Лабораторна робота №9

ОЗНАЙОМЛЕННЯ З СИСТЕМОЮ

УПРАВЛІННЯ ВЕРСІЯМИ GIT

***Мета робот:***

- придбання та закріплення навичок створення та використання

репозиторіїв Git;

- розробки найпростіших Java програм з використанням сховища Git;

- ознайомлення з можливостями середовища розробки Eclipse та його

розширення EGit.

**1 ІНДИВІДУАЛЬНЕ ЗАВДАННЯ**

1) Подготовить хранилище для размещения проекта:

- создать директорию с названием хранилища и перейти в неё:

mkdir surname\_name

cd surname\_name

- выполнить инициализацию локального хранилища:

git init

- создать директорию с названием проекта, например "task01", и перейти

в неё;

- в директории проекта создать текстовый файл ".gitignore" с

содержимым:

\*.class

- зафиксировать изменения в хранилище:

git add .

git commit -m "task01: init storage, add .gitignore"

2) Выполнить разработку простейшей консольной программы на Java,

например, выводящей на экран все аргументы командной строки, заданные

при запуске и разместить её в локальном хранилище:

- в директории проекта "task01" создать файл "Main.java" с содержимым

- выполнить компиляцию и проверку работоспособности:

javac Main.java

java Main arg1 arg2 arg3

- добавить всё содержимое директории в индекс и зафиксировать

изменения в репозитории (выполнить коммит):

git add .

git commit -m "task01: create simple java program"

- отредактировать файл "Main.java", добавить комментарии в стиле

- проверить результат работы javadoc:

javadoc -private -d doc \*.java

start doc/index.html

- отредактировать файл ".gitignore":

doc

\*.class

- зафиксировать изменения в репозитории и просмотреть историю

изменений:

git commit -am "task01: add doc comments"

git log --stat

3) На информационной странице проекта на сайте "gitlab.com" получить

(скопировать в буфер обмена) адрес для доступа к удалённому хранилищу

по протоколу HTTPS в виде "https://gitlab.com/namespace/project.git"

(обеспечивается преподавателем) и подключить удалённый репозиторий к

локальному:

- перейти в директорию ранее созданного локального хранилища;

- подключить удалённый репозиторий для последующей синхронизации

(вставить адрес из буфера обмена):

git remote add origin https://gitlab.com/namespace/project.git

- опубликовать проект в удалённом репозитории, сохранив связь между

локальными и удалёнными ветками (флаг "-u"):

git push -u origin master

При наличии ошибки выполнения предыдущей команды нужно

повторить команду с флагом "-f", т.е. :

git push -f origin master

4) Продолжить разработку программы в другом каталоге (на новом рабочем

месте):

- сменить текущую директорию, создать новую и перейти в неё.

- выполнить клонирование удалённого хранилища:

git clone https://gitlab.com/namespace/project.git

-выполнить разработку проекта: вывести строки массива "args" –

параметра метода "Main.main()", обеспечив лексикографический

ипорядок, и зафиксировать изменения в локальном хранилище;

- зафиксировать изменения в удалённом репозитории:

git push origin master

- на сайте "gitlab.com" проверить удалённый репозиторий по адресу

"https://gitlab.com/namespace/project", разделы Files и Commits.

5) Выполнить разработку Java проекта в Eclipse с размещением в удалённом

хранилище согласно требованиям раздела "Настройка среды

разработки / Структура проекта" по адресу:

6

"https://gitlab.com/khpi-se-java/info/wikis/Eclipse"

- в Eclipse создать Java проект, используя название проекта GitLab;

- средствами EGit добавить существующий локальный репозиторий

(меню "Window", команда "Show View/Other/Git/Git Repositories",

представление "Git Repositories", команда "Add an existing local Git

repository", указать директорию с существующим хранилищем);

- если название проекта Eclipse совпадает с названием директории

программы, созданной в п.1, то эту директорию следует переименовать

в файловом менеджере (назвать, например, "src");

- поместить проект Eclipse в существующее локальное хранилище

(команда "Team/Share Project..." контекстного меню проекта в Package

Explorer);

- создать пакет вида "корневой\_пакет.task01";

- переместить исходные файлы программы, созданной в п.2, в

директорию пакета "task01" проекта Eclipse и удалить старую

директорию;

- выполнить компиляцию и запуск программы в Eclipse;

-при необходимости выполнить рефакторинг, добавить комментарии;

- выполнить генерацию Javadoc и убедиться что директория с

документацией исключена из репозитория Git (при необходимости

отредактировать файл .gitignore);

- зафиксировать изменения в локальном и удалённом репозиториях (add

to index, commit, push).

- проверить состояние проекта на сайте "gitlab.com".

6) Проверить исходный код с помощью модуля расширения "Checkstyle"

https://gitlab.com/khpi-se-java/info/wikis/Software#eclipse

выполнить рефакторинг с учётом полученных замечаний и зафиксировать

изменения в локальном и удалённом репозиториях.

7) В отчёте привести последовательность действий, описание и копии экрана

выполнения задания.

**2 ОПИС ПРОГРАМИ**

**2.1 Ієрархія та структура класів**

**class Main-містить роботу з параметрами командної строки**

**2.2 Опис програми**

На рисунку 2.1 наведена структура розробленої програми

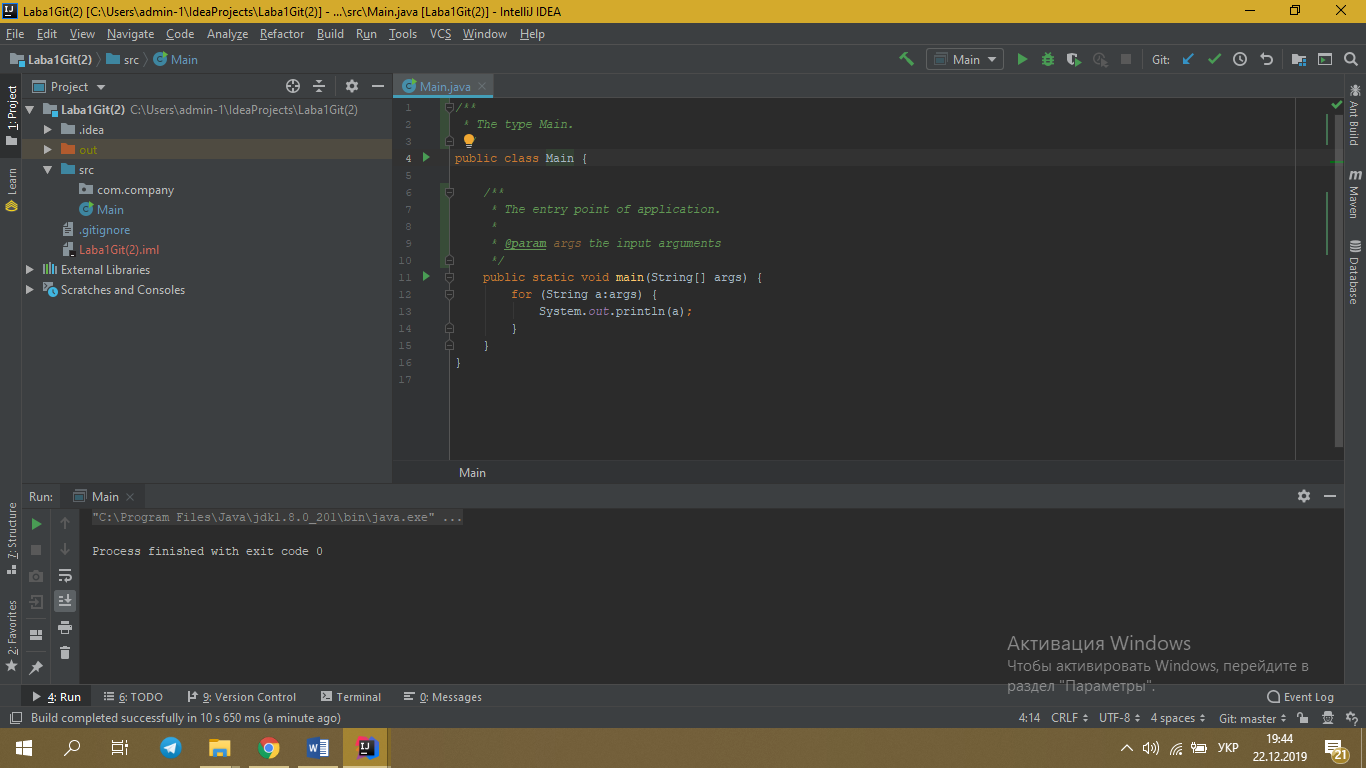


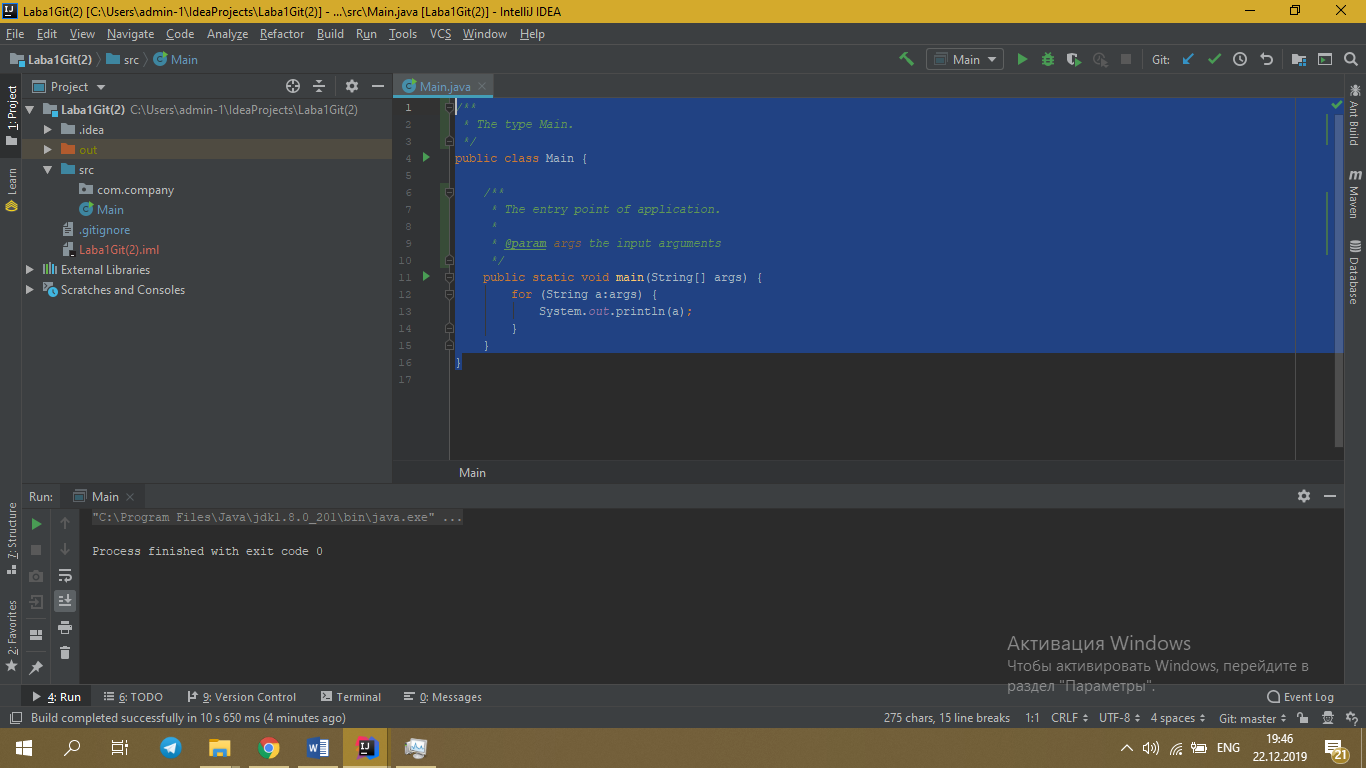
Рис 2.1 – структура програми

**2.3 Важливі фрагменти програми**

2.3.1 Файл*Main.java*

*/\*\*  
 \* The type Main.  
 \*/*public class Main {  
  
 */\*\*  
 \* The entry point of application.  
 \*  
 \** ***@param*** *args the input arguments  
 \*/* public static void main(String[] args) {  
 for (String a:args) {  
 System.*out*.println(a);  
 }  
 }  
}

**3 РЕЗУЛЬТАТИ**



**ВИСНОВКИ**

Навчились працювати з контролером версій git.