Introduction to Programming

Lab 12

Alaa Aldin Hajjar, Mahmoud Naderi, Marko Pezer,
Mosab Mohamed, Rawan Ali

Agenda

- Using file streams
- Exceptions
- VCS

FileOutputStream

Method	Description
`void write(byte[] ary)`	It is used to write **ary.length** bytes from the byte array to the file output stream.
`void write(byte[] ary, int off, int len)`	It is used to write **len** bytes from the byte array starting at offset **off** to the file output stream.
`void write(int b)`	It is used to write the specified byte to the file output stream.
`void close()`	It is used to close the file output stream.

FileInputStream

Method	Description
`int available()`	It is used to return the estimated number of bytes that can be read from the input stream.
`int read()`	It is used to read the byte of data from the input stream.
`int read(byte[] b)`	It is used to read up to **b.length** bytes of data from the input stream.
`int read(byte[] b, int off, int len)`	It is used to read up to **len** bytes of data from the input stream.
`long skip(long x)`	It is used to skip over and discards x bytes of data from the input stream.
`void close()`	It is used to close the stream.

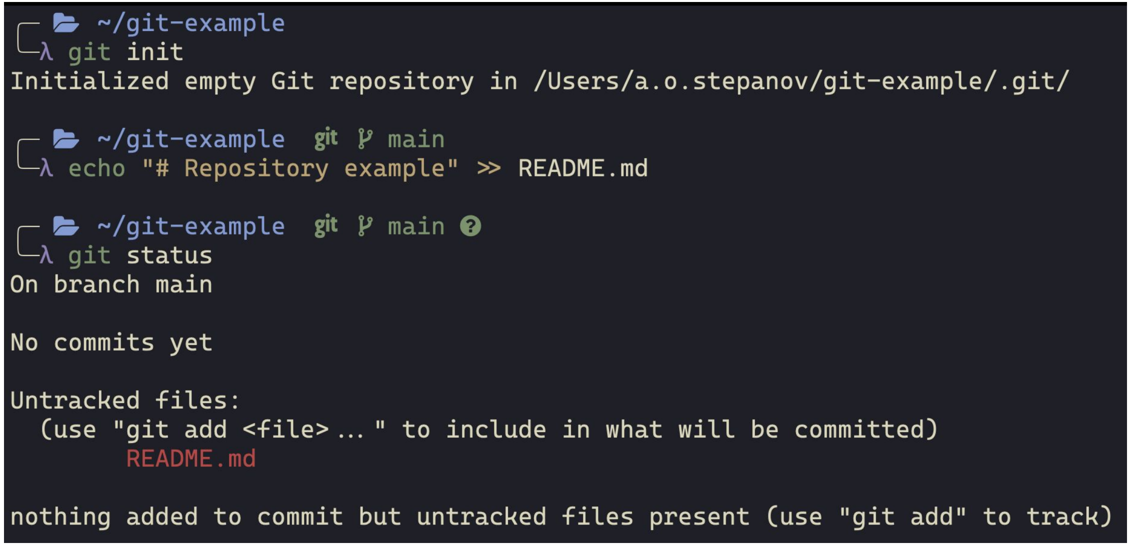
File Copying Example

```java
import java.io.*;

public class Main {
    public static void main(String[] args) {
        try(FileInputStream in = new FileInputStream("input.txt");
            FileOutputStream out = new FileOutputStream("output.txt"))
        {
            byte[] buffer = new byte[in.available()];
            in.read(buffer, 0, buffer.length);
            out.write(buffer, 0, buffer.length);
        } catch(IOException ex){
            System.out.println(ex.getMessage());
        }
    }
}
```

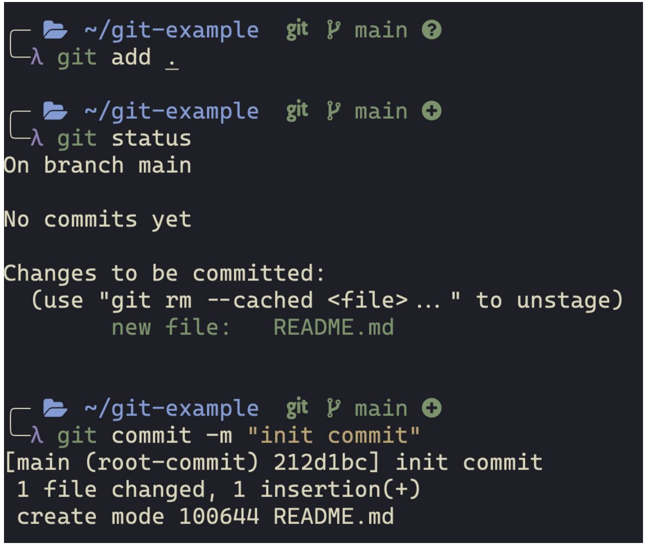
Git

Git is a distributed version control system that tracks changes in any set of computer files, usually used for coordinating work among programmers who are collaboratively developing source code during software development. Its goals include speed, data integrity, and support for distributed, non-linear workflows.

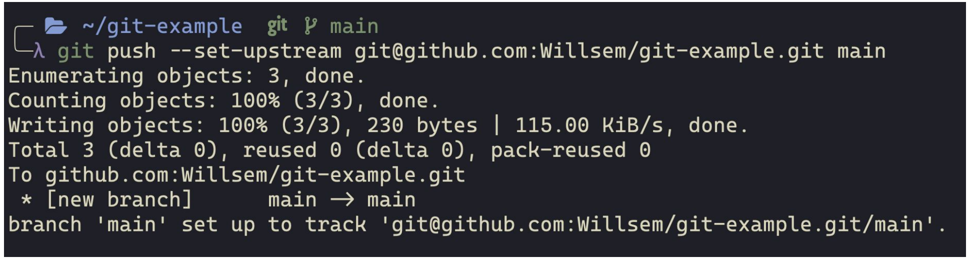
Git Example



```
┌─ 🗁 ~/git-example
└─λ git init
Initialized empty Git repository in /Users/a.o.stepanov/git-example/.git/

┌─ 🗁 ~/git-example  git  main
└─λ echo "# Repository example" >> README.md

┌─ 🗁 ~/git-example  git  main ?
└─λ git status
On branch main

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        README.md

nothing added to commit but untracked files present (use "git add" to track)
```

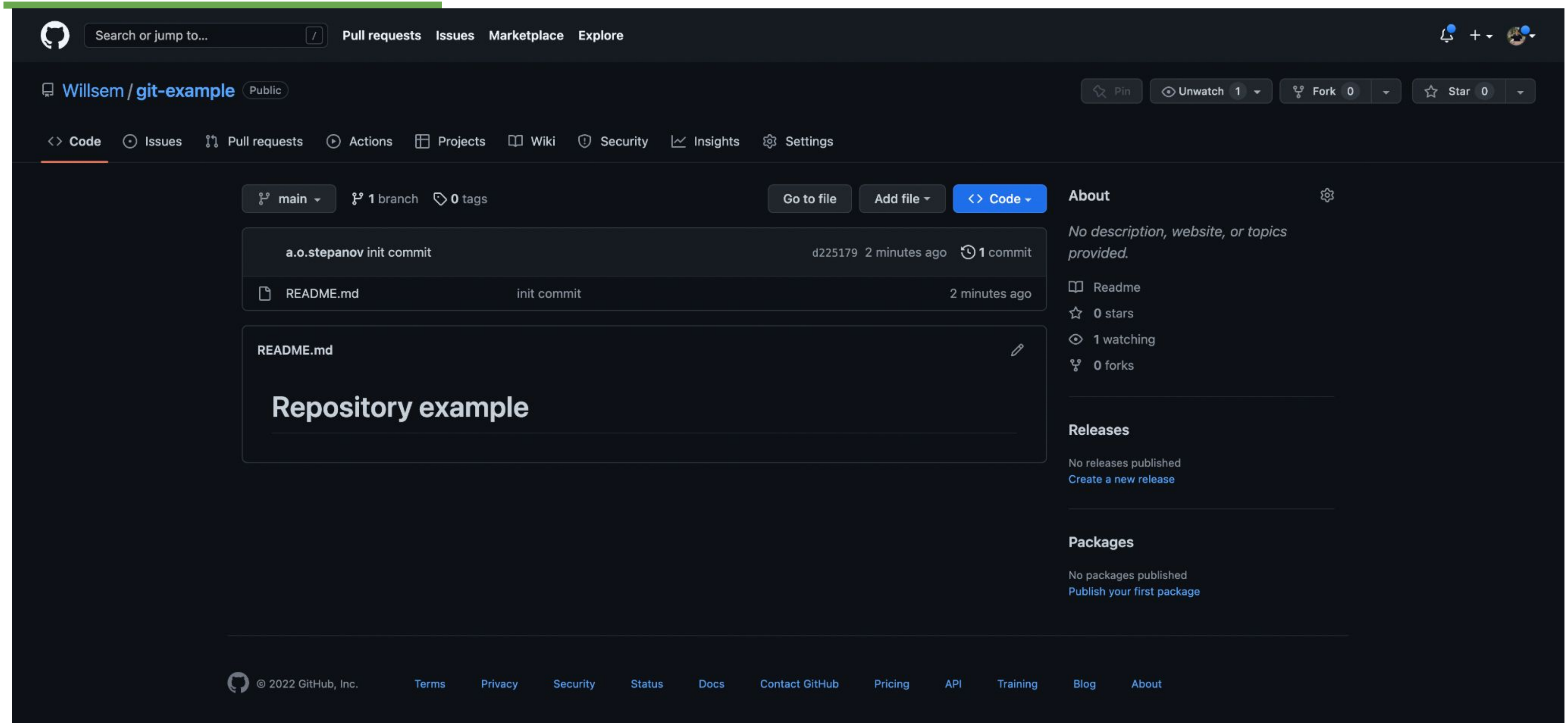


```
┌─ 🗁 ~/git-example  git  main ?
└─λ git add .

┌─ 🗁 ~/git-example  git  main ⊕
└─λ git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   README.md


┌─ 🗁 ~/git-example  git  main ⊕
└─λ git commit -m "init commit"
[main (root-commit) 212d1bc] init commit
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
```



```
┌─ 🗁 ~/git-example  git  main
└─λ git push --set-upstream git@github.com:Willsem/git-example.git main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 230 bytes | 115.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:Willsem/git-example.git
 * [new branch]      main -> main
branch 'main' set up to track 'git@github.com:Willsem/git-example.git/main'.
```

Git Example



a.o.stepanov init commit d225179 2 minutes ago 1 commit

README.md init commit 2 minutes ago

README.md

Repository example

Exercise 1

1. Create VCS account, if doesn’t exist for you, e.g. GitHub account
2. Create repository for ITP course exercises **(not assignments)**
3. Add at least previous week exercises to this repository
4. Keep doing this constantly for the remaining part of the course

Exercise 2

Write a program that reads the data from a text file and writes it into another text file. Handle the following exceptions:

- input file does not exist
- no write-permission for the output file

Exercise 2 (VCS)

Commit a new task to your repository

Exercise 3

Write a program that reads an input from the file, which accepts two integer parameters, and divides the first integer by the second.

You have to catch all the possible exceptions (such as parsing errors, non-integer errors, and arithmetic errors) and print the appropriate message.

Exercise 3 (VCS)

Commit a new task to your repository

Exercise 4

Update the code from a previous task to throw suppressed exceptions after printing the error messages

Exercise 4 (VCS)

Commit the changes of the third task, which you added in the previous one

Exercise 5

The method below downloads an image (actually any file). Your task is to edit the code so that it could handle all the possible exceptions.

```java
public static void saveImage(String imageUITP_Week12_DSAI03ITP_Week12_DSAI03rl) {
    URL url = new URL(imageUrl);
    String fileName = url.getFile();
    String destName = "./figures" + fileName.substring(fileName.lastIndexOf("/"));
    System.out.println(destName);

    InputStream is = url.openStream();
    OutputStream os = new FileOutputStream(destName);

    byte[] b = new byte[2048];
    int length;

    while ((length = is.read(b)) != -1) {
        os.write(b, 0, length);
    }

    is.close();
    os.close();
}
```

Exercise 5 (VCS)

Commit the changes.

Create a remote repository in GitHub then push today’s code to it.

Extra: add README.md file with a description of this repository to the root of it

References

- FileOutputStream
- FileInputStream
- Git documentation