

Лабораторная работа №7

Разработка адаптируемого ПО

1 Цель работы

1.1 Изучение процесса разработки ПО с поддержкой внешней интеграции при помощи скриптов.

2 Литература

2.1 Зверева В. П., Сопровождение и обслуживание программного обеспечения компьютерных систем : учебник для студ. учреждений сред. проф. Образования / В. П. Зверева, А. В. Назаров. – М. : Издательский центр «Академия», 2018. – 256 с.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание лабораторной работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Реализация поддержки скриптинга

5.1.1 Создать новый проект оконного приложения

5.1.2 Установить пакет Microsoft.CodeAnalysis.CSharp.Scripting

5.1.3 Создать класс Automation который будет содержать набор методов доступных для внешнего доступа. Добавьте в класс метод Message, который принимает строку и выводит MessageBox пользователю.

5.1.4 Создать класс ScriptingHost. Данный класс предназначен для обработки и выполнения скриптов. Добавить в класс метод для выполнения кода скрипта

```
public async Task ExecuteAsync(string code)
{
    await CSharpScript.EvaluateAsync(code, scriptOptions, new
Automation());
}
```

5.1.5 scriptOptions определить приватным полем и инициализировать в конструкторе

```
scriptOptions = ScriptOptions.Default
.WithReferences(AppDomain.CurrentDomain.GetAssemblies())
.WithImports("System", "System.Collections.Generic",
"System.Linq");
```

5.1.6 Добавить в программу поле ввода для скриптов, и кнопку «Выполнить». Реализовать их функциональность.

5.1.7 Протестировать работу скриптов.

5.2 Поддержка внешних скриптов

5.2.1 Добавить в ScriptingHost метод Run() в котором создать

именованный канал для приема команд извне

```
while (true)
{
    using (NamedPipeServerStream server = new
NamedPipeServerStream("mypipe", PipeDirection.In))
    {
        server.WaitForConnection();
        await HandleClient(server);
    }
}
```

В методе HandleClient необходимо считывать команды от клиента и выполнять соответствующую команду.

5.2.2 Сервер запускать в отдельном Task при запуске приложения.

5.2.3 Создать новый проект типа консольное приложение со следующим кодом для тестирования работы именowanego канала.

```
var pipe = new System.IO.Pipes.NamedPipeClientStream(".",
"mypipe", System.IO.Pipes.PipeDirection.Out);
pipe.Connect();
var writer = new StreamWriter(pipe);
writer.WriteLine("Message(\"Hello\")");
writer.Flush();
writer.Close();
pipe.Close();
```

5.2.4 Протестировать работу именowanego канала.

5.3 Добавить в приложение интерфейс для работы с событиями – пользователь может планировать различные события на определенные даты и время, указывать их описание. Список событий отображается на странице.

5.4 Реализовать методы для автоматизации работы с событиями (CRUD) посредством скриптов.

5.5 Добавить возможность выполнять скрипты из заранее записанных текстовых файлов.

6 Порядок выполнения работы

6.1 Повторить теоретический материал п. 3.1;

6.2 Выполнить задания 5.1-5.5

6.3 Ответить на контрольные вопросы п. 8;

6.4 Заполнить отчет п. 7.

7 Содержание отчета

7.1 Титульный лист;

7.2 Цель работы;

7.3 Ответы на контрольные вопросы п. 6.3;

7.4 Вывод по проделанной работе.

8 Контрольные вопросы

8.1 Какие способы для взаимодействия между приложениями существуют