mistakes is counted in the method “CompareTestPhrase”
  + Percentage of mistakes = (the number of mistakes) / (a number of entered words) <https://github.com/LenaPesochek/project/blob/main/TypingTest/TypingTest/Testing.cs#L24>
  + Time spent on testing = (time for testing) – (time left when the timer stopped) <https://github.com/LenaPesochek/project/blob/main/TypingTest/TypingTest/Testing.cs#L74>
  + Speed of printing = (a number of correct entered words) / (time spent on testing) <https://github.com/LenaPesochek/project/blob/main/TypingTest/TypingTest/Testing.cs#L81>
  + Setting of testing status: “Test is finished”, “Test is stopped”, “Time is over” <https://github.com/LenaPesochek/project/blob/main/TypingTest/TypingTest/Testing.cs#L109>

# Displaying the results of the Typing test

To display the results of the test you need to write a public method in the class “Form”:

public void FillResultTab(string testStatus, string testTime, string speed, int mistakes, int mistakesPercent) {

TestResultLabel.Text = testStatus;

TestTimeLabel.Text = testTime;

SpeedLabel.Text = speed;

MistakesCountLabel.Text = string.Format("{0} ({1}%)", mistakes, mistakesPercent);

}

This method is called in the method “Stop” in the class “Testing” <https://github.com/LenaPesochek/project/blob/main/TypingTest/TypingTest/Testing.cs#L67>

# Starting Typing test again

To start the typing test again it’s necessary to set default values to the form controls properties “text” and “checked” by writing a method “ClearTestingForm” (<https://github.com/LenaPesochek/project/blob/main/TypingTest/TypingTest/Form.cs#L102>) and to set “InitialTab” active:

private void TestingAgainButton\_Click(object sender, EventArgs e) {

SelectTabByName("InitialTab");

ClearTestingForm();

}

# Making flat buttons of tab pages invisible

To make flat buttons of tab pages invisible we should set TabControl properties in the method “InitializeComponent” in the class Form:

this.TabControl.Appearance = System.Windows.Forms.TabAppearance.FlatButtons;

this.TabControl.ItemSize = new System.Drawing.Size(0, 1);

this.TabControl.SizeMode = System.Windows.Forms.TabSizeMode.Fixed;