Оглавление

1. Описание задачи
2. Как решалась задача
3. Краткое описание модулей и их предназначение
   1. Как модули связываются друг с другом
   2. Форматы данных, протоколов
4. Подробные описания каждого модуля
   1. Задача, которая стоит перед модулем
   2. Из каких частей состоит модуль
   3. Нюансы реализации
5. Требования к системе, версии языка
6. Конфигурация внешнего окружения, зависимости
7. *Описание задачи*

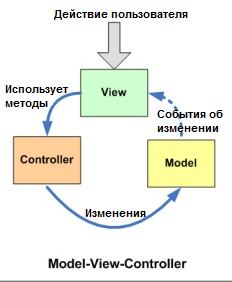
Программа должна по введённому пользователем логину на GitHub отображать список репозиториев данного аккаунта. При выборе определенного репозитория должен отображаться список коммитов в данном репозитории, а также информация о статусе файлов данного коммита. Также требуется возможность сохранить выбранный репозиторий.

1. *Как решалась задача*

Для решения поставленных задач был выбран паттерн MVC,так как программа писалась командой из 4 человек, паттерн MVC разделяет графический интерфейс, бизнес-логику и данные, что позволяет членам команды легко читать и вносить изменения в код. Использовалась сторонняя библиотека Octokit для работы с API GitHub.

Программная система была разделена на следующие модули:

* Controller
* Model
* Form1 (View)
* Сторонняя библиотека Octokit



1. *Краткое описание модулей и их предназначение*

Controller

Обеспечивает связь между пользователем и системой - контролирует ввод данных пользователем и использует модель и представление для реализации необходимой реакции.

Model

Модель предоставляет знания: данные и методы работы с этими данными, реагирует на запросы, изменяя своё состояние.

Form1 (View)

Отвечает за отображение информации (визуализацию).

Octokit

Сторонняя библиотека, которая используется для быстрой связи и обработки данных базы данных GitHub.

* 1. *Как модули связываются друг с другом*

Контроллер перехватывает событие извне и в соответствии с заложенной в него логикой, реагирует на это событие изменяя Mодель, посредством вызова соответствующего метода. После изменения Модель использует событие о том, что она изменилась, и все подписанные на это события Представления, получив его, обращаются к Модели за обновленными данными, после чего их и отображают.

* 1. *Форматы данных, протоколов*

Для хранения базы логинов пользователей используется текстовый файл .txt.

1. *Подробные описания каждого модуля*

Model

* 1. *Задача, которая стоит перед модулем*

Хранить список репозиториев в виде листа, содержать методы для проверки и добавления репозитория в список.

* 1. Из каких частей состоит модуль

Список адресов репозиториев

Методы их обработки

Controller

1. Задача, которая стоит перед модулем

Перехват событий извне, на событие вызывается соответствующий метод. Реализация методов – получение списка репозиториев, списка коммитов, информации о коммитах, сохранение репозитория, проверка интернет - соединения, удаление базы логинов пользователей.

1. Из каких частей состоит модуль

Методы:

* DeleteBase
* InternetIsConnected
* GetRepositoryCommits
* GetCommitedInfo
* Save
* GetRepositoriesByUser

1. Нюансы реализации

Использование библиотеки Octokit.

Form1 (View)

1. *Задача, которая стоит перед модулем*

Отображение данных, средства для работы пользователя с программой.

1. Из каких частей состоит модуль

* loginСomboBox
* repositoriesListBox
* commitsListBox
* commitedInfoListBox
* saveCurrentRepositoryButton
* обработчики событий
* методы GetUsername() и ClearAllListBoxes()

Сторонняя библиотека Octokit

* 1. Задача, которая стоит перед модулем

Работа с GitHub API

* 1. Из каких частей состоит модуль

Содержит методы для работы с API

1. *Требования к системе, версии языка*

Работа с программой осуществляется запуском файла repo checker.exe

Windows 10, 8.1, 8, 7

1. *Конфигурация внешнего окружения, зависимости*

Необходимо постоянное интернет – соединение для работы программы, есть ограничение на количество запросов к API.