#### Министерство науки и высшего образования Российской Федерации

#### Федеральное государственное бюджетное образовательное учреждение

#### высшего образования

#### «Владимирский государственный университет

#### имени Александра Григорьевича и Николая Григорьевича Столетовых»

**(ВлГУ)**

**Кафедра информационных систем и программной инженерии**

Лабораторная работа №4

по дисциплине «Алгоритмы и структуры данных»

Выполнил:

студент гр. ЗИСТуд-121

Коркишко Д.А.

Приняла:

Шамышева О.Н.

Владимир, 2021 г.

**Задания**

Задание 1.

* Создать две формы.
* На кнопку ”Показать форму2” Form1 написать следующий обработчик, который будет открывать вторую форму.
* На кнопку «Закрыть форму2» Form2 написать следующий обработчик, который будет закрывать Form2.
* На событие FormClose написать следующий обработчик, который будет возвращать текст label в прежнее состояние “сообщение” при закрытии Form2.
* Написать обработчик на кнопку «Изменить текст сообщения на форме1» Form2, при выполнении которого «Сообщение» меняется на «Новое сообщение».
* Написать обработчик на кнопку «Изменить название формы2» в Form1, который меняет название формы2.
* Изменить обработчик, чтобы вторая форма открывалась в модальном режиме.

Задание 2.

* На главной форме расположите 2 кнопки – TButton; 2 кнопки – Ttool Button; два пункта меню, два пункта всплывающего меню.
* Вторая форма содержит 10 Button – ов и Edit1. Вторая форма вызывается динамически – на 1-ые кнопки, из формы Form1 – все кнопки на второй форме принимают значение 0, 1, 2, …9; на вторые кнопки из формы Form 1 – все кнопки на второй форме принимают значение а, б, в, …к.
* При создании второй формы динамически создается кнопка с названием «Close», обработчик на эту кнопку закрывает вторую форму и разрушает Form 2.
* При нажатии на любую из кнопок 2-ой формы в Edit1 на Form 2 добавляется название нажатой кнопки.
* При закрытии Form 2 на форме 1 динамически создается Edit в который помещается результат из Edit1 form2.

**Выполнение**

Выполнение заданий было реализовано с использованием языка C#.

Выполнение первой задачи

Для начала был создан проект Windows Forms (рисунок 1).

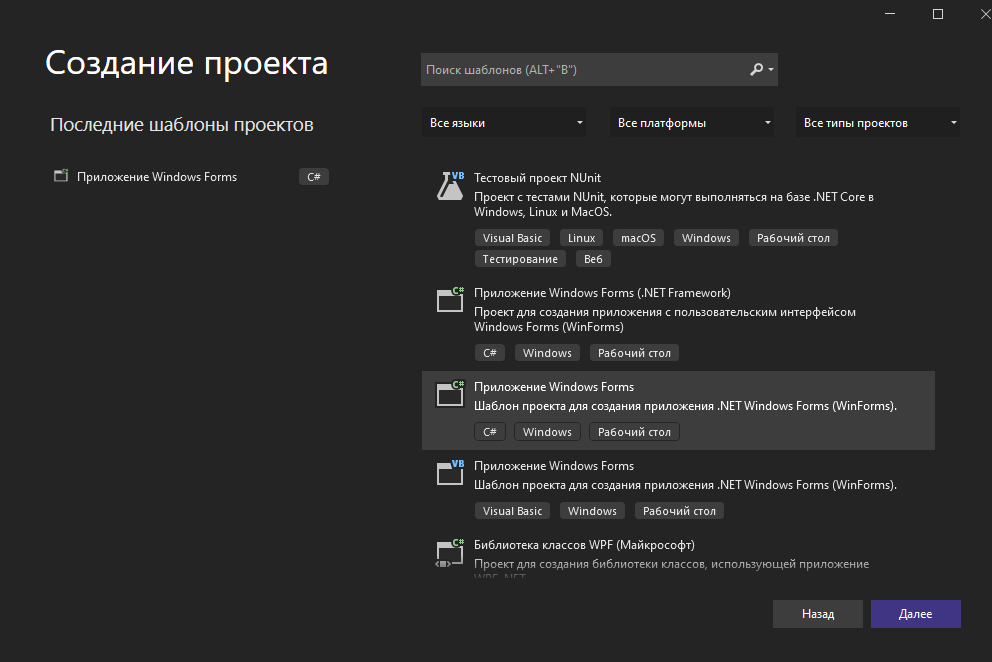


Рисунок 1 – Создание приложения

Далее в макете приложения были созданы формы расположены и добавлены элементы согласно заданию (рисунок 2, рисунок 3).

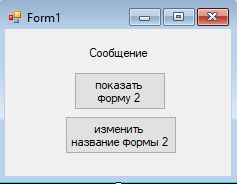


Рисунок 2 – Первая форма Form1

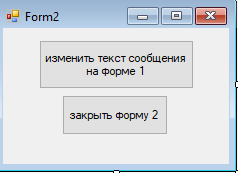


Рисунок 3 – Вторая форма Form2

Далее были созданы обработчики на нажатие кнопку “Показать форму 2” в Form1 (рисунок 4).

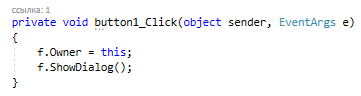


Рисунок 4 – Обработчик нажатия на кнопку “Показать форму 2”

На рисунке 4, открытие второй формы является модельным как сказано в одном из заданий. Результат работы обработчика указан ниже.

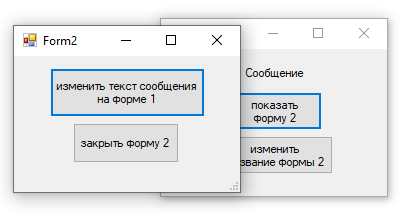


Рисунок 5 – Результат работы обработчика

Следующим шагом было добавление обработчиков для нажатия на кнопку ”Закрыть форму 2” (рисунок 6).

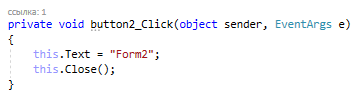


Рисунок 6 – Обработчики нажатия на кнопку ”Закрыть форму 2”

В обработчике перед закрытие восстанавливается название формы “Form2” и закрывается сама форма. Результатом работы обработчика представлен ниже.

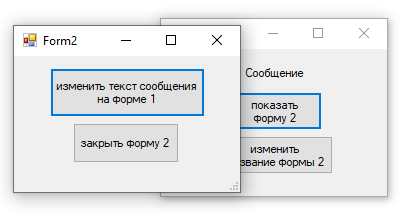


Рисунок 7 – До нажатия на кнопку

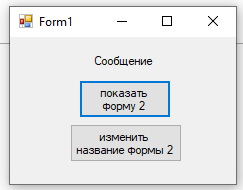


Рисунок 8 – После нажатия на кнопку

Далее в обработчик открытие Form2 была добавлена строчка, для возвращения текста label в обратное значение “Сообщение” (рисунок 9).

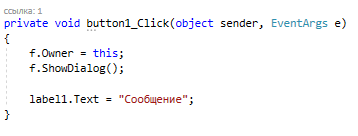


Рисунок 9 – Дополненный обработчик открытия второй формы

Далее следует написать обработчик на кнопку «Изменить текст сообщения на форме1» Form2, при выполнении которого «Сообщение» меняется на «Новое сообщение» (рисунок 10).

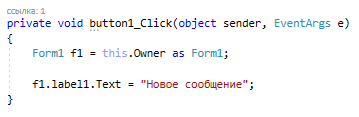


Рисунок 10 – Обработчик нажатия на кнопку

Результат работы обработчика представлен ниже

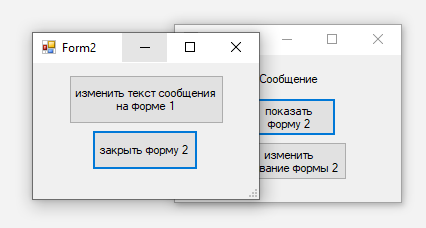


Рисунок 11 – До нажатия на кнопку

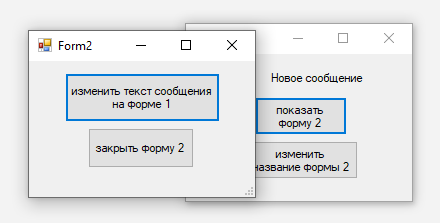


Рисунок 12 – После нажатия на кнопку

Далее следует написать обработчик на кнопку «Изменить название формы2» в Form1, который меняет название формы2 (рисунок 13).

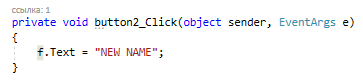


Рисунок 13 – После нажатия на кнопку

Результат работы обработчика представлен ниже.

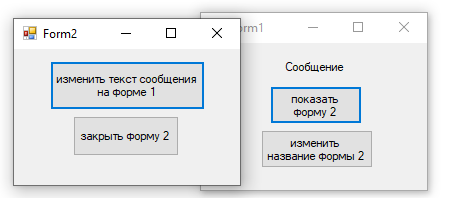


Рисунок 14 – До нажатия на кнопку

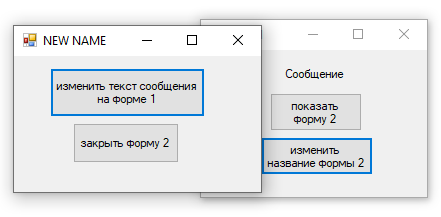


Рисунок 15 – После нажатия на кнопку

Выполнение второй задачи

Для начала были созданы формы, которые представлены ниже

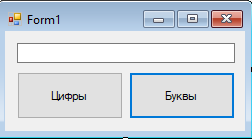


Рисунок 10 – Первая форма Form1

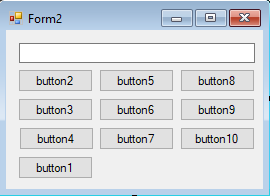


Рисунок 11 – Вторая форма Form2

Для того чтобы вторая форма вызывалась динамически – на 1-ые кнопки, из формы Form1 – все кнопки на второй форме принимают значение 0, 1, 2, …9; на вторые кнопки из формы Form 1 – все кнопки на второй форме принимают значение а, б, в, …к, был написан обработчики нажатия на кнопки. Обработчики представлены ниже.

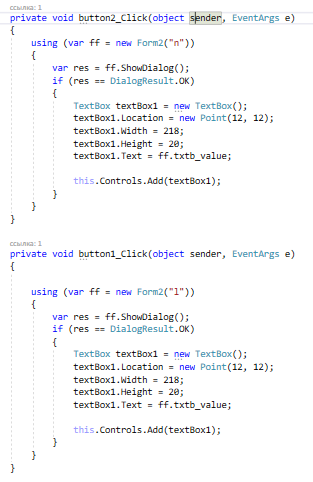


Рисунок 12 – Обработчики нажатия на кнопки

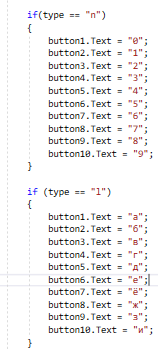


Рисунок 13 – Обработчик в Form2

Результат работы обработчиков представлен ниже.

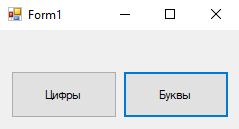


Рисунок 14 – До нажатия на кнопки

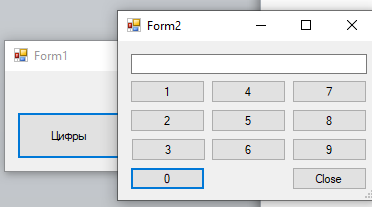


Рисунок 15 – Нажатие на кнопку «Цифры»

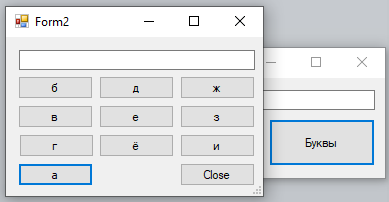


Рисунок 16 – Нажатие на кнопку «Буквы»

Чтобы кнопка «Close» динамически создавалась, был написан обработчик (рисунок 16).

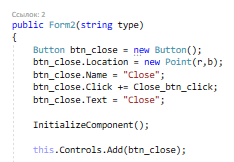


Рисунок 17 – Динамическое создание кнопки

Для того чтобы при нажатии на любую из кнопок 2-ой формы в Edit1 на Form 2 добавляется название нажатой кнопки и при закрытии Form 2 на форме 1 динамически создается Edit в который помещается результат из Edit1 form2 были написаны обработчики представленные ниже.

.

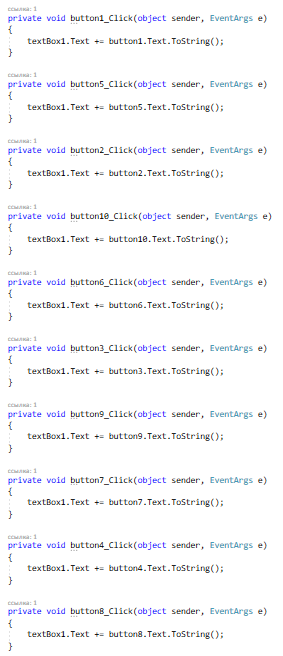


Рисунок 18 – Обработчик нажатия на кнопки второй формы

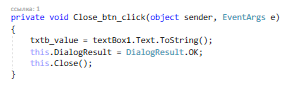


Рисунок 18 – Обработчик кнопки «Close» и возвращение   
значения в первую форму

Результаты работы обработчиков представлены ниже.

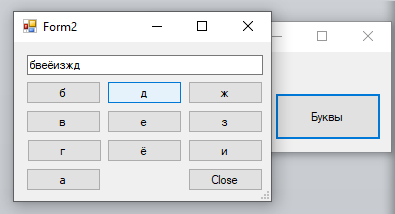


Рисунок 18 – Ввод значения в форму

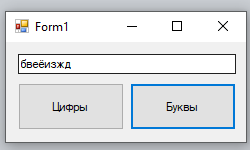


Рисунок 18 – После нажатия на кнопку «Close»

**Контрольные вопросы**

1. Из каких файлов состоит приложение? Какие файлы (с каким расширением) каким модулям соответствуют?

Любой проект имеет, по-крайней мере, шесть файлов, связанных с ним. Три из них относятся к управлению проектом из среды и напрямую программистом не меняются. Вот эти файлы : · Главный файл проекта, изначально называется PROJECT1.DPR. · Первый модуль программы /unit/, который автоматически появляется в начале работы. Файл называется UNIT1.PAS по умолчанию, но его можно назвать любым другим именем, вроде MAIN.PAS. · Файл главной формы, который по умолчанию называется UNIT1.DFM, используется для сохранения информации о внешнем виде главной формы. · Файл PROJECT1.RES содержит иконку для проекта, создается автоматически. · Файл, который называется PROJECT1.OPT по умолчанию, является текстовым файлом для сохранения установок, связанных с данным проектом. Например, установленные Вами директивы компилятора сохраняются здесь. · Файл PROJECT1.DSK содержит информацию о состоянии рабочего пространства. Разумеется, если сохранить проект под другим именем, то изменят название и файлы с расширением RES, OPT и DSK. После компиляции программы получаются файлы с расширениями: DCU - скомпилированные модули EXE - исполняемый файл DSM - служебный файл для запуска программы в среде, очень большой, рекомендуется стирать его при окончании работы. ~PA, ~DP - backup файлы Редактора.

1. Как создать новое приложение в Delphi?

Чтобы открыть новое приложение выберете в меню File\New\Application – откроется новый проект с одной формой Form1(модуль Unit1)

1. Функции модуля проекта?

Чтобы добовить в проект еще модуль выберете в меню File\New\Form – добавиться форма Form2 (модуль Unit2); Также можно добавить новый модуль выбрав кнопку «New» на панели интрументов окно на котором можно видеть все типы модулей. Выберите Forms. В ваш проект добавиться форма Form2 ( модуль Unit2). Чтобы от формы перейти к Инспектору Объектов и к коду программы нажимайте F11; Чтобы добавить модули и формы на просмотр используйте пункт меню View\Unit ,Forms. Добавьте на просмотр модуль проекта Project1 . Этот модуль не связан ни с одной формой.

1. Какие вы знаете типы модулей в Delphi?

Все типы модулей в Delphi находятся на странице New Items ( см.выше). Рассмотрим некоторые из них:

- Application - главный файл проекта (.dpr);

- Data Module – модуль данных (.dtm);

- Form – модуль – форма (.frm);

- Report - модуль – отчета;

- Thread Object - модуль – потока;

- Unit - модуль – для программы Object Pascal (.pas);

1. Как добавить новый модуль в проект? Как добавить существующий модуль?

Чтобы добовить в проект еще модуль выберете в меню File\New\Form – добавиться форма Form2 (модуль Unit2); Также можно добавить новый модуль выбрав кнопку «New» на панели интрументов окно на котором можно видеть все типы модулей.

1. Как просматривать модули?

Чтобы добавить модули и формы на просмотр используйте пункт меню View\Unit, Forms.

1. Как установить “видимость” одного модуля из другого?

Чтобы в Form1 можно было использовать компоненты Form2 ( и наоборот) необходимо выбрать меню File\Use Unit и после выбора нужного модуля нажать OK.

1. Как обратиться к компонентам, находящимся в другом модуле?

К компонентам, находящимся на Form2 можно обратиться из Unit1 , указав сначала их принадлежность к Form1 : Например : Form1.Button1.Caption := ‘Новое название’;

1. Что такое модуль-форма? Какие вы знаете свойства и методы компонента Формы.

Форма – это важнейший визуальный компонент, который представляет собой видимое окно Windows. В системе Delphi форма является компонентом, который служит контейнером для всех других компонентов. Приложение может иметь несколько форм, одна из которых считается главной и при запуске программы отображается первой.

1. Чем отличаются модальные формы от не модальных?

Другим отличием этих методов является то, что не модальная форма не возвращает никакого значения при завершении работы, а модальная форма при завершении возвращает в качестве результата одно из значений перечисления.

1. Что необходимо сделать, чтобы Form2 создавалась динамически на событие onClick компонента Button1?

Для того, чтобы создать динамически форму необходимо: - вызвать пункты меню Project-Options и переместить динамически создаваемую форму в окно Available forms. Теперь форма Form1 будет создаваться при запуске проекта, а форму Form2 необходимо создавать внутри программы, скажем на обработчик события onClick компонента Button.

1. Как организовать вызов формы с параметром?

Для того, чтобы загружать одну и туже форму (например Form2) с разными параметрами необходимо в Unit2 объявить процедуру ( например execute с параметром deys : char).

1. Как динамически создать компонент?

Чтобы создать динамически компонент необходимо написать следующее:

begin

edit1:= TEdit.Create(Form2); //создание компонента типа

TEdit form2.InsertControl(edit1); //вставить компонент в коллекцию компонентов на

//форме, где разместили компонент типа

TEdit. Edit1.Name := 'Edit1'; // присвоили имя компоненту.

edit1.Left := 100; // Задание координат.

edit1.Text := ''; // Очистка строки ввода.

end;

1. Как обратиться к свойствам и методам динамически созданного компонента?

Если требуется создать обработчик, например на событие onClick на динамически создаваемую кнопку , то необходимо в коде программы написать следующее:

begin

button := Tbutton.create(form2);

form2.InsertControl(button);

button .Name := 'Button';

Button.OnClick := Form2.ButtonClick;

End;

1. Как организовать событие динамически созданного компонента?

Все используемые процедуры в программе должны быть описаны в разделе implementation и объявлены в разделе interface. Здесь придется этим заняться вручную - т.е. объявить процедуру Form2.ButtonClick и записать ее реализацию.

1. Что такое Sender?

Параметр Sender – это ссылка на компонент, который вызвал данный обработчик . Если процедуру вызовет Button1 то Sender будет указывать на Button1; если процедуру вызовет Button2 то Sender будет указывать на Button2.

1. Чем операция is отличается от as?

Существуют две операции, которые часть путают между собой . Операция is: дает True , если типы совместимы или False в противном случае. Например, (Sender is TButton) = True если обработчик был вызван любой кнопкой. Операция as возвращает компонент и мы можем обратиться к любым его свойствам и методам. Перед использованием as всегда необходимо ‘проверить’ sender на is.

1. Что такое контейнер объектов? Что такое коллекция объектов данного контейнера?

Controls – это коллекция объектов, раcположенных на контейнере объекто. К каждому компоненту можно обратиться по индексу Controls[i]. Нумерация начинается с 0 до (ControlCount -1). ControlCount - количество объектов на форме.

1. Что такое MDI приложение.

MDI приложение – это приложение, в котором имеется одна главная форма, и одну (редко несколько) дочерних форм.

1. Какое свойство накапливает количество дочерних форм.

1) Откройте новый проект. Cвойство Form1.FormStyle установите в fsMDIForm. Добавьте на форму панель инструментов.

2) Добавьте в приложение вторую форму и свойство Form1.FormStyle установите в fsMDIChild. Теперь ваше приложение является MDI приложением.

3) Объявите Form2 как динамически создаваемую форму. Установите область видимости таким образом, чтобы главная форма “видела” дочернюю форму. Чтобы создать дочернюю форму (на событие onclick кнопки панели инструментов) необходимо воспользоваться методом create.

1. Как можно обратиться к дочерней форме.

Для того, чтобы иметь возможность закрывать дочерние формы, можно воспользоваться свойством ActiveMDIChild - активная дочерняя форма procedure TForm1.ToolButton2Click(Sender: TObject);

begin

ActiveMDIChild.Free;

end;