# Лабораторная работа №7 Разработка адаптируемого ПО

1. Цель работы
   1. Изучение процесса разработки ПО с поддержкой внешней интеграции при помощи скриптов.
2. Литература
   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
   1. Повторить теоретический материал (см. п.2).
   2. Изучить описание лабораторной работы.
4. Основное оборудование
   1. Персональный компьютер.
5. Задание
   1. Реализация поддержки скриптинга
      1. Создать новый проект оконного приложения
      2. Установить пакет Microsoft.CodeAnalysis.CSharp.Scripting
      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.
      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());

}

* + 1. scriptOptions определить приватным полем и инициализировать в конструкторе

scriptOptions = ScriptOptions.Default

.WithReferences(AppDomain.CurrentDomain.GetAssemblies())

.WithImports("System", "System.Collections.Generic", "System.Linq");

* + 1. Добавить в программу поле ввода для скриптов, и кнопку «Выполнить». Реализовать их функциональность.
    2. Протестировать работу скриптов.
  1. Поддержка внешних скриптов
     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.

* + 1. Сервер запускать в отдельном Task при запуске приложения.
    2. Создать новый проект типа консольное приложение со следующим кодом для тестирования работы именованного 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();

* + 1. Протестировать работу именованного канала.
  1. 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.
  2. Реализовать методы для автоматизации работы с событиями (CRUD) посредством скриптов.
  3. Добавить возможность выполнять скрипты из заранее записанных текстовых файлов.

1. Порядок выполнения работы
   1. Повторить теоретический материал п. 3.1;
   2. Выполнить задания 5.1-5.5
   3. Ответить на контрольные вопросы п. 8;
   4. Заполнить отчет п. 7.
2. Содержание отчета
   1. Титульный лист;
   2. Цель работы;
   3. Ответы на контрольные вопросы п. 6.3;
   4. Вывод по проделанной работе.
3. Контрольные вопросы
   1. Какие способы для взаимодействия между приложениями существуют