

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
“Национальный исследовательский университет ИТМО”  
ФКТиУ, Кафедра Информатики и вычислительной техники

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2  
“Методы и средства программной инженерии”  
Вариант № 40 000

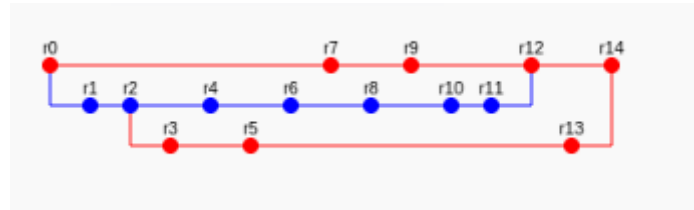
Выполнили:  
Студенты группы Р3232  
Куценко А. В.  
Федоров Н. С.

Преподаватель:  
Покид А. В.

Санкт-Петербург  
2021 г.

# Задание

Вариант 40000



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:

Цвет элементов схемы указывает на пользователя, совершившего действие (красный - первый, синий - второй).

Цифры над узлами - номер ревизии. Ревизии создаются последовательно.

Необходимо разрешать конфликты между версиями, если они возникают.

Отчёт по работе должен содержать:

Задание и блок-схему в соответствии с вариантом.

Список команд, использованных при создании и конфигурации репозитория в домашнем каталоге пользователя.

Номера ревизий и соответствующие им последовательности команд с комментариями (для svn и git).

Выводы по работе.

## Git

```
git clone https://github.com/alex-grandson/mispi-lab2-git.git
```

r0:

```
git commit --author="red <>" -m "init r0"
```

r1:

```
git checkout -b "blue-branch"
```

Switched to a new branch 'blue-branch'

```
git add *
```

```
git commit --author="blue <>" -m "r1"
```

r2:

```
git add *
```

```
git commit --author="blue <>" -m "r2"
```

r3:

```
git checkout -b "red-branch"
```

```
git add *
```

```
git commit --author="red <>" -m "r3"
```

r4:

```
git checkout blue-branch
```

```
git add *
```

```
git commit --author="blue <>" -m "r4"
```

r5:

```
git checkout red-branch
```

```
git add *
```

```
git commit --author="red <>" -m "r5"
```

r6:

```
git checkout blue-branch
```

```
git add *
```

```
git commit --author="blue <>" -m "r6"
```

r7:

```
git checkout master
```

```
git add *
```

```
git commit --author="red <>" -m "r7"
```

r8:

```
git checkout blue-branch
```

```
git add *
```

```
git commit --author="blue <>" -m "r8"
```

r9:

```
git checkout master
```

```
git add *
```

```
git commit --author="red <>" -m "r9"
```

r10:

```
git checkout blue-branch
add *
commit --author="blue <>" -m "r10"
```

```
r11:
add *
commit --author="blue <>" -m "r11"
```

```
r12:
git add *
git commit --author="blue <>" -m "r12"
git merge blue-branch
Auto-merging G.java
CONFLICT (content): Merge conflict in G.java
Auto-merging D.java
CONFLICT (content): Merge conflict in D.java
Auto-merging C.java
CONFLICT (content): Merge conflict in C.java
Automatic merge failed; fix conflicts and then commit the result.
```

```
nano C.java
git add C.java
nano D.java
git add D.java
nano G.java
git add G.java
git commit --author="red <>" -m "r12 merge"
```

```
r13:
git checkout red-branch
git add *
git commit --author="red <>" -m "r13"
```

```
r14:
git merge --no-commit red-branch
warning: Cannot merge binary files: * (HEAD vs. red-branch)
CONFLICT (add/add): Merge conflict in *
Auto-merging *
Automatic merge failed; fix conflicts and then commit the result.
```

```
git add "*"
git commit --author="red <>" -m "merge binary"
```

```
git add *
```

```
git commit --author="red <>" -m "r14"
```

## SVN

```
svnadmin create lab
```

```
cd lab
```

```
svn mkdir -m "lab struct" file:///home/s285484/misp_svn/lab/trunk
```

```
svn mkdir -m "lab struct" file:///home/s285484/misp_svn/lab/branches
```

```
mkdir develop
```

```
svn checkout file:///home/s285484/misp_svn/lab/trunk/ develop
```

```
cd develop
```

```
r0:
```

```
cp ~/commits/commit0/* .
```

```
svn add *
```

```
A    C.java
```

```
A    D.java
```

```
A    G.java
```

```
A    K.java
```

```
svn commit -m "r0" --username=red
```

```
Adding    C.java
```

```
Adding    D.java
```

```
Adding    G.java
```

```
Adding    K.java
```

```
Transmitting file data ....
```

```
r1:
```

```
mkdir branches
```

```
svn copy file:///home/s285484/misp_svn/lab/trunk
```

```
file:///home/s285484/misp_svn/lab/branches/blueBranch -m "create blueBranch"
```

```
svn switch file:///home/s285484/misp_svn/lab/branches/blueBranch --username=blue
```

```
svn rm *
```

```
cp ~/commits/commit1/* .
```

```
svn add *
```

```
svn commit -m "r1" --username=blue
```

```
r2:
```

```
svn rm *
cp ~/commits/commit2/* .
svn add *
svn commit -m "r2" --username=blue
```

```
r3:
svn copy file:///home/s285484/misp_svn/lab/branches/blueBranch
file:///home/s285484/misp_svn/lab/branches/redBranch2 -m "create redBranch2"
svn merge file:///home/s285484/misp_svn/lab/branches/redBranch2
svn rm *
cp ~/commits/commit3/* .
svn add *
svn commit -m "r3" --username=red
```

```
r4:
svn sw file:///home/s285484/misp_svn/lab/branches/blueBranch --username=blue
svn rm *
cp ~/commits/commit4/* .
svn add *
svn commit -m "r4" --username=blue
```

```
r5:
svn sw file:///home/s285484/misp_svn/lab/branches/redBranch2 --username=red
svn rm *
cp ~/commits/commit5/* .
svn add *
svn commit -m "r5" --username=red
```

```
r6:
svn sw file:///home/s285484/misp_svn/lab/branches/blueBranch --username=blue
svn rm *
cp ~/commits/commit6/* .
svn add *
svn commit -m "r6" --username=blue
```

```
r7:
svn sw file:///home/s285484/misp_svn/lab/branches/develop
svn rm *
cp ~/commits/commit7/* .
svn add *
svn commit -m "r7" --username=red
```

```
r8:
svn sw file:///home/s285484/misp_svn/lab/branches/blueBranch
```

```
svn rm *  
cp ~/commits/commit8/* .  
svn add *  
svn commit -m "r8" --username=blue
```

```
r9:  
svn sw file:///home/s285484/misp_svn/lab/branches/develop  
svn rm *  
cp ~/commits/commit9/* .  
svn add *  
svn commit -m "r9" --username=red
```

```
r10:  
svn sw file:///home/s285484/misp_svn/lab/branches/blueBranch  
svn rm *  
cp ~/commits/commit10/* .  
svn add *  
svn commit -m "r10" --username=blue
```

```
r11:  
svn rm *  
cp ~/commits/commit11/* .  
svn add *  
svn commit -m "r11" --username=blue
```

```
r12:  
svn sw file:///home/s285484/misp_svn/lab/branches/develop  
svn merge file:///home/s285484/misp_svn/lab/branches/blueBranch  
svn rm *  
cp ~/commits/commit12/* .  
svn add *  
svn commit -m "r12" --username=red
```

```
r13:  
svn sw file:///home/s285484/misp_svn/lab/branches/redBranch2 --username=red  
svn rm *  
cp ~/commits/commit12/* .  
svn add *  
svn commit -m "r12" --username=red
```

```
r14:  
svn sw file:///home/s285484/misp_svn/lab/branches/develop  
svn merge file:///home/s285484/misp_svn/lab/branches/redBranch2  
svn rm * --force
```

```
cp ~/commits/commit14/* .  
svn add *  
svn commit -m "r14" --username=red
```

Вопросы по самой лабе:

Как решали конфликты?

Действия с ветками: создание в одну команду и в две

## Вывод

В данной лабораторной работе были рассмотрены основы работы с системами контроля версий на примере двух разных VCS: Git и Subversion. Мы научились инициализировать репозитории и управлять процессом разработки.