CSC 444

ASSIGNMENT 2

**Prepared by: Ardit Pajo**

**Student No: 999352169**

* **Specifications of Source Code Control System**

The purpose of this assignment is to build a source code control system. I decided to use the ruby programming language in developing this program. The reason behind this decision is that because most of what the source control does is handling files and directories and ruby has some built-in functions that makes such a program easy to implement. My program is called rvc.rb and is composed of a main class that is called RVC(revision control). Inside this class I have all the main commands of the system implemented and some helper functions that will be used extensively in modifying the files.

**The program will be able to perform the following functions:**

* Put a file under source code control 🡪 **ruby rvc.rb add <file1> < file2> …**
* Check in a modified version of a file allowing a user to associate a message with that check in and assign a version number to the checkin 🡪 **ruby rvc.rb commit “message …”**
* Checkout the most recent version of the file 🡪 **ruby rvc.rb checkout**
* Checkout the specified version number of a file 🡪 **ruby rvc.rb checkout version\_nr**
* List all the versions of the file and their messages 🡪 **ruby rvc.rb log**
* Check the status of the current files 🡪 **ruby rvc.rb status**
* Create a new branch/change branch 🡪 **ruby rvc.rb branch branch\_name**
* Revert a change in the current branch to a specified version number 🡪 **ruby rvc.rb revert version\_number**
* **Source Code Control System Design**

The program works in the following way: When the program is first called the constructor of the class checks if there is a directory called repository inside the directory from which the program is called. If the directory exist then the repository has already been created. If the directory doesn’t exist than this is the first time this program is being called from this folder and created a new directory called repository. Inside this directory a file called “\_currentBranch.txt” is created which will be initialized as follows “main 0” where the first name is the name of the current branch and the second is the version number of the commits. Another file called “\_branches.txt” will also be created which will contain all the names of the branches in the current repository. At first this file will contain only “main 0”. I decided that every branch will have its own directory. So when the repository is created another directory called main will be created. Inside main a log.txt file is initialized to empty which will store all the log messages in the current branch. Each new branch will contain its own log.txt file. Below I will describe the commands that are supported by the program and their design:

**The system commands and their implementations are:**

**1. ruby rvc.rb add <file1> <file2>** … : This command prepares all the specified file for commit. When this command is run by the user it first check if this files already exists inside the repository and if they do it prints a message informing the user. If this file do not exist in repository than it creates a new copy of this file inside the same folder by adding an ‘@’ before the name of each file.

**2. ruby rvc.rb commit “Message ….” :** This command performs 2 checks. First it check if there is any file that is ready for commit by checking for files that start with “@” in the current directory. The second check is if the files that are inside the repository have been modified since the last commit. After this check it copies this files over in the folder of the current branch that we are working on by increasing the version number and naming the files in this way “filename&version.type”. I have decided that for ease of implementation a new file is created every time a modification has been made to the file. After all new file are created a message with the “Message …” the version number and the file is written to the log.txt of the current branch.

**3. ruby rvc.rb log :** This command prints the contents of log.txt in the command line.

**4. ruby rvc.rb status :** This command goes through all the files in the current directory and checks if any of them is in the repository and has been modified, and prints and ‘M’ besides the name of the file. It also checks if any of the files has been added and is ready for commit. It prints an ‘A’ besides those files. All the other files that are not in repository or are not added for commit it prints a ‘?’.Meaning that this is a new file that doesn’t exist in the repository.

**5. ruby rvc.rb revert version\_number:** This command removes all the files that have a version number greater than the one specified in the command. It updates the version number in the “\_branches.txt” and in “\_currentBranch.txt”. Removes all the logs that have a version number greater than the one specified.

**6. ruby rvc.rb branch branch\_name :** This command checks if the branch\_name already exists in “\_branches.txt” and if it does than it just changes the current branch name in the “\_currentBranch.txt” and if the version number is smaller than the one of the previous branch it will ask the user if he wants to update the current branch. If the branch doesn’t exist it creates a new directory adds the branch name in the “\_branches.txt”, updates the “\_currentBranch.txt” and copies all files from current branch to the new branch.

**7. ruby rvc.rb checkout filename version\_number :** This command goes through all the files with the specified filename in repository and checks out the file with the specified version number if it exists in the repository.

* **Tests**

The current files in the directory are:

**canvasDraw.js, file\_search.rb, counting\_sort.rb and rvc.**

|  |  |  |  |
| --- | --- | --- | --- |
| Nr | Input | Expected | Got |
| 1 | ruby rvc.rb add canvasDraw.js | >File canvasDraw.js was added successfully. Ready for commit!!! | C:\Users\ardit\Desktop\RVC\tests\test1_a.jpg |
| 2 | ruby rvc.rb add canvasDraw.c | >The file canvasDraw.c doesn’t exist int the current folder! | C:\Users\ardit\Desktop\RVC\tests\test1_b.jpg |
| 3 | ruby rvc.rb add canvasDraw.js | >File canvasDraw.js has already been added for commit !!! | C:\Users\ardit\Desktop\RVC\tests\test1_c.jpg |
| 4 | ruby rvc.rb add rvc.rb | >Can’t add the programs file | C:\Users\ardit\Desktop\RVC\tests\test1_d.jpg |
| 5 | ruby rvc.rb commit “My first commit” | >The commit was successful version 1 | C:\Users\ardit\Desktop\RVC\tests\test2_a.jpg |
| 6 | ruby rvc.rb commit “My first commit” | >Nothing to commit!!! | C:\Users\ardit\Desktop\RVC\tests\test2_b.jpg |
| 7 | ruby rvc.rb add canvasDraw.js | File canvasDraw.js already exists in the repository | C:\Users\ardit\Desktop\RVC\tests\test2_c.jpg |
| 8 | >ruby rvc.rb status  >ruby rvc.rb add file\_search.rb  >ruby rvc.rb status | >file\_search.rb ---?  counting\_sort.rb --?  >File file\_search.rb was added successfully.Ready for commit!!!  > file\_search.rb ---?  counting\_sort.rb --A | C:\Users\ardit\Desktop\RVC\tests\test3_a.jpg |
| 9 | >ruby rvc.rb commit “Second commit”  > ruby rvc.rb log | >The commit was succesfull version 2  > will print the contents of log file | C:\Users\ardit\Desktop\RVC\tests\test3_b.jpg |
| 10 | >ruby rvc.rb add counting\_sort.rb file\_search.rb  >ruby rvc.rb commit “Third commit”  >ruby rvc.rb log | > File counting\_sort.rb was added successfully  >File file\_search.rb was added successfully  >The commit was successful version 3  > will print the log file | C:\Users\ardit\Desktop\RVC\tests\test3_c.jpg |
| 11 | I have modified the canvasDraw.js  >ruby rvc.rb status  >ruby rvc.rb commit “Second commit”  >ruby rvc.rb log | > canvasDraw.js --M  >The commit was successful  > will print the log file | C:\Users\ardit\Desktop\RVC\tests\test3_d.jpg |
| 12 | >ruby rvc.rb branch ardit | > Create branch ardit | C:\Users\ardit\Desktop\RVC\tests\test3_f.jpg |
| 13 | >ruby rvc.rb branch second | > Create branch second | C:\Users\ardit\Desktop\RVC\tests\test3_g.jpg |
| 14 | >ruby rvc.rb branch ardit  > ruby rvc.rb log | > Current branch changed to ardit  > will print the log file | C:\Users\ardit\Desktop\RVC\tests\test4_a.jpg |
| 15 | Modify the file\_search.rb  >ruby rvc.rb status  >ruby rvc.rb commit “Changed the file\_search.rb”  >ruby rvc.rb log | > file\_search – M  > The commit was successful version 5  > will print the log file | C:\Users\ardit\Desktop\RVC\tests\test4_b.jpg |
| 16 | >ruby rvc.rb branch main  >y  > ruby rvc.rb log | > There were some modifications in previous branch do you want to apply this to main y/n?  > the updated log files at main | C:\Users\ardit\Desktop\RVC\tests\test4_c.jpg |
| 17 | >ruby rvc.rb checkout file\_search .rb 5  >ruby rvc.rb checkout file\_search .rb 2  >ruby rvc.rb checkout file\_search .rb 3 | > Checking out file\_search.rb version 5  > Checking out file\_search.rb version 2  > File file\_search.rb with version 3 not found in repository | C:\Users\ardit\Desktop\RVC\tests\test4_d.jpg |
| 18 | >ruby rvc.rb  revert 3  >ruby rvc.b log | > will remove all the files with version greater than 3  > will print the updated log file | C:\Users\ardit\Desktop\RVC\tests\test4_f.jpg |