

Университет ИТМО
Факультет программной инженерии и компьютерной техники

Методы и средства программной
инженерии.
Лабораторная работа №2.
Системы контроля версий

Смирнов Виктор Игоревич	P32131
Шиняков Артём Дмитриевич	R32372
Вариант	1009

Содержание

1	Задание	1
2	Взаимодействие с репозиторием через SVN	1
3	Взаимодействие с репозиторием через GIT	8
4	Вывод	15

1 Задание

Сконфигурировать в своём домашнем каталоге репозитории svn и git и загрузить в них начальную ревизию файлов с исходными кодами (в соответствии с выданным вариантом).

Воспроизвести последовательность команд для систем контроля версий svn и git, осуществляющих операции над исходным кодом, приведённые на блок-схеме.

При составлении последовательности команд необходимо учитывать следующие условия:

1. Цвет элементов схемы указывает на пользователя, совершившего действие (красный - первый, синий - второй).
2. Цифры над узлами - номер ревизии. Ревизии создаются последовательно.
3. Необходимо разрешать конфликты между версиями, если они возникают.

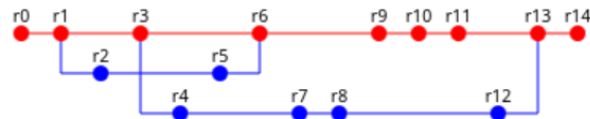


Рис. 1: История ревизий репозитория

2 Взаимодействие с репозиторием через SVN

```
1 #!/bin/bash
2
3 . ci/svn/lib/head.sh --source-only
4 TARGET="main"
5 . ci/svn/lib/dsl.sh --source-only
6
7 begin
8   log "pwd: $(pwd)"
9
10  call clean
11  call r_init
12  call r0
13  call r1
14  call r2
15  call r3
16  call r4
17  call r5
18  call r6
19  call r7
20  call r8
21  call r9
22  call r10
23  call r11
24  call r12
25  call r13
26  call r14
27
28  svn checkout file://$HOME/.svnrepos/$REPO_NAME out
29  cd out
```

```

30  svn log > $VSC_NAME-log.txt
31  cd ..
32  end

1  #!/bin/bash
2
3  . ci/svn/lib/head.sh --source-only
4  TARGET="init"
5  . ci/svn/lib/dsl.sh --source-only
6
7  begin
8    log "repo is $REPO_NAME"
9
10   mkdir -p ~/.svnrepos/
11   svnadmin create ~/.svnrepos/$REPO_NAME
12
13   log "repository created"
14
15   svn mkdir -m "Create repository structure." \
16     file://$HOME/.svnrepos/$REPO_NAME/trunk \
17     file://$HOME/.svnrepos/$REPO_NAME/branches \
18     file://$HOME/.svnrepos/$REPO_NAME/tags
19
20   log "repository initialized"
21
22   mkdir -p playground/$REPO_NAME
23   cd playground/$REPO_NAME
24
25   svn checkout file://$HOME/.svnrepos/$REPO_NAME/trunk trunk
26   cd trunk
27
28   svn add --force .
29   svn commit -m "Initial import."
30   svn update
31  end

```

```

1  #!/bin/bash
2
3  . ci/svn/lib/head.sh --source-only
4  BRANCH="trunk"
5  COMMIT="commit0"
6  TARGET="$BRANCH:$COMMIT"
7  . ci/svn/lib/dsl.sh --source-only
8
9  begin
10   enter
11
12   edit A.java
13   edit B.java
14   edit E.java
15   edit F.java
16
17   svn add *
18   svn commit -m "$TAG added A, B, E, F"
19  end

```

```

1  #!/bin/bash
2
3  . ci/svn/lib/head.sh --source-only
4  BRANCH="trunk"
5  COMMIT="commit1"
6  TARGET="$BRANCH:$COMMIT"
7  . ci/svn/lib/dsl.sh --source-only
8
9  begin
10   enter
11
12   edit A.java
13   edit B.java
14   edit E.java
15   edit F.java
16
17   svn commit -m "$TAG edited A, B, E, F"

```

```

18 end

1 #!/bin/bash
2
3 . ci/svn/lib/head.sh --source-only
4 BRANCH="feature-1"
5 COMMIT="commit2"
6 TARGET="$BRANCH:$COMMIT"
7 . ci/svn/lib/dsl.sh --source-only
8
9 begin
10   branch trunk "$BRANCH" "Creating a branch for a feature #1"
11
12   enter
13
14   edit A.java
15   edit B.java
16   edit E.java
17   edit F.java
18
19   svn commit -m "$TAG edited A, B, E, F"
20   log "committed changes to $BRANCH"
21 end

1 #!/bin/bash
2
3 . ci/svn/lib/head.sh --source-only
4 BRANCH="trunk"
5 COMMIT="commit3"
6 TARGET="$BRANCH:$COMMIT"
7 . ci/svn/lib/dsl.sh --source-only
8
9 begin
10   enter
11
12   edit "*"
13   edit "3yNy8wQeGi.Xzj"
14   edit A.java
15   edit B.java
16   edit E.java
17   edit F.java
18
19   svn add "*"
20   svn add "3yNy8wQeGi.Xzj"
21
22   svn commit -m "$TAG edited A, B, E, F, added *, 3yNy8wQeGi.Xzj"
23   log "committed changes to $BRANCH"
24 end

1 #!/bin/bash
2
3 . ci/svn/lib/head.sh --source-only
4 BRANCH="feature-2"
5 COMMIT="commit4"
6 TARGET="$BRANCH:$COMMIT"
7 . ci/svn/lib/dsl.sh --source-only
8
9 begin
10   branch "trunk" $BRANCH "Creating a branch for a feature #2"
11
12   enter
13
14   delete "*"
15   delete "3yNy8wQeGi.Xzj"
16   edit A.java
17   edit B.java
18   edit E.java
19   edit F.java
20
21   svn delete "*"
22   svn delete "3yNy8wQeGi.Xzj"
23
24   svn commit -m "$TAG edited A, B, E, F, removed *, 3yNy8wQeGi.Xzj"

```

```
25 log "committed changes to $BRANCH"
26 end
```

```
1 #!/bin/bash
2
3 . ci/svn/lib/head.sh --source-only
4 BRANCH="feature-1"
5 COMMIT="commit5"
6 TARGET="$BRANCH:$COMMIT"
7 . ci/svn/lib/dsl.sh --source-only
8
9 begin
10 enter
11
12 edit A.java
13 edit B.java
14 edit E.java
15 edit F.java
16
17 svn commit -m "$TAG edited A, B, E, F"
18 log "committed changes to $BRANCH"
19 end
```

```
1 #!/bin/bash
2
3 . ci/svn/lib/head.sh --source-only
4 BRANCH="trunk"
5 COMMIT="commit6"
6 TARGET="$BRANCH:$COMMIT"
7 . ci/svn/lib/dsl.sh --source-only
8
9 begin
10 enter
11
12 svn merge ^/branches/feature-1
13 log "merged feature-1 into trunk"
14
15 delete "3yNy8wQeGi.Xzj"
16 edit A.java
17 edit B.java
18 edit E.java
19 edit F.java
20
21 svn remove "3yNy8wQeGi.Xzj"
22
23 svn commit -m "merged feature-1, $TAG edited A, B, E, F, removed 3yNy8wQeGi.Xzj"
24 log "committed changes to $BRANCH"
25 end
```

```
1 #!/bin/bash
2
3 . ci/svn/lib/head.sh --source-only
4 BRANCH="feature-2"
5 COMMIT="commit7"
6 TARGET="$BRANCH:$COMMIT"
7 . ci/svn/lib/dsl.sh --source-only
8
9 begin
10 enter
11
12 edit A.java
13 edit B.java
14 edit E.java
15 edit F.java
16
17 svn commit -m "$TAG edited A, B, E, F"
18 log "committed changes to $BRANCH"
19 end
```

```
1 #!/bin/bash
2
3 . ci/svn/lib/head.sh --source-only
4 BRANCH="feature-2"
```

```

5 COMMIT="commit8"
6 TARGET="$BRANCH:$COMMIT"
7 . ci/svn/lib/dsl.sh --source-only
8
9 begin
10   enter
11
12   edit "*"
13   edit "3yNy8wQeGi.Xzj"
14   edit A.java
15   edit B.java
16   edit E.java
17   edit F.java
18
19   svn add "*"
20   svn add "3yNy8wQeGi.Xzj"
21
22   svn commit -m "$TAG edited A, B, E, F, added *, 3yNy8wQeGi.Xzj"
23 end

```

```

1 #!/bin/bash
2
3 . ci/svn/lib/head.sh --source-only
4 BRANCH="trunk"
5 COMMIT="commit9"
6 TARGET="$BRANCH:$COMMIT"
7 . ci/svn/lib/dsl.sh --source-only
8
9 begin
10   enter
11
12   delete "*"
13   edit A.java
14   edit B.java
15   edit E.java
16   edit F.java
17
18   svn remove "*"
19
20   svn commit -m "$TAG edited A, B, E, F, removed *"
21   log "committed changes to $BRANCH"
22 end

```

```

1 #!/bin/bash
2
3 . ci/svn/lib/head.sh --source-only
4 BRANCH="trunk"
5 COMMIT="commit10"
6 TARGET="$BRANCH:$COMMIT"
7 . ci/svn/lib/dsl.sh --source-only
8
9 begin
10   enter
11
12   edit "*"
13   edit "67VNlROFbP.TcV"
14   edit A.java
15   edit B.java
16   edit E.java
17   edit F.java
18
19   svn add "*"
20   svn add "67VNlROFbP.TcV"
21
22   svn commit -m "$TAG edited A, B, E, F, restored *, added 67VNlROFbP.TcV"
23   log "committed changes to $BRANCH"
24 end

```

```

1 #!/bin/bash
2
3 . ci/svn/lib/head.sh --source-only
4 BRANCH="trunk"
5 COMMIT="commit11"

```

```

6 TARGET="$BRANCH:$COMMIT"
7 . ci/svn/lib/dsl.sh --source-only
8
9 begin
10   enter
11
12   delete "67VNlROFbP.TcV"
13   edit   "*"
14   edit   A.java
15   edit   B.java
16   edit   E.java
17   edit   F.java
18
19   svn remove "67VNlROFbP.TcV"
20
21   svn commit -m "$TAG edited A, B, E, F, *, removed 67VNlROFbP.TcV"
22   log "committed changes to $BRANCH"
23 end

```

```

1 #!/bin/bash
2
3 . ci/svn/lib/head.sh --source-only
4 BRANCH="feature-2"
5 COMMIT="commit12"
6 TARGET="$BRANCH:$COMMIT"
7 . ci/svn/lib/dsl.sh --source-only
8
9 begin
10   enter
11
12   delete "*"
13   delete "3yNy8wQeGi.Xzj"
14   edit A.java
15   edit B.java
16   edit E.java
17   edit F.java
18
19   svn remove "*"
20   svn remove "3yNy8wQeGi.Xzj"
21
22   svn commit -m "$TAG edited A, B, E, F, removed *, 3yNy8wQeGi.Xzj"
23   log "committed changes to $BRANCH"
24 end

```

```

1 #!/bin/bash
2
3 . ci/svn/lib/head.sh --source-only
4 BRANCH="trunk"
5 COMMIT="commit13"
6 TARGET="$BRANCH:$COMMIT"
7 . ci/svn/lib/dsl.sh --source-only
8
9 begin
10   enter
11
12   svn merge ~/branches/feature-2
13   log "merged feature-2 into trunk"
14
15   edit A.java
16   edit B.java
17   edit E.java
18   edit F.java
19
20   svn commit -m "$TAG edited A, B, E, F, merged feature-2 into $BRANCH"
21   log "committed changes to $BRANCH"
22 end

```

```

1 #!/bin/bash
2
3 . ci/svn/lib/head.sh --source-only
4 BRANCH="trunk"
5 COMMIT="commit14"
6 TARGET="$BRANCH:$COMMIT"

```

```

7 . ci/svn/lib/dsl.sh --source-only
8
9 begin
10   enter
11
12   edit "rvvddKJVqH.1iP"
13   edit A.java
14   edit B.java
15   edit E.java
16   edit F.java
17
18   svn add "rvvddKJVqH.1iP"
19
20   svn commit -m "$TAG edited A, B, E, F, added rvvddKJVqH.1iP"
21   log "committed changes to $BRANCH"
22 end

```

```

1 set -e
2
3 cd $(dirname -- "$0"; )
4 cd ../..
5
6 VSC_NAME="svn"
7 REPO_NAME="semt-assignment-vcs-$VSC_NAME-repository"
8 SCRIPT="ci/svn"

```

```

1 TAG="[$VSC_NAME:$TARGET]"
2
3 log() {
4   echo "$TAG $1"
5 }
6
7 remove() {
8   rm -rf $1
9   log "removed $1"
10 }
11
12 copy() {
13   cp "$1" "$2"
14   log "copied $1 to $2"
15 }
16
17 call() {
18   bash "$SCRIPT/$1.sh"
19 }
20
21 begin() {
22   log "started $TARGET"
23 }
24
25 end() {
26   log "finished $TARGET"
27 }
28
29 enter() {
30   SRC=../../..../history/$COMMIT
31   cd playground/$REPO_NAME/$BRANCH
32
33   USERNAME="blue"
34   if [[ $BRANCH = "trunk" ]]; then
35     USERNAME="red"
36   fi
37   svn update --username $USERNAME
38 }
39
40 edit() {
41   copy "$SRC/$1" "$1"
42 }
43
44 delete() {
45   remove $1
46 }
47

```



```

48 branch() {
49     # $1 - source branch name, e.g. "trunk" or "branches/my-branch"
50     # $2 - target branch name, e.g. "my-branch"
51     # $3 - message
52
53     svn copy \
54         file://$HOME/.svnrepos/$REPO_NAME/$1 \
55         file://$HOME/.svnrepos/$REPO_NAME/branches/$2 \
56         -m "$3"
57     log $3
58
59     cd playground/$REPO_NAME
60     svn checkout \
61         file://$HOME/.svnrepos/$REPO_NAME/branches/$2 $2
62     log "checkout to branch $2"
63     cd ../../
64 }

```

3 Взаимодействие с репозиторием через GIT

```

1 #!/bin/bash
2
3 . ci/git-r/lib/head.sh --source-only
4 TARGET="main"
5 . ci/git-r/lib/dsl.sh --source-only
6
7 begin
8     call clean
9     call init
10    call init_users
11    call r0
12    call r1
13    call r2
14    call r3
15    call r4
16    call r5
17    call r6
18    call r7
19    call r8
20    call r9
21    call r10
22    call r11
23    call r12
24    call r13
25    call r14
26
27    cd ~/.gitrepo/$REPO_NAME
28    mkdir logs
29    cd logs
30    git log > $VSC_NAME-log.txt
31    git log --pretty=format:@"%h %s" --graph > $VSC_NAME-graph.txt
32 end

```

```

1 #!/bin/bash
2
3 . ci/git-r/lib/head.sh --source-only
4 TARGET="init"
5 . ci/git-r/lib/dsl.sh --source-only
6
7 begin
8     log "repo is $REPO_NAME"
9
10    mkdir -p ~/.gitrepo
11    cd ~/.gitrepo
12    git init $REPO_NAME --bare
13
14    git config --global pull.rebase false
15    log "repository created"
16
17 end

```

```

1 #!/bin/bash

```

```

2
3 . ci/git-r/lib/head.sh --source-only
4 BRANCH="master"
5 COMMIT="commit0"
6 TARGET="$BRANCH:$COMMIT"
7 NAME=$ARTEM
8 EMAIL="Artem@itmo.ru"
9 . ci/git-r/lib/dsl.sh --source-only
10
11 begin
12   enter
13
14   set_name $NAME
15   set_email $EMAIL
16
17   cp $HISTORY_PATH/$COMMIT/* .
18   add_all
19
20   comm "Start of project. Added initial files."
21
22   git push origin
23 end

```

```

1 #!/bin/bash
2
3 . ci/git-r/lib/head.sh --source-only
4 BRANCH="master"
5 COMMIT="commit1"
6 TARGET="$BRANCH:$COMMIT"
7 NAME=$ARTEM
8 EMAIL="Artem@itmo.ru"
9 . ci/git-r/lib/dsl.sh --source-only
10
11 begin
12   enter
13
14   git pull origin
15
16   cp $HISTORY_PATH/$COMMIT/* .
17   add_all
18
19   comm "Added: bb - print class name in F.java."
20
21   git push origin
22 end

```

```

1 #!/bin/bash
2
3 . ci/git-r/lib/head.sh --source-only
4 BRANCH="second_branch"
5 COMMIT="commit2"
6 TARGET="$BRANCH:$COMMIT"
7 NAME=$VITYA
8 EMAIL="Vitya@itmo.ru"
9 . ci/git-r/lib/dsl.sh --source-only
10
11 begin
12   enter
13
14   set_name $NAME
15   set_email $EMAIL
16
17   git pull origin
18
19   git checkout -b second_branch
20
21   cp $HISTORY_PATH/$COMMIT/* .
22   add_all
23
24   comm "Created second branch, files are in the same state as in r0 commit."
25
26   git push origin second_branch
27 end

```

```

1 #!/bin/bash
2
3 . ci/git-r/lib/head.sh --source-only
4 BRANCH="master"
5 COMMIT="commit3"
6 TARGET="$BRANCH:$COMMIT"
7 NAME=$ARTEM
8 EMAIL="Artem@itmo.ru"
9 . ci/git-r/lib/dsl.sh --source-only
10
11 begin
12   enter
13
14   git pull origin
15
16   cp $HISTORY_PATH/$COMMIT/* .
17   add_all
18
19   comm "Added: pp fuction - returns Object in F class, * file - contains chinese, 3
        yNy8wQeGi.Xzj file - contains binary something "
20
21   git push origin
22 end

```

```

1 #!/bin/bash
2
3 . ci/git-r/lib/head.sh --source-only
4 BRANCH="third_branch"
5 COMMIT="commit4"
6 TARGET="$BRANCH:$COMMIT"
7 NAME=$VITYA
8 EMAIL="Vitya@itmo.ru"
9 . ci/git-r/lib/dsl.sh --source-only
10
11 begin
12   enter
13
14   git checkout master
15
16   git pull origin
17
18   git checkout -b third_branch
19
20   cp $HISTORY_PATH/$COMMIT/* .
21
22   git rm "*" -f
23   git rm 3yNy8wQeGi.Xzj -f
24
25   add_all
26
27   comm "Created third branch, files are in the same state as in r0 commit., files * and
        3yNy8wQeGi.Xzj were removed"
28
29   git push origin third_branch
30 end

```

```

1 #!/bin/bash
2
3 . ci/git-r/lib/head.sh --source-only
4 BRANCH="second_branch"
5 COMMIT="commit5"
6 TARGET="$BRANCH:$COMMIT"
7 NAME=$VITYA
8 EMAIL="Vitya@itmo.ru"
9 . ci/git-r/lib/dsl.sh --source-only
10
11 begin
12   enter
13
14   git pull origin master
15
16   git checkout second_branch
17

```

```

18 cp $HISTORY_PATH/$COMMIT/* .
19 add_all
20
21 comm "Added: bb - print class name in F.java."
22
23 git push origin second_branch
24 end

1 #!/bin/bash
2
3 . ci/git-r/lib/head.sh --source-only
4 BRANCH="master"
5 COMMIT="commit6"
6 TARGET="$BRANCH:$COMMIT"
7 NAME=$ARTEM
8 EMAIL="Artem@itmo.ru"
9 . ci/git-r/lib/dsl.sh --source-only
10
11 begin
12   enter
13
14   git pull origin
15
16   git checkout second_branch
17
18   git checkout third_branch
19
20   git checkout master
21
22   git merge second_branch -m "Second branch does not contain any new features"
23   cp $HISTORY_PATH/$COMMIT/* .
24   add_all
25
26   git rm 3yNy8wQeGi.Xzj -f
27
28   comm "Added: pp fuction - returns Object in F class, * file - contains chinese, 3
        yNy8wQeGi.Xzj file - contains binary something, interfaces A,B,E turned into classes
        "
29
30   git push origin
31 end

1 #!/bin/bash
2
3 . ci/git-r/lib/head.sh --source-only
4 BRANCH="third-branch"
5 COMMIT="commit7"
6 TARGET="$BRANCH:$COMMIT"
7 NAME=$VITYA
8 EMAIL="Vitya@itmo.ru"
9 . ci/git-r/lib/dsl.sh --source-only
10
11 begin
12   enter
13
14   git pull origin master
15
16   git checkout third_branch
17
18   cp $HISTORY_PATH/$COMMIT/* .
19   add_all
20
21   comm "Added: bb function - returns Object in F.java"
22
23   git push origin third_branch
24 end

1 #!/bin/bash
2
3 . ci/git-r/lib/head.sh --source-only
4 BRANCH="third-branch"
5 COMMIT="commit8"
6 TARGET="$BRANCH:$COMMIT"

```

```

7 NAME=$VITYA
8 EMAIL="Vitya@itmo.ru"
9 . ci/git-r/lib/dsl.sh --source-only
10
11 begin
12   enter
13
14   git pull origin third_branch
15
16   cp $HISTORY_PATH/$COMMIT/* .
17   add_all
18
19   comm "Added: pp fuction - returns Object in F class, * file - contains chinese, 3
        yNy8wQeGi.Xzj file - contains binary something"
20
21   git push origin third_branch
22 end

```

```

1 #!/bin/bash
2
3 . ci/git-r/lib/head.sh --source-only
4 BRANCH="master"
5 COMMIT="commit9"
6 TARGET="$BRANCH:$COMMIT"
7 NAME=$ARTEM
8 EMAIL="Artem@itmo.ru"
9 . ci/git-r/lib/dsl.sh --source-only
10
11 begin
12   enter
13
14   git pull origin
15
16   git checkout master
17
18   cp $HISTORY_PATH/$COMMIT/* .
19   add_all
20
21   git rm "*" -f
22
23   comm "Added: nn fuction - returns Object in F class, * file - was removed, classes A,
        B, E got one new method each "
24
25   git push origin
26 end

```

```

1 #!/bin/bash
2
3 . ci/git-r/lib/head.sh --source-only
4 BRANCH="master"
5 COMMIT="commit10"
6 TARGET="$BRANCH:$COMMIT"
7 NAME="Artem"
8 EMAIL="Artem@itmo.ru"
9 . ci/git-r/lib/dsl.sh --source-only
10
11 begin
12   enter
13
14   git pull origin
15
16   cp $HISTORY_PATH/$COMMIT/* .
17   add_all
18
19   comm "Added: A, B, E, F classes got one new method each, * and 67VNIROFbP.TcV files
        were created."
20
21   git push origin
22 end

```

```

1 #!/bin/bash
2
3 . ci/git-r/lib/head.sh --source-only

```

```

4 BRANCH="master"
5 COMMIT="commit11"
6 TARGET="$BRANCH:$COMMIT"
7 NAME=$ARTEM
8 EMAIL="Artem@itmo.ru"
9 . ci/git-r/lib/dsl.sh --source-only
10
11 begin
12     enter
13
14     git pull origin
15
16     cp $HISTORY_PATH/$COMMIT/* .
17     add_all
18
19     git rm 67VNlROFbP.TcV -f
20
21     comm "Added: A, B, E, F classes got one new method each, 67VNlROFbP.TcV file - removed"
22     .
23     git push origin
24 end

```

```

1 #!/bin/bash
2
3 . ci/git-r/lib/head.sh --source-only
4 BRANCH="third-branch"
5 COMMIT="commit12"
6 TARGET="$BRANCH:$COMMIT"
7 NAME=$VITYA
8 EMAIL="Vitya@itmo.ru"
9 . ci/git-r/lib/dsl.sh --source-only
10
11 begin
12     enter
13
14     git checkout third_branch
15
16     cp $HISTORY_PATH/$COMMIT/* .
17     add_all
18
19     git rm "*" -f
20     git rm 3yNy8wQeGi.Xzj -f
21
22     comm "Added: mm function - returns Object in F.java, * and 3yNy8wQeGi.Xzj files were removed."
23
24     git push origin third_branch
25 end

```

```

1 #!/bin/bash
2
3 . ci/git-r/lib/head.sh --source-only
4 BRANCH="master"
5 COMMIT="commit13"
6 TARGET="$BRANCH:$COMMIT"
7 NAME=$ARTEM
8 EMAIL="Artem@itmo.ru"
9 . ci/git-r/lib/dsl.sh --source-only
10
11 begin
12     enter
13
14     git pull origin
15
16     git checkout third_branch
17
18     git checkout master
19
20     {
21         git merge third_branch
22     }||{
23         git checkout second_branch F.java

```

```

24     git rm "*" -f
25 }
26
27 cp $HISTORY_PATH/$COMMIT/* .
28
29 add_all
30
31 comm "Merged master and third branch. Added new functions in B, E, F"
32
33 git push origin
34
35 end

1  #!/bin/bash
2
3  . ci/git-r/lib/head.sh --source-only
4  BRANCH="master"
5  COMMIT="commit14"
6  TARGET="$BRANCH:$COMMIT"
7  NAME=$ARTEM
8  EMAIL="Artem@itmo.ru"
9  . ci/git-r/lib/dsl.sh --source-only
10
11 begin
12     enter
13
14     git pull origin
15
16     cp $HISTORY_PATH/$COMMIT/* .
17     add_all
18
19     comm "Added: A, B, E, F classes got one new method each, rvvddKJVqH.1ip file - added."
20
21     git push origin
22 end

1  set -e
2
3  cd $(dirname -- "$0"; )
4  cd ../../
5  HISTORY_PATH=$(pwd)/history
6  USERS_REPO=~/.user_repo
7  ARTEM="Artem"
8  VITYA="Vitya"
9
10 VSC_NAME="git-r"
11 REPO_NAME="semt-assignment-vcs-$VSC_NAME-repository"
12 SCRIPT="ci/git-r"
13 URL=git+ssh://s337054@se.ifmo.ru:2222/home/studs/s337054/srv/git/project.git

1  TAG="[$VSC_NAME:$TARGET]"
2
3  log() {
4      echo "$TAG $1"
5  }
6
7  remove() {
8      rm -rf $1
9      log "removed $1"
10 }
11
12 add_all(){
13     git add .
14     log "added files from $TAG"
15 }
16
17 call() {
18     bash "$SCRIPT/$1.sh"
19 }
20
21 begin() {
22     log "started $TARGET"

```

```

23 }
24
25 end() {
26     log "finished $TARGET"
27 }
28
29 set_name() {
30     git config --local user.name "$1"
31 }
32
33 set_email() {
34     git config --local user.email "$2"
35 }
36
37 enter() {
38     cd $USERS_REPO/$NAME
39 }
40
41 comm() {
42     git commit -m "$1"
43 }

```

4 Вывод

Выполнив данную лабораторную работу мы научились использовать базовые функции таких известных систем контроля версий как git и svn. Оба пакета программного обеспечения предоставляют весь необходимый функционал для удобного использования, но управляются пользователем по-разному, что делает одну СКВ предпочтительнее другой в зависимости от сложившейся ситуации. Например, интерфейс системы контроля версий git немного проще svn и предлагает более лаконичную (не факт) схему работы с репозиторием. А svn в свою очередь может предложить очень удобный механизм частичного монтажа поддиректорий репозитория, такая функциональность может быть полезна, когда мы имеем дело с монорепозиториями и не хотим видеть сразу все содержимое большого проекта.