Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт перспективной инженерии

Департамент цифровых, робототехнических систем и электроники

**ОТЧЕТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ №2**

**дисциплины**

**«Системы реального времени»**

**Вариант 11**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Выполнил:  Репкин Александр Павлович  3 курс, группа ИВТ-б-о-22-1,  09.03.01 «Информатика и вычислительная техника», направленность (профиль) «Программное обеспечение средств вычислительной техники и автоматизированных систем», очная форма обучения  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | | Проверил:  Ассистент департамента цифровых, робототехнических систем и электроники Громаков В.А.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | |  | |

Отчет защищен с оценкой \_\_\_\_\_\_\_\_\_\_\_ Дата защиты\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ставрополь, 2024 г.

**Тема:** Изучение иерархической архитектуры сервисов

**Цель:** ознакомиться с принципами разделения программного кода, особенностями работы с нотификациями и модулями.

**Порядок выполнения работы:**

1. Запущена программа для визуального программирования. В соответствии с вариантом (№11) получено индивидуальное задание: создать программу с декомпозицией на основе использования модуля: 1. Организовать добавление входящих и исходящих сообщений модуля. 2. С помощью простого диалога ввести своё имя. 3. При нажатии кнопки «Приветствие» в модуле составляется предложение: “Здравствуйте, “Имя»”! Как дела?”. Сначала в диаграмме “Button” были установлены блоки, позволяющие получить от пользователя его имя и устанавливающие текст для кнопки – “Приветствие”, после чего установлено 3 блока – обработка нажатия кнопки, получение имени пользователя из переменной name и передача имени пользователя активности. В Activity потребовалось установить два блока – Calculate (Для создания надписи “Здравствуйте, Имя\_Пользователя! Как дела?”) и SimpleDialog, выводящий данное сообщение.

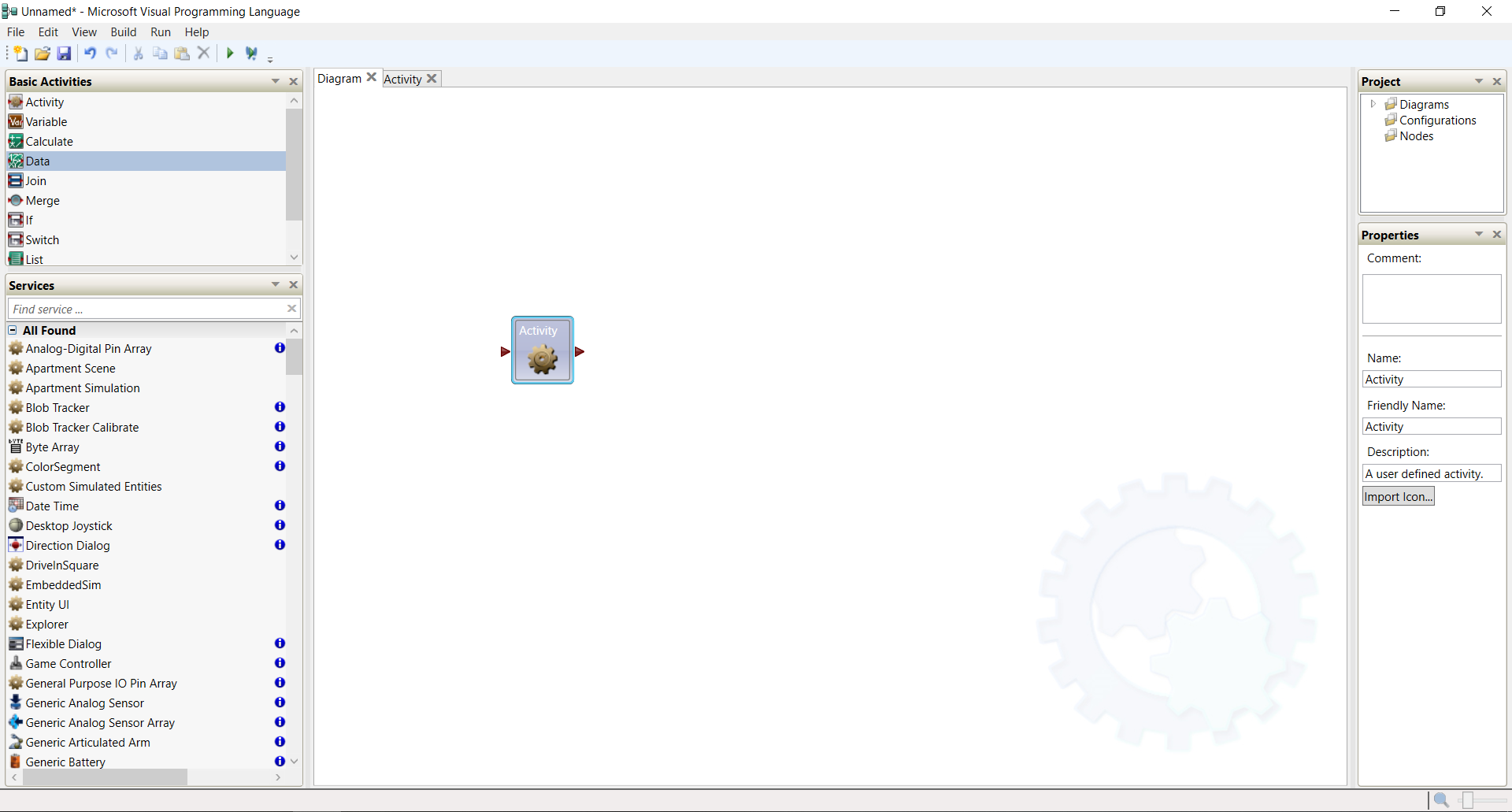


Рисунок 1 – Запущенная программа Microsoft Visual Programming Language и блок Activity

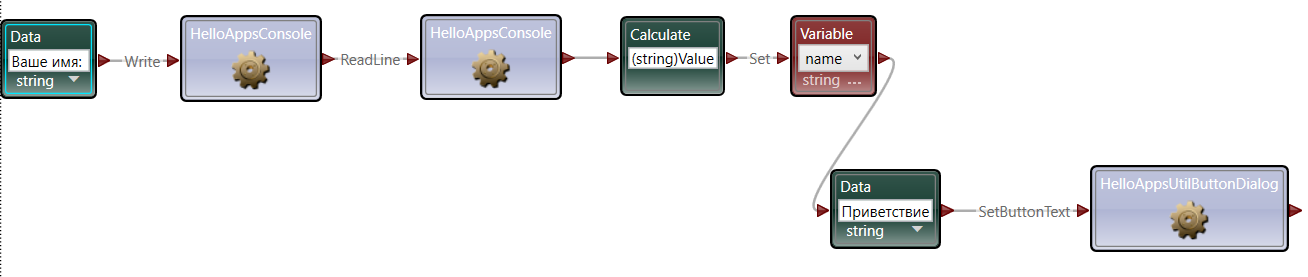


Рисунок 2 – Для кнопки устанавливается текст “Приветствие” и получение имени пользователя



Рисунок 3 – Обработка пользователем нажатия на кнопку, вызов Activity с именем пользователя

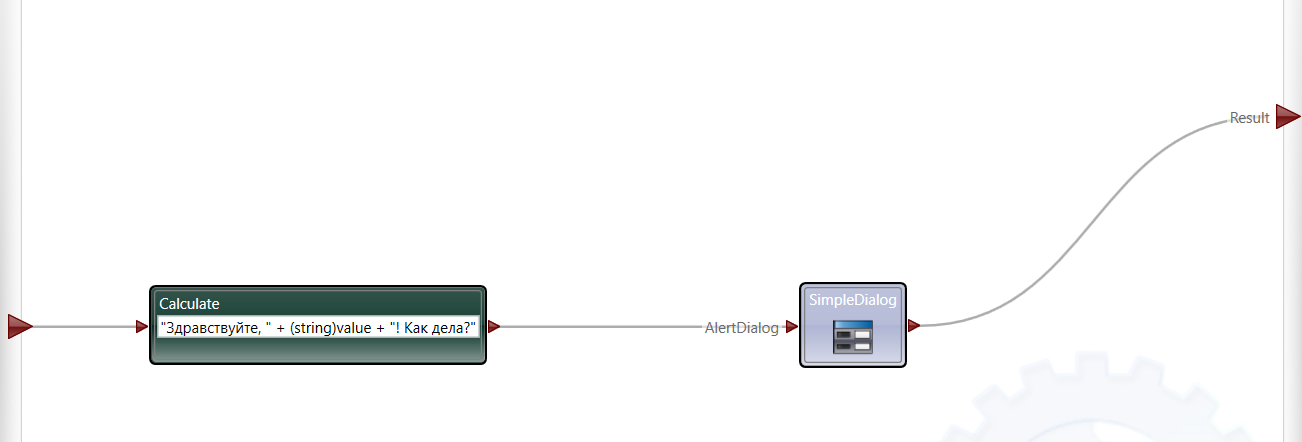


Рисунок 4 – Содержимое Activity, вывод SimpleDialog с именем пользователя

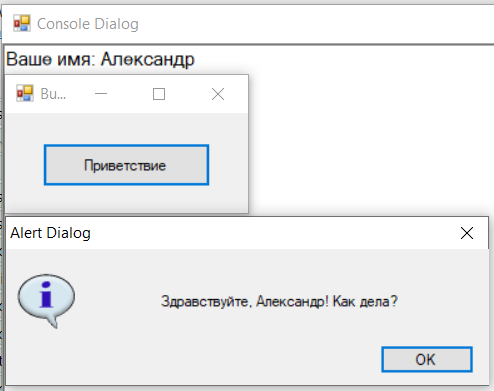


Рисунок 5 – Пример выполнения программы

**Ответы на контрольные вопросы:**

1. Определение модуля?

**Ответ:** модуль – это замкнутая программа, которую можно вызвать из любого другого модуля в программе и можно отдельно компилировать.

1. Факторы, способствующие созданию модульных программ?

**Ответ:** создание модульных программ объясняется следующими факторами:

1. Программные модули решают небольшую функциональную задачу, используют на входе и на выходе немного данных; внутренние переменные модуля не связаны с внутренними переменными других модулей => отдельные модули могут создаваться и отлаживаться различными разработчиками независимо друг от друга.
2. Модульные программы легко читать, сопровождать и модифицировать; исправление отдельного модуля вызывает минимальные изменения в других модулях, связанных с ним по управлению и информации.
3. Модульные программы обладают повышенной надежностью, так как при их разработке существует возможность распределения работ по созданию модулей различной сложности и важности между программистами различной квалификации.
4. Можно создавать библиотеки наиболее употребительных подпрограмм, которые затем можно использовать в качестве комплектующих частей при разработке других приложений.
5. Процедура загрузки всей программы в оперативную память упрощается при использовании метода оверлейности.
6. Возникает много естественных контрольных точек для наблюдения за продвижением проекта по управлению и по информации.
7. Иерархия – это?

**Ответ:** иерархия – свойство упорядоченного множества компонентов, между которыми установлено отношение приоритета (Компоненты, между которыми отсутствует предпочтительность, образуют один иерархический уровень.).

1. Свойства, присущие архитектуре программного обеспечения как иерархической системе?

**Ответ:** архитектуре ПО как иерархической системе присущ ряд свойств, важнейшими из которых являются:

1. Вертикальная соподчиненность – последовательное упорядоченное расположение взаимодействующих компонентов, составляющих данный комплекс программ.
2. Компоненты одного уровня обеспечивают реализацию функций компонентов следующего уровня.
3. Каждый уровень иерархии реализуется через функции компонентов более нижних уровней.
4. Каждый компонент знает о компонентах более низких уровней и ничего не знает о компонентах более высоких уровней.
5. Право вмешательства и приоритетного воздействия на компоненты любых уровней со стороны компонентов более высоких иерархических уровней.
6. Взаимозависимость действий компонентов верхних уровней от реакций на воздействия и от функционирования компонент нижних уровней, информация о которых передается верхним уровням.
7. Шаги, позволяющие организовать декомпозицию программы с помощью нотификации?

**Ответ:** нотификация – сообщение о состоянии сервиса. Сервис изменяет свое состояние при поступлении в него сообщений от других сервисов или блоков. На примере таймера это может выглядеть как, например истечение установленного времени. В случае кнопочного интерфейса – нажатие на кнопку пользователем. В этом случае в сервисе формируется сообщение, которое может быть отправлено через разъем нотификации. Шаги, позволяющие организовать декомпозицию программы с помощью нотификации и модуля на примере индивидуального задания:

1. Установка текста на кнопке.

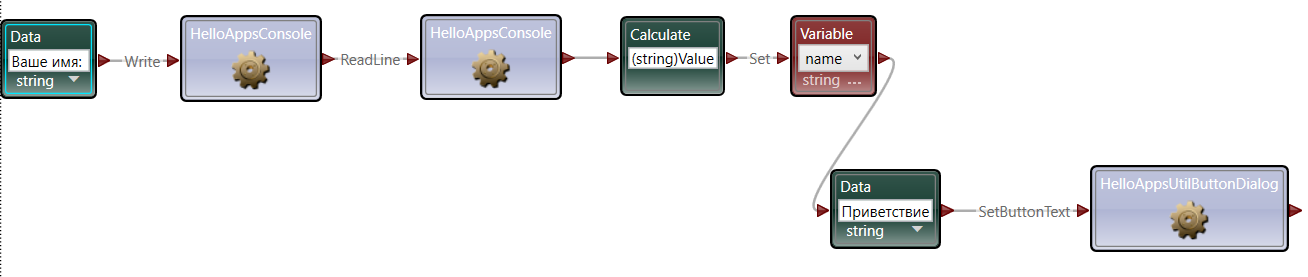


Рисунок 6 – 1 Шаг декомпозиции программы

1. Установка блоков для выполнения Activity при нажатии на кнопку.



Рисунок 7 – 2 Шаг декомпозиции программы

1. Заполнение Activity блоками для выполнения нужных действий.

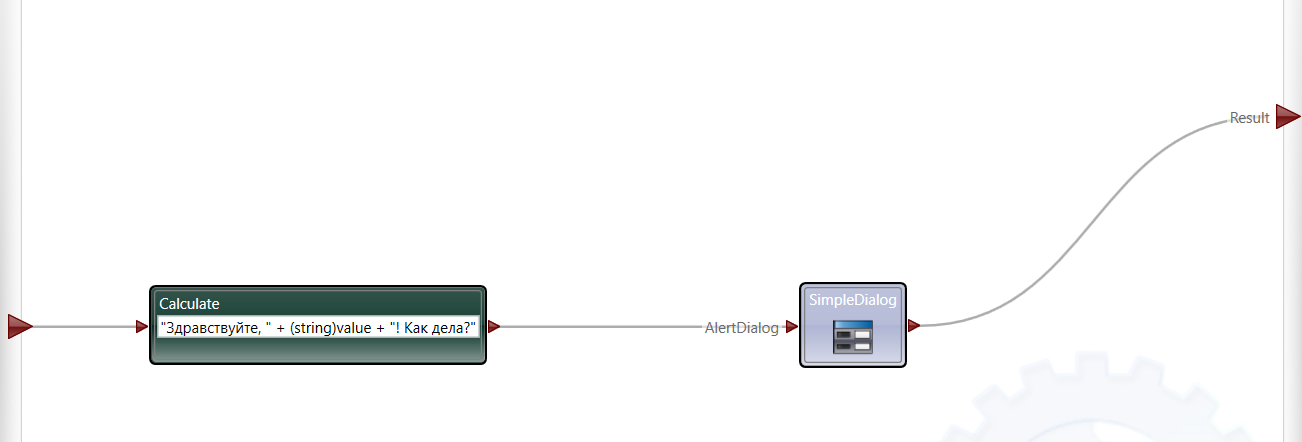


Рисунок 8 – 3 Шаг декомпозиции программы

1. Запуск программы для проверки.

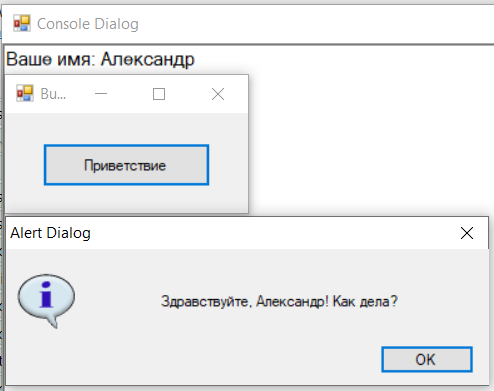


Рисунок 9 – Проверка декомпозиции программы

1. Понятие и элементы модуля в VPL?

**Ответ:** модуль (Activity) – это блок, внутри которого можно создавать диаграммы для экономии места в рабочей области диаграммы. В итоге получаются вложенные друг в друга диаграммы. Созданный модуль можно скомпилировать как сервис и использовать в дальнейшем написании программ.

1. Шаги, позволяющие организовать декомпозицию программы с помощью модуля?

**Ответ:** для декомпозиции программы при помощи модуля необходимо: добавить сервис и модуль, откройте модуль двойным щелчком, добавить в модуль блоки и сервисы, заполнить блоки, соединить входящие и исходящие разъемы, соединить сервис с модулем, запустить на выполнение программу.

**Вывод:** в ходе выполнения лабораторной работы проведено ознакомление с принципами разделения программного кода, особенностями работы с нотификациями и модулями.