

СОЗДАНИЕ ТЕСТИРУЮЩЕЙ СИСТЕМЫ В ПРИЛОЖЕНИИ WINDOWS FORMS(C#)

*Тазетдинова Ю.А., к. ф.-м. н., доцент
Габдуллин Д.Р., студент 1 курса ф.-м. факультета
Плетнев М.Э., студент 1 курса ф.-м. факультета
Осиев Е.Г., студент 1 курса ф.-м. факультета
г. Бирск, ФГБОУ ВО Бирский филиал УУНУТ*

Аннотация. В работе описан процесс создания тестирующей системы в приложении Windows Forms. Разработан интерфейс и функционал программы на платформе Microsoft Visual Studio 2022. На главном окне приложения и расположен тест. Произведено тестирование приложения, которое не выявило ошибок. Данные, представленные в работе, можно использовать для создания тестовых систем подобного типа.

Ключевые слова: тестирование, приложение Windows Forms, C#.

Актуальность и выбор темы состоит в том, что проблема контроля за уровнем знаний, умений и навыков учащихся – вечная проблема образования, и в разные времена предлагалось решать её по-разному, преобладали то одни формы контроля, то другие, создавались контролирующие системы. Тест может служить методом педагогического измерения на любом этапе обучения. Тестовые задания эффективны и для самопроверки знаний. Использование тестов позволяет сэкономить массу времени, проверить широкий спектр знаний учащихся по предмету, уровень сформированности некоторых навыков и умений.

Для создания программы мы используем программу Visual Studio 2022 года. В Visual Studio при создании проекта используем шаблон приложение Windows Form и будем работать на языке C#.

Windows Forms — интерфейс программирования приложений (API), отвечающий за графический интерфейс пользователя и являющийся частью Microsoft .NET Framework. Данный интерфейс упрощает доступ к

элементам интерфейса Microsoft Windows за счет создания обёртки для существующего Win32 API в управляемом коде. Причём управляемый код — классы, реализующие API для Windows Forms, не зависят от языка разработки. То есть программист одинаково может использовать Windows Forms как при написании ПО на C#, C++, так и на VB.Net, J# и др.

Все начинается с объекта **Form** на нем мы расположим другие объекты. Внешний вид приложения является нам преимущественно через формы. Формы являются основными строительными блоками. Они предоставляют контейнер для различных элементов управления. А механизм событий позволяет элементам формы отзываться на ввод пользователя, и, таким образом, взаимодействовать с пользователем. У каждого объекта есть свои свойства, могут повторяться или же быть уникальными.

Для создания программы мы использовали такие объекты как **Label**, **Button**, **TextBox**, **CheckBox**, **RadioButton** и **GroupBox**. Для отображения простого текста на форме, доступного только для чтения, служит элемент **Label**.

Button. Обработывая событие нажатия кнопки, мы может производить те или иные действия. Для ввода и редактирования текста предназначены текстовые поля - элемент **TextBox**. Элемент **CheckBox** или флажок предназначен для установки одного из двух значений: отмечен или не отмечен. Чтобы отметить флажок, надо установить у его свойства **Checked** значение **true**. Кроме свойства **Checked** у элемента **CheckBox** имеется свойство **CheckState**, которое позволяет задать для флажка одно из трех состояний - **Checked** (отмечен), **Indeterminate** (флажок не определен - отмечен, но находится в неактивном состоянии) и **Unchecked** (не отмечен). На элемент **CheckBox** похож элемент **RadioButton** или переключатель. Переключатели располагаются группами, и включение одного переключателя означает отключение всех остальных. Чтобы установить у

переключателя включенное состояние, надо присвоить его свойству `Checked` значение **true**.

Для создания группы переключателей, из которых можно бы было выбирать, надо поместить несколько переключателей в какой-нибудь контейнер, например, в элементы **GroupBox** или **Panel**. Переключатели, находящиеся в разных контейнерах, будут относиться к разным группам.

GroupBox представляет собой специальный контейнер, который ограничен от остальной формы границей. Он имеет заголовок, который устанавливается через свойство `Text`. Чтобы сделать **GroupBox** без заголовка, в качестве значения свойства `Text` просто устанавливается пустая строка. Нередко этот элемент используется для группирования переключателей - элементов **RadioButton**, так как позволяет разграничить их группы.

Для элементов **Label**, **Button**, **CheckBox**, **RadioButton** и **GroupBox** мы можем изменить текст что находится на них, используя свойство `Text`. Пример: `Button.Text = "Проверить ответ"`.

Используя элементы **TextBox**, **CheckBox** и **RadioButton**, мы можем создать три типа вопросов: вопрос требующий вода ответа, вопрос требующий несколько ответов и вопрос требующий один ответ соответственно.

Для **TextBox** мы пишем условие: `if (textBox.Text== "Ответ")`, `textBox.Text` это одно из свойств **TextBox**, отвечающие за то, что мы видим в качестве текста в **TextBox**.

Условие **if** заработает после нажатия **Button** благодаря условию `button_Click`, являющийся часть контейнера **Button**.

Если текст был вбит в **TextBox** и была нажата **Button** мы покажем элемент **Label**, который мы предварительно скрыли условием `Visible`

установив его на значение **False**, и после используя команду `label.Visible=true`, сделаем его снова видимым. После используем свойство **Text** изменим вид текст на слово верно, командой `label.Text = "Верно"`. В случае если ответа нет или он не был вбит, то мы выведем ответ “Неверно” или “Не введен текст” соответственно.

Для элементов **CheckBox** и **RadioButton** мы используем те же действия кроме условия и случая, когда не выбран ответ. В последнем случае мы меняем текст на “Не выбран, ни один вариант”.

Для **CheckBox** мы пишем условие `if (checkBox1.Checked==true) & (checkBox2.Checked==true) & (checkBox3.Checked==false) & (checkBox4.Checked== false)`. **checkBox1** и **checkBox2** это имена элементов.

А для **RadioButton** условием является `if (radioButton1.Checked==true)`.

Для того, чтобы начать выполнение теста, достаточно запустить программу. При выполнении теста пользователь должен ответить на 10 вопросов. Приложение будет анализировать ответы и выводить результат.

Пользователь сможет увидеть свои ошибки и проанализировать их.

The screenshot shows a window titled "Form1" with a test interface. The interface is divided into two main sections: "С выбором одного или нескольких ответов" (Multiple choice) and "С выбором одного ответа" (Single choice). Each question has a "проверка" (check) button and a "Результат" (result) label. The questions are:

- Какие символы разрешается использовать в имени файла или имени директории в Windows?
 - Цифры
 - Латинские, русские буквы
 - Символы, знаки препинания
- Какой символ заменяет любое число любых символов?
 - ?
 - *
 - !
- Единицы измерения информации: ...
 - бит
 - терабайт
 - гигабайт
 - гигагад
- Какое расширение у исполняемых файлов?
 - exe, doc
 - bak, bat
 - exe, com, bat
- Какой символ заменяет только один символ в имени файла?
 - ?
 - *
 - !
- Какое расширение у исполняемых файлов?
 - exe, doc
 - bak, bat
 - exe, com, bat
- Может ли в одном каталоге быть два файла с одинаковыми именами?
 - Да
 - Нет
- Может ли в одном каталоге быть два файла с одинаковыми именами?
 - Да
 - Нет
- Основоположником отечественной вычислительной техники является:
 - Золотарев Лев Викторович
 - Попов Зиксандр Глебович
 - Лебедев Сергей Алексеевич
- Во время исполнения прикладная программа хранится в памяти...
 -

Рис. 1. Главный экран приложения.

Созданное приложение имеет очень простой графический интерфейс, дает возможность педагогам в режиме реального времени оценить уровень подготовки учеников, что влечет за собой сокращение времени на оценивание учащихся. Основная работа по обработке информации возлагается на вычислительную технику.

Литература

1. Шилдт Г. Полный справочник по C#. – М.: Издательский дом “Вильямс”, 2004. – 744 с.
2. Павловская Т.А. C#. Программирование на языке высокого уровня. Учебник для вузов. – СПб.: Питер, 2009. – 432 с.
3. Что такое label? URL:
<https://metanit.com/sharp/windowsforms/4.2.php> (дата обращения 19.03.2023)
4. Основы форм. URL:
<https://metanit.com/sharp/windowsforms/2.1.php> (дата обращения 19.03.2023)
5. Элементы управления. URL:<https://metanit.com/sharp/windowsforms/4.1.php> (дата обращения 19.03.2023)