МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное ГОСУДАРСТВЕННОЕ бюджетное ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»   
  
 Факультет информатики и вычислительной техники

Кафедра информационной безопасности

Отчёт

По лабораторной работе №1

“Система контроля версийна языке С#”

По дисциплине “Программирование на алгоритмических языках”

Выполнил: студент группы БИ-31

Анисимов В.В.

Проверил: доцент кафедры ИБ Сучков Д.С.

Йошкар-Ола

2019 г

**Цель работы:** Используя объектно-ориентированный подход, разработать упрощенную систему контроля версий на языке C#.

**Описание реализованных методов**

1. init [dir\_path] – инициализация СКВ для папки с файлами (или без), путь к которой указан в dir\_path**:**

public void Init(Dictionary dictionary)

{

\_dictionary.Add(dictionary);

}

public void Init(string path)

{

if (CheckDirrectoryLike(path))

{

\_dictionary.Add(new Dictionary(path));

}

}

1. status – отображение статуса отслеживаемых файлов последней проинициализированной папки (какие файлы отслеживаются, краткая информация по ним) Note: красным выделяется измененный файл, зеленым, соответственно, нет.

public void Status()

{

if (\_choosenDictionary == null)

{

Console.WriteLine("Сначала выберите дирректорию!");

return;

}

if (!\_choosenDictionary.Applyed)

{

Console.WriteLine($"\nДерриктория: {\_choosenDictionary.path}");

var dictionaryCopy = new Dictionary(\_choosenDictionary.path);

for (int i = 0; i < \_choosenDictionary.GitFiles.Count; i++)

{

var dictionaryOwner = \_choosenDictionary;

CheckForDeleted(ref dictionaryOwner, ref dictionaryCopy, i);

}

foreach (var file in \_choosenDictionary.GitFiles)

{

if(!file.Applyed)Console.WriteLine(

$"Название:{file.FileName},Размер {file.Size},Дата создания{file.Created},Дата модификации{file.Modified},Статус:{file.FileStatus}");

}

}

}

1. add [file\_path] – добавить файл под версионный контроль

public void Init(Dictionary dictionary)

{

\_dictionary.Add(dictionary);

}

1. remove [file\_path] – удалить файл из-под версионного контроля

public void Remove(string fullPath)

{

for (int i = 0; i < \_choosenDictionary.GitFiles.Count; i++)

{

if (\_choosenDictionary.GitFiles[i].FileName == fullPath)

{

\_choosenDictionary.GitFiles[i].FileStatus = FileStatus.Removed;

return;

}

}

Console.WriteLine("Файла не обнаружено");

}

1. apply [dir\_path] – сохранить все изменения в отслеживаемой папке (удалить все метки к файлам и сохранить изменения в них).

public void Apply(string dirPath)

{

foreach (var dictionary in \_dictionary)

{

if (dictionary.path == dirPath)

{

foreach (var file in dictionary.GitFiles)

{

if (file.FileStatus != FileStatus.New)

{

//file.FileStatus = FileStatus.Null;

file.Applyed = true;

}

}

Console.WriteLine("Данные успешно сохранены");

return;

}

}

Console.WriteLine("Файла не обнаружено");

}

1. checkout [dir\_path] OR [dir\_number] – перейти к указанной отслеживаемой директории.

private bool CheckDirrectoryLike(string path)

{

foreach (var dirrectory in \_dictionary)

{

if (dirrectory.path == path)

{

Console.WriteLine("Дерриктория уже находится под версионным контролем");

return false;

}

}

return true;

}

**Вывод:** в ходе работы была реализована простая система контроля версий на языке C#.