

**Мета:** ознайомитися з процесом встановлення та опціями інтегрованого середовища JetBrains IntelliJ IDEA, створити просту програму мовою Java, виконати компіляцію та запустити програму. Навчитися зберігати версії проєкту за допомогою системи контролю версій Git.

### Хід роботи

- 1 Запустити інтегроване середовище для розробки Java-програм.
- 2 Створити новий проєкт, а в ньому файл з вихідним кодом.
- 3 Замінити у вихідному коді "This task done by ..." на Ваші Ім'я, прізвище та групу.

```

1 public class HelloWorldDemo {
2     Run | Debug
3     public static void main(String[] argv) {
4         System.out.println(x:"Hello world");
5         System.out.println(x:"This task done by Lashko Ruslan, group PZ-21-1/9");
6     }
7 }

```

Рисунок 1 – Результат виконання пункту 1-3

- 4 Скомпілювати та виконати програму.

```

PS C:\Users\katar\Desktop\УЧ\ТП> java HelloWorldDemo
Hello world
This task done by Lashko Ruslan, group PZ-21-1/9

```

Рисунок 2 – Результат виконання пункту 4

					<b>ФКЗЕ. 121ТП00.01ЛР</b>			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.	Лашко Р.А.				Ознайомлення з інтегрованим середовищем розробки Java програм. Використання систем контролю версій на основі Git.	Літ.	Арк.	Аркушів
Перевір.	Новодравнова						1	7
Реценз.						Група ПЗ-21-1/9		
Н. Контр.	Саприкіна І.Г.							
Затверд.	Саприкіна І.Г.							

5 Зареєструватися на будь-якому публічному Git-сервісі.

Обрав найпопулярніший, GitHub.

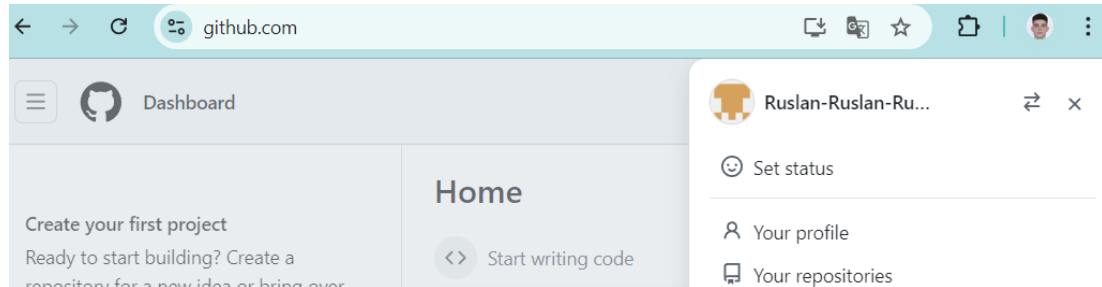


Рисунок 3 – Результат виконання пункту 5

6 Створити новий віддалений репозиторій для збереження лабораторних робіт з Java. Тип віддаленого репозиторію обрати самостійно. Свій вибір обґрунтувати.

7 Надати викладачу доступ до цього репозиторію. Рівень доступу визначити самостійно. Вибір обґрунтувати.

Обрав public тип, щоб хто завгодно зміг подивитись репозиторій. Назва Ruslan-Ruslan-Ruslan/labatm

### Who has access

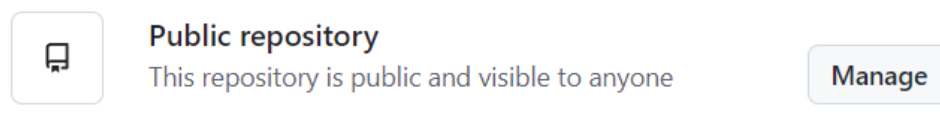


Рисунок 4 – Результат виконання пункту 7

8 Створений проект з найпростішою Java-програмою розташувати в цьому репозиторії. Завантажити першу версію у репозиторій.

					ФКЗЕ. 121ТП00.01ЛР	Арк.
						2
Змін.	Арк.	№ докум.	Підпис	Дата		

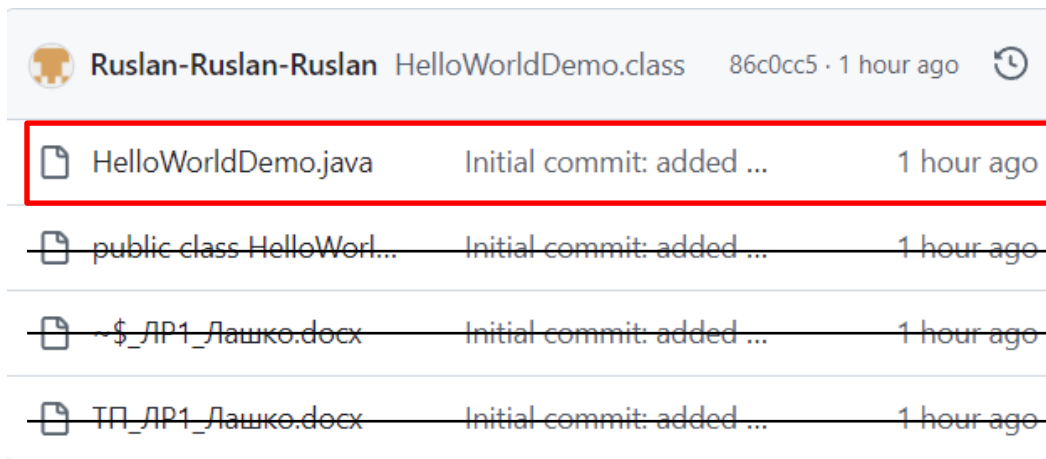


Рисунок 5 – Результат виконання пункту 8

```
PS C:\Users\katar> cd C:\Users\katar\Desktop\УЧ\ТП
PS C:\Users\katar\Desktop\УЧ\ТП> git remote add origin https://github.com/Ruslan-Ruslan-Ruslan/labatm
error: remote origin already exists.
PS C:\Users\katar\Desktop\УЧ\ТП> git remote set-url origin https://github.com/Ruslan-Ruslan-Ruslan/labatm
PS C:\Users\katar\Desktop\УЧ\ТП> git add .
PS C:\Users\katar\Desktop\УЧ\ТП> git commit -m "Initial commit: added HelloWorld program"
Author identity unknown

*** Please tell me who you are.

Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'katar@kateryna.(none)')
PS C:\Users\katar\Desktop\УЧ\ТП> git config --global user.email "lashkoruslan@gmail.com"
PS C:\Users\katar\Desktop\УЧ\ТП> git config user.name "Ruslan-Ruslan-Ruslan"
PS C:\Users\katar\Desktop\УЧ\ТП> git commit -m "Initial commit: added HelloWorld program"
[master (root-commit) 7b65fcc] Initial commit: added HelloWorld program
5 files changed, 12 insertions(+)
create mode 100644 HelloWorldDemo.class
create mode 100644 HelloWorldDemo.java
create mode 100644 public class HelloWorldDemo {.java
create mode 100644 "~$ _\320\233\320\2401_\320\233\320\260\321\210\320\272\320\276.docx"
create mode 100644 "~\320\242\320\237_\320\233\320\2401_\320\233\320\260\321\210\320\272\320\276.docx"
PS C:\Users\katar\Desktop\УЧ\ТП> git push -u origin master
info: please complete authentication in your browser...
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
```

Рисунок 6 – Покрокове виконання пункту 8

9 Внести зміни у вихідний код, додавши у нього рядок, де на екран виводиться додатковий рядок тексту, у якому повідомляється поточна дата. Змінений варіант проєкту завантажити у віддалений репозиторій.

```
C: > Users > katar > Desktop > УЧ > ТП > J HelloWorldDemo.java > ...
1 public class HelloWorldDemo {
2     public static void main(String[] argv) {
3         System.out.println("Hello world");
4         System.out.println("This task done by Lashko Ruslan, group
5
6         LocalDate currentDate = LocalDate.now();
7         System.out.println("Current date: " + currentDate);
8     }
9 }
10
```

Рисунок 7 – Покрокове виконання пункту 9

```
PS C:\Users\katar\Desktop\УЧ\ТП> java HelloWorldDemo
Hello world
This task done by Lashko Ruslan, group PZ-21-1/9
Current date: 2024-09-08
```

Рисунок 8 – Результат виконання зміненого коду

```
PS C:\Users\katar\Desktop\УЧ\ТП> git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 234 bytes | 234.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Ruslan-Ruslan-Ruslan/labatm
7b65fcc..86c0cc5 master -> master
```

Рисунок 9 – Завантаження зміненого варіанту

					ФКЗЕ. 121ТП00.01ЛР	Арк.
						4
Змін.	Арк.	№ докум.	Підпис	Дата		

## Контрольні питання:

1 Які інструменти потрібні для створення Java програм (мінімальний набір та професійний)?

- текстовий редактор, бажано з підсвіткою синтаксису мови Java;
- JDK (Java Development Kit)
- засіб контролю відповідності стандартам тексту програми;
- засіб компіляції Java-програм;
- засіб для запуску Java-програм на виконання (Java Virtual Machine або JVM);
- засіб відлагодження (помічник у виправленні помилок або debugger).

2 Які файли проєкту будуть створені під час виконання цієї лабораторної роботи?

Файли вихідного коду та файли git.

3 Яке ПЗ повинно бути встановлено на ПК, або іншому пристрої, де потрібно виконувати Java програми?

JDK (Java Development Kit) для компіляції та запуску Java-коду та IDE або текстовий редактор для написання коду (наприклад, IntelliJ IDEA або Visual Studio Code).

4 Навіщо потрібен контроль версій.

Контроль версій дозволяє зберігати проміжні результати роботи над проєктом, відстежувати зміни, повертатися до попередніх версій у разі помилок або втрати даних, а також ефективно координувати роботу кількох людей над одним проєктом.

5 Які системи контролю версій існують.

					<b>ФКЗЕ. 121ТП00.01ЛР</b>	Арк.
						5
Змін.	Арк.	№ докум.	Підпис	Дата		

- локальні системи контролю версій;
- розподілені системи контролю версій;
- клієнт-серверні (централізовані) системи контролю версій.

6 Які особливості системи контролю версій Git Вам відомі.

Кожен користувач має повну копію репозиторію, швидка робота, легка робота з гілками.

7 Які основні поняття, властиві Git Ви знаєте.

Репозиторій - місце, де зберігається весь код і історія змін.

Коміт - збереження змін в репозиторії.

Гілка - версія проєкту, яка розвивається паралельно з іншими гілками.

Злиття - об'єднання змін із різних гілок.

8 Як розмістити проєкт на сервісі Git.

Створіть новий репозиторій на GitHub.

Додайте віддалений репозиторій: `git remote add origin https://github.com/ім'я_на_GitHub/назва_репозиторію.git`

Завантажте код у віддалений репозиторій: `git push -u origin master`

9 Опишіть, як зберегти поточну версію проєкту у Git-репозиторії.

Додайте файли до індексу: `git add .`

Створіть коміт: `git commit -m "Повідомлення про зміни"`

Завантажте коміт у віддалений репозиторій: `git push origin master.`

10 Як повернутись до попередньої версії проєкту.

Перегляньте історію комітів: `git log`

Поверніться до потрібного коміту: `git checkout <commit-id>`

Щоб повернути зміни у гілку: `git reset --hard <commit-id>`

**Висновок:** ознайомився з процесом встановлення та опціями інтегрованого середовища JetBrains IntelliJ IDEA, створити просту програму мовою Java, виконати компіляцію та запустити програму. Навчитися зберігати версії проєкту за допомогою системи контролю версій Git.

					ФКЗЕ. 121ТП00.01ЛР	Арк.
						7
Змін.	Арк.	№ докум.	Підпис	Дата		