



Заняття 1: Вступ до курсу

JAVA. Starter



Ігор Лютак

Контактні дані

Ігор Лютак

Скайп ihorlt

Пошта igor@lyutak.com

Мета

Ознайомитись із основними підходами при розробленні програмного забезпечення

Освоїти основи роботи із середовищем розроблення коду

Навчитись базовим командам при роботі із інструментами розробника програмного забезпечення

Вхідні вимоги:

Наявність операційної системи Лінукс / Windows

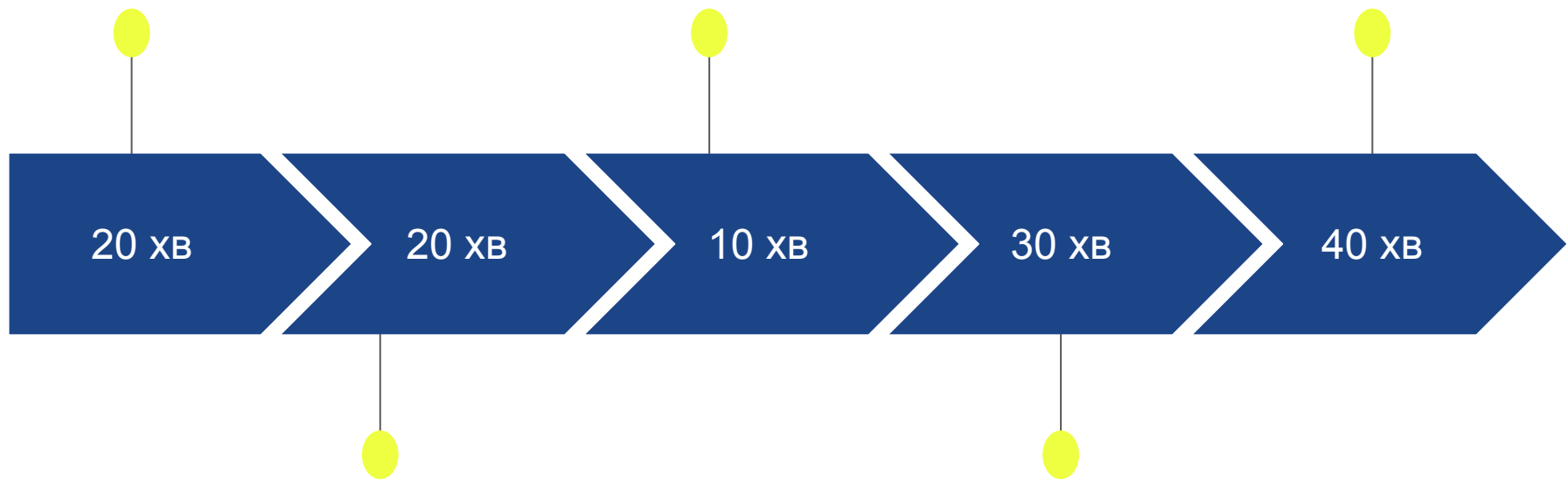
Базові навички роботи в Лінукс / Windows

План

Огляд обов'язків
до девелопера

Налаштування
Git

Практична
робота



Операційна система
Лінукс. Налаштування
Джава та IntelliJ

Налаштування
GitHub. Основні
команди роботи з Git

Література

<https://docs.oracle.com/javase/tutorial/>

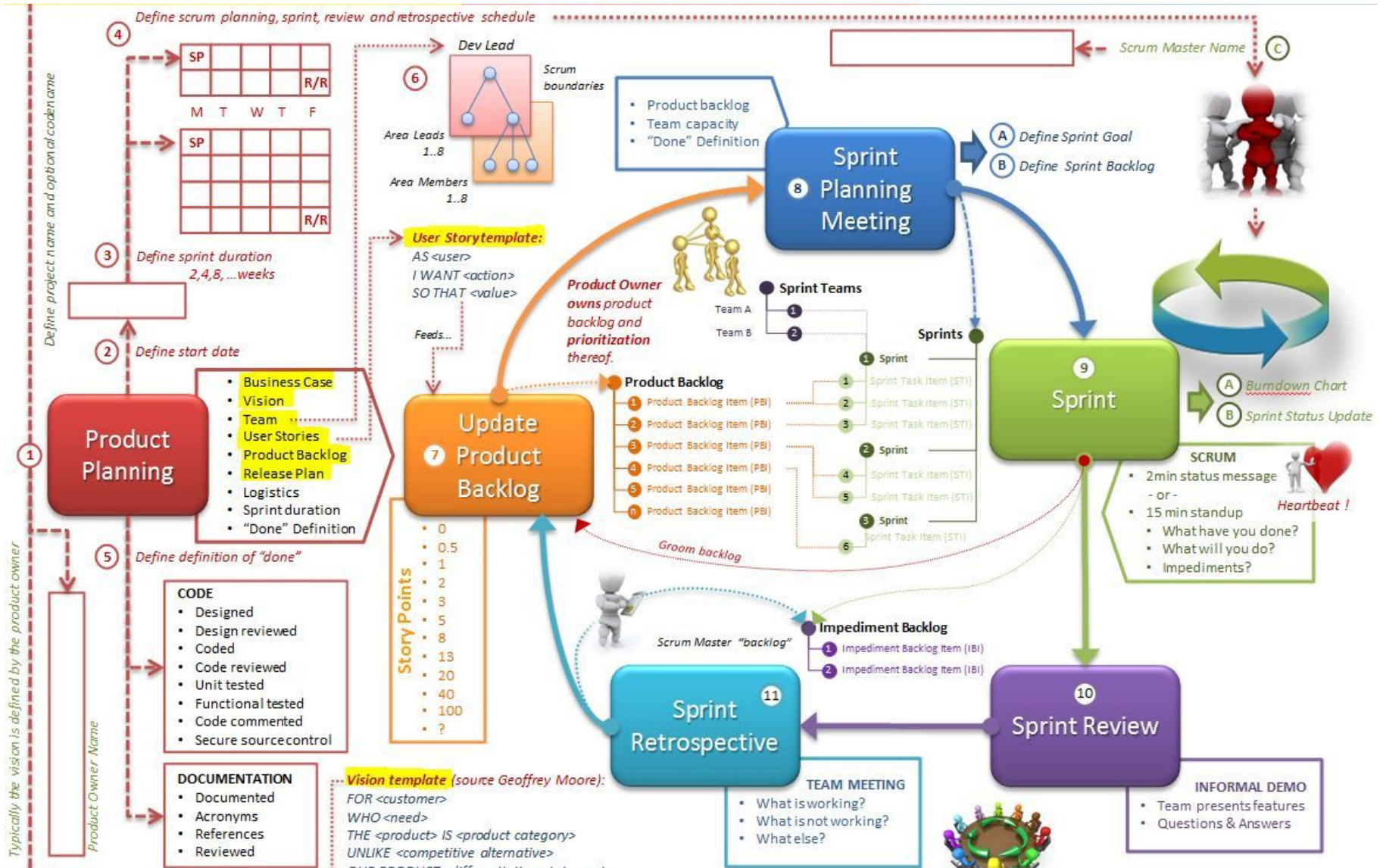
<https://www.jetbrains.com/idea/documentation/>

Java 8. Полное руководство

Философия Java

<https://githowto.com/ru>

Поняття методології розроблення ПО (Scrum)



Додатково

http://info.javarush.ru/javarush_articles/2015/08/19/%D0%92%D0%B2%D0%B5%D0%B4%D0%B5%D0%BD%D0%B8%D0%B5-%D0%B2-Scrum.html

Інсталяція Джава на Лінукс

В консолі ввести команди:

```
sudo apt update
```

```
sudo add-apt-repository ppa:webupd8team/java
```

```
sudo apt update
```

```
sudo apt install oracle-java8-installer
```

або для 9-ї версії

```
sudo apt install oracle-java9-installer
```

```
sudo update-alternatives --config java
```

Для змінної середовища JAVA_HOME ввести рядок

JAVA_HOME="YOUR_PATH" у файлі, відкритий командою:

```
sudo nano /etc/environment
```

Скачати із веб-сайту версію Ultimate:

<https://www.jetbrains.com/idea/download/#section=linux>

Розпакувати, і запустити із консолі командою

`./idea.sh`

Ввести ліцензійні параметри

Прийняти набір пакетів по замовчуванню

Створити проект Java → Вибрати Project SDK

(usr/lib/jvm/java-8-oracle) → Create project from template (Hello

World) → назва проекту у папці hello → Почекати, поки

проіндексуються файли із JDK → запустити на виконання проект.

Інсталяція Джава на Windows

Для полегшення інсталяції необхідно поставити менеджер пакетів

Chocolatey. Запустити **PowerShell** версії 3+

Комбінація клавіш  + R і ввести **PowerShell**, запустити в режимі адміністратора.

В консолі, що відкриється ввести команду:

```
Set-ExecutionPolicy -ExecutionPolicy AllSigned
```

```
iwr https://chocolatey.org/install.ps1
```

```
-UseBasicParsing | iex
```

Відкрити консоль Windows як адміністратор (Win + r → cmd) і ввести:

Інсталяція Джава **C:\> choco install jdk8**

Інсталяція Git **C:\> choco install git**

Розширення пакету

```
C:\> choco install chocolatey-core.extension
```

Інсталяція Chrome `C:\> choco install googlechrome`

Інсталяція Firefox `C:\> choco install firefox`

Інсталяція IntelliJ IDEA (Ultimate Edition)

```
C:\> choco install intellijidea-community
```

Інсталяція Maven `C:\> choco install maven`

Інсталяція MySQL `C:\> choco install mysql.installer`

Інсталяція MySQL Workbench

```
C:\> choco install mysql.workbench
```

Інсталяція Джава на Windows

продовження

Інсталяція Apache Tomcat `C:\> choco install tomcat`

Інсталяція Eclipse `C:\> choco install eclipse-java-kepler`

Інсталяція Eclipse IDE for Java EE Developers (Luna)

`C:\> choco install eclipse-jee-luna`

Або STS `C:\> choco install springtoolsuite`

Інсталяція Skype `C:\> choco install skype`

Інсталяція Winrar `C:\> choco install winrar`

Інсталяція LibreOffice `C:\> choco install libreoffice`

Для оновлення копії пакету `choco upgrade libreoffice`

Перша програма

```
/**  
 * Коментар  
 */  
class HelloWorldApp {                                // декларація класу  
  
    public static void main(String[] args) {          //метод, з якого починається  
                                                         виконання програми  
  
        System.out.println("Hello World!");          // Виводить рядок  
    }  
}  
  
/* text */  
  
// text        відповідно багаторядковий та однорядковий коментар
```

Інструменти розроблення / Git

Git — потужна і складна розподілена система контролю версій.

Інсталяція Git на комп'ютер

```
apt install git
```

Установка імені і електронної пошти

```
git config --global user.name "Your Name"
```

```
git config --global user.email  
"your_email@whatever.com"
```

Подивитись на зміни конфігурації

```
git config --list
```

Створення Git репозиторію

продовження

Зайти в папку проекту

```
cd hello
```

Створюємо новий
репозиторій

```
git init
```

Добавляємо файли проекту

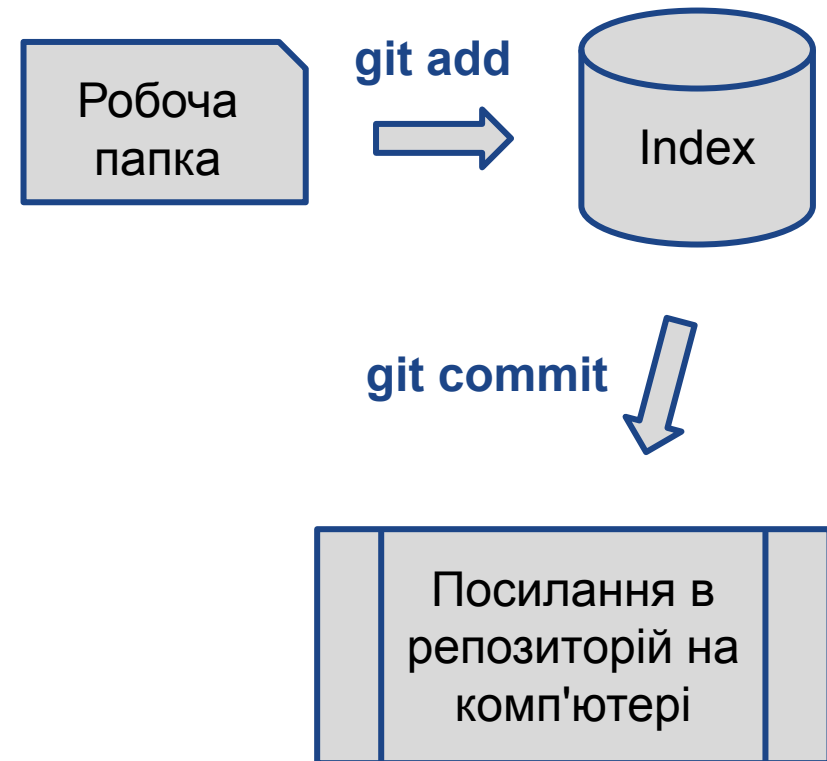
```
git add src/Main.java
```

Перевіряємо статус файлів

```
git status
```

Записуємо додані файли у
репозиторій

```
git commit -m "First"
```



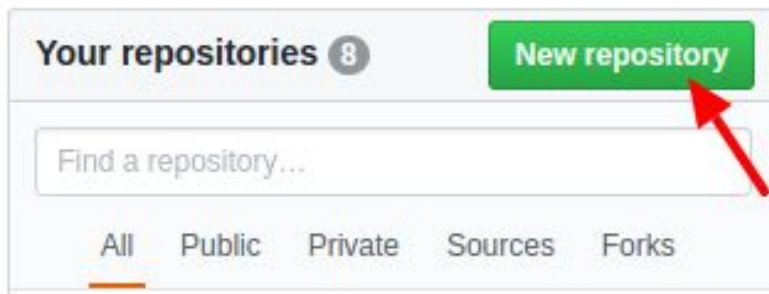
Серверний Git репозиторій

продовження

Створюємо користувача на сервері

<https://github.com/>

Створюємо новий репозиторій на сервері



Називаємо і натискаємо

`Create repository`

Додаємо інформацію про серверний репозиторій до локального git

```
git remote add origin  
https://github.com/ ...  
.git
```

Записуємо файли з локального репозиторію на серверний

```
git push -u origin master
```

Вводите дані користувача (ел. пошта і пароль)

Відміна змін

Внесіть зміни у файл проекту.
Додайте до репозиторію

```
git add src/Main.java
```

Перевірте стан

```
git status
```

Скиньте всі проіндексовані
зміни до тих, які записані в
локальному репозиторію

```
git reset HEAD  
src/Main.java
```

Команда **reset** (за замовчуванням) не змінює робочий каталог. Тому він все ще містить небажані зміни. Ми можемо використовувати команду **checkout**, щоб видалити небажані зміни в робочому каталозі до версії коміту

```
git checkout src/Main.java
```

```
git status
```

або для всіх файлів

```
git checkout master
```

Відгалуження репозиторію

Створити нову гілку і зробити її активною

```
git checkout -b  
    project_branch
```

Створюємо нову гілку, а потім впливаємо в неї зміни з гілки, що є на github

```
git pull origin  
    Project_branch
```

Якщо команда не спрацювала, треба спробувати виконати оновлення:

```
git remote update
```

Переглянути список гілок

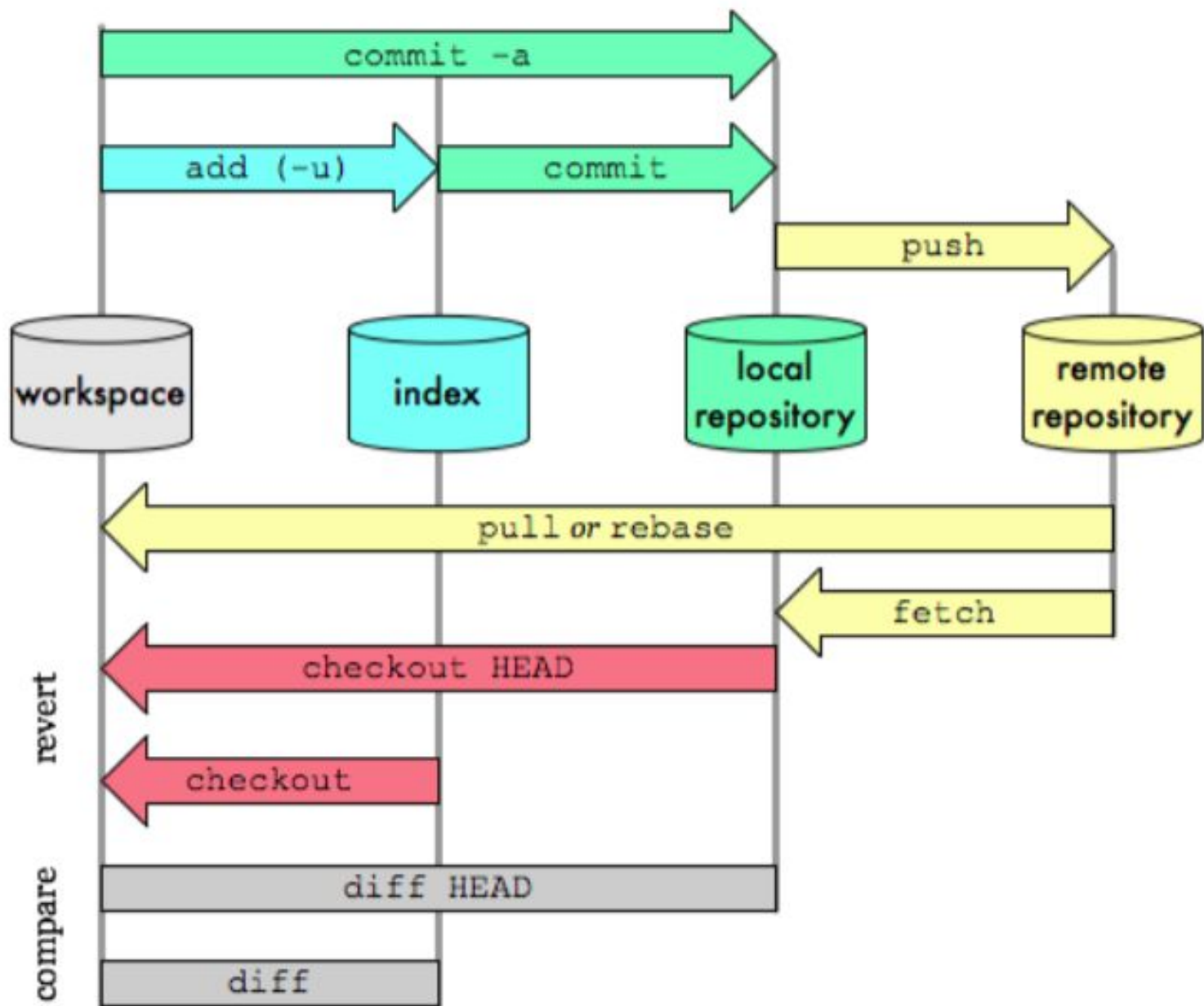
```
git branch
```

Як правило, в репозиторії існує дві основні гілки – dev і master. dev – загальна гілка для розроблення і тестування. Master – гілка для викладки продукту на стороні сервера.

Переглянути історію змін

```
git log
```

Робота Git



Завдання

Проісталювати необхідні пакети програм

Створити репозиторій на сервері із назвою JavaStarter

Створити гілку dev

Скласти програму, що містить список із 3 завдань і виводить ті, які необхідно зробити завтра. (Виводить в консоль, текст завдання)

Завантажити на сервер завдання у гілку dev

Створити файл README.me і описати роботу програми

Зареєструватись на <https://trello.com/> для командної роботи