

Лабораторна робота №2. Вступ до програмування, основи debug процесу.

1 Вимоги

1.1 Розробник

- Носов Микола
- студент групи кіт 120-б
- виконано 20.10.2020

1.2 Загальне завдання:

Створити git репозиторій на gitlab та скопуватися із ним зі свого комп'ютера. Освоїти debug.

Крок 1:

Знайти свій особистий репозиторій на gitlab та скопувати його у вільну папку (рис 1.1, 1.2)

Крок 2:

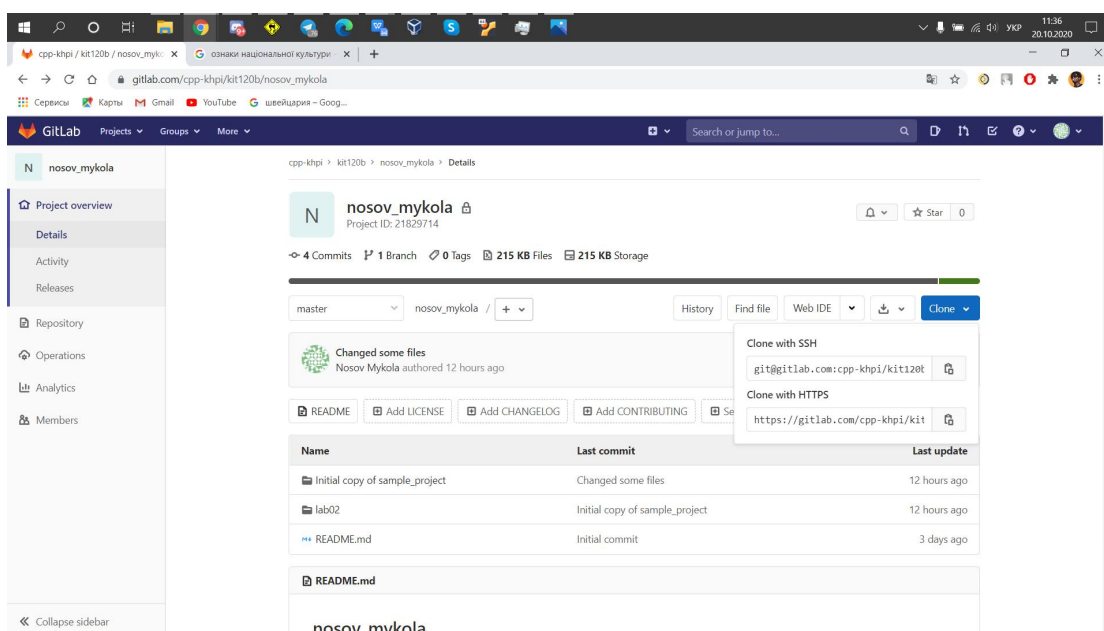
Склонувати проєкт з першої лабораторної роботи та додати новий клас "Людина" (рис 1.3, 1.4, 1.5, 1.6)

Крок 3:

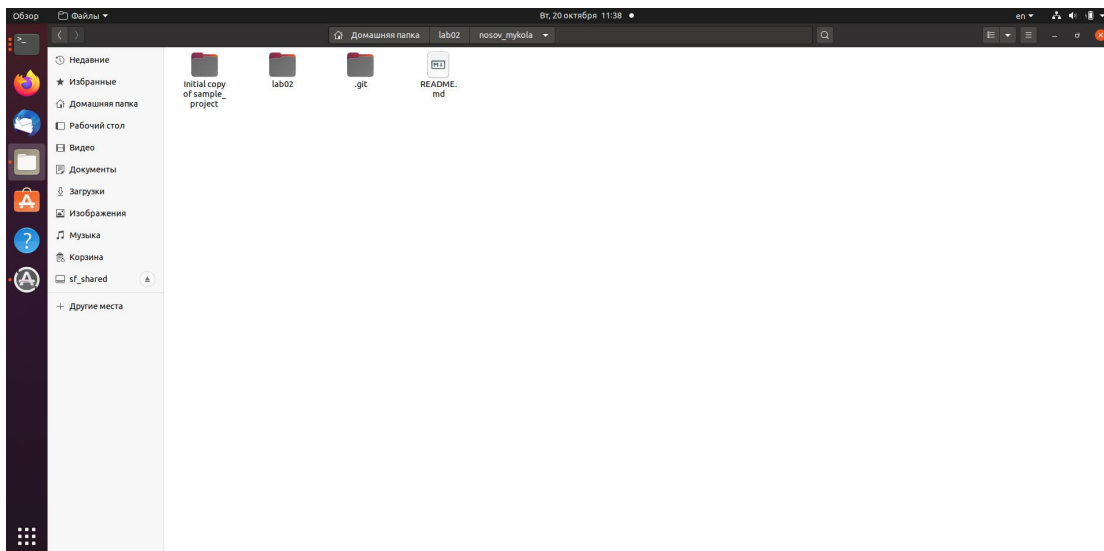
відкрити дебагер та зафіксувати значення змінних (рис 1.7, 1.8, 1.9)

Крок 4:

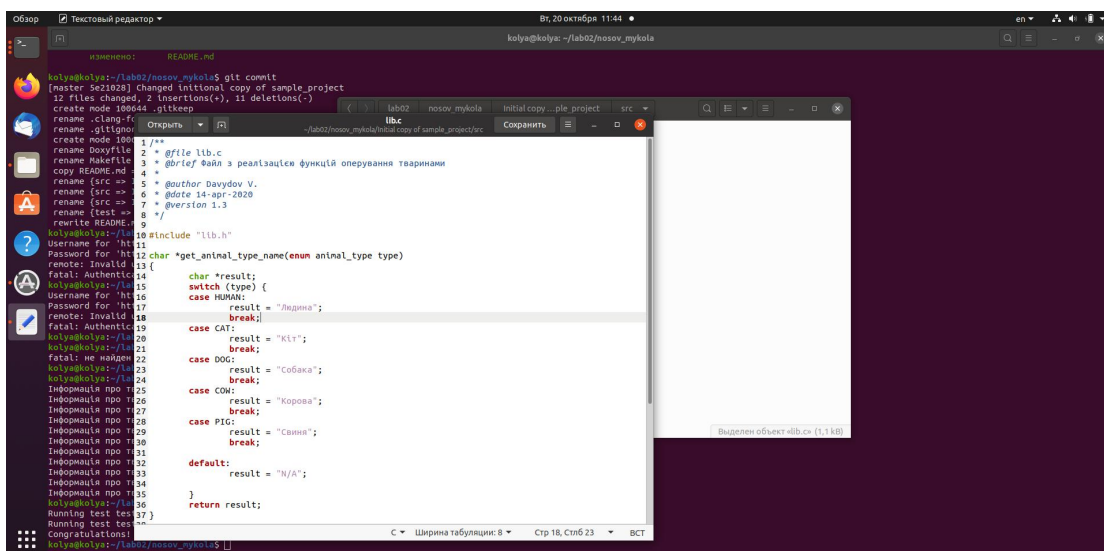
запустити проєкт (рис 1.10)



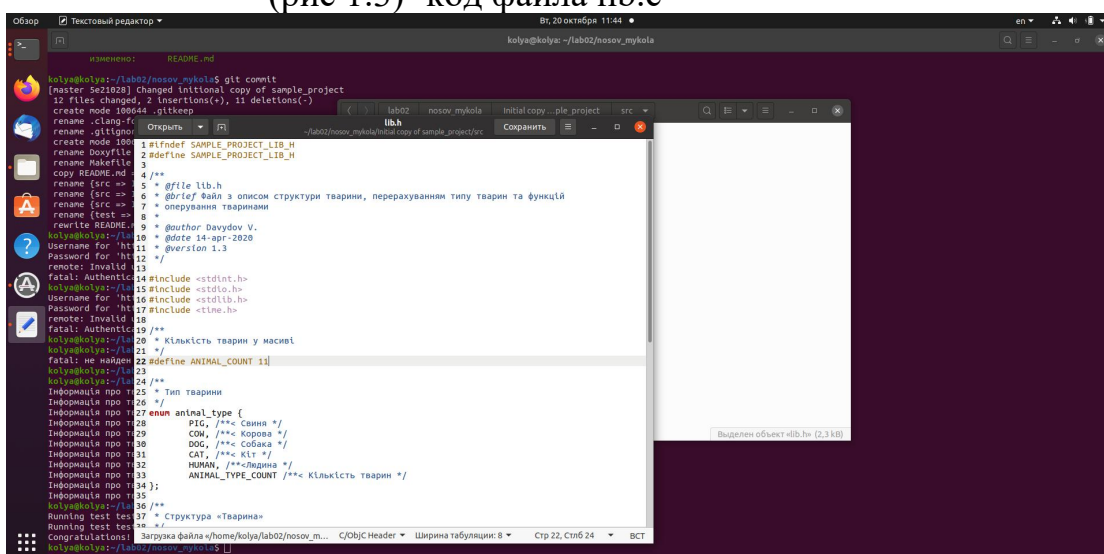
(рис 1.1) - облачна дерикторія



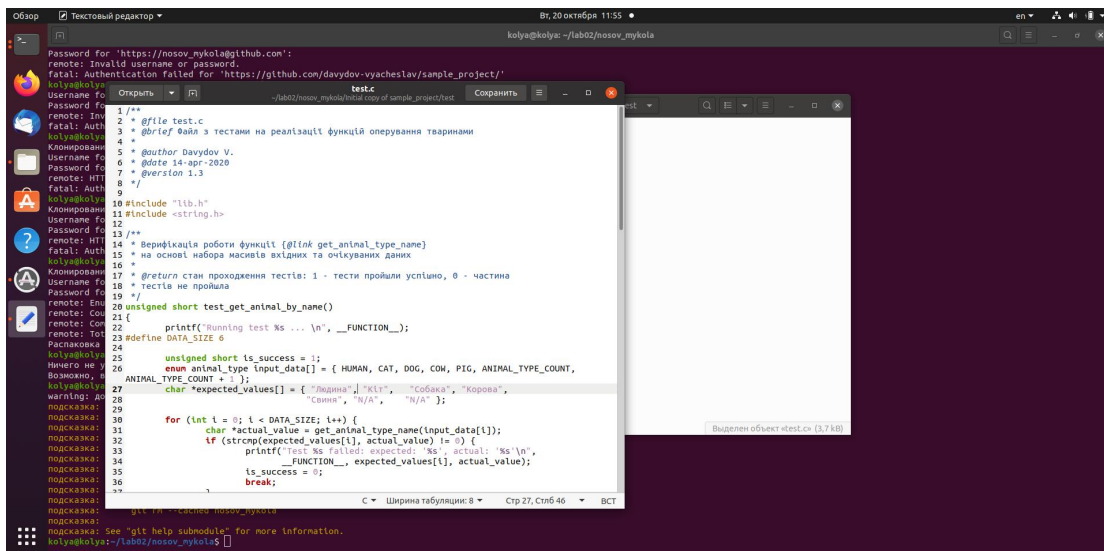
(рис 1.2)- репозиторія на моєму комп'ютері



(рис 1.3) -код файла lib.c

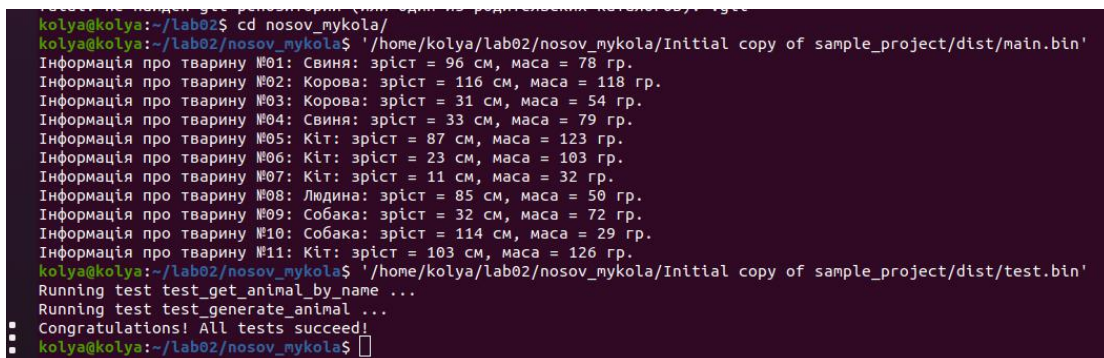


(рис 1.4) -код файла lib.h



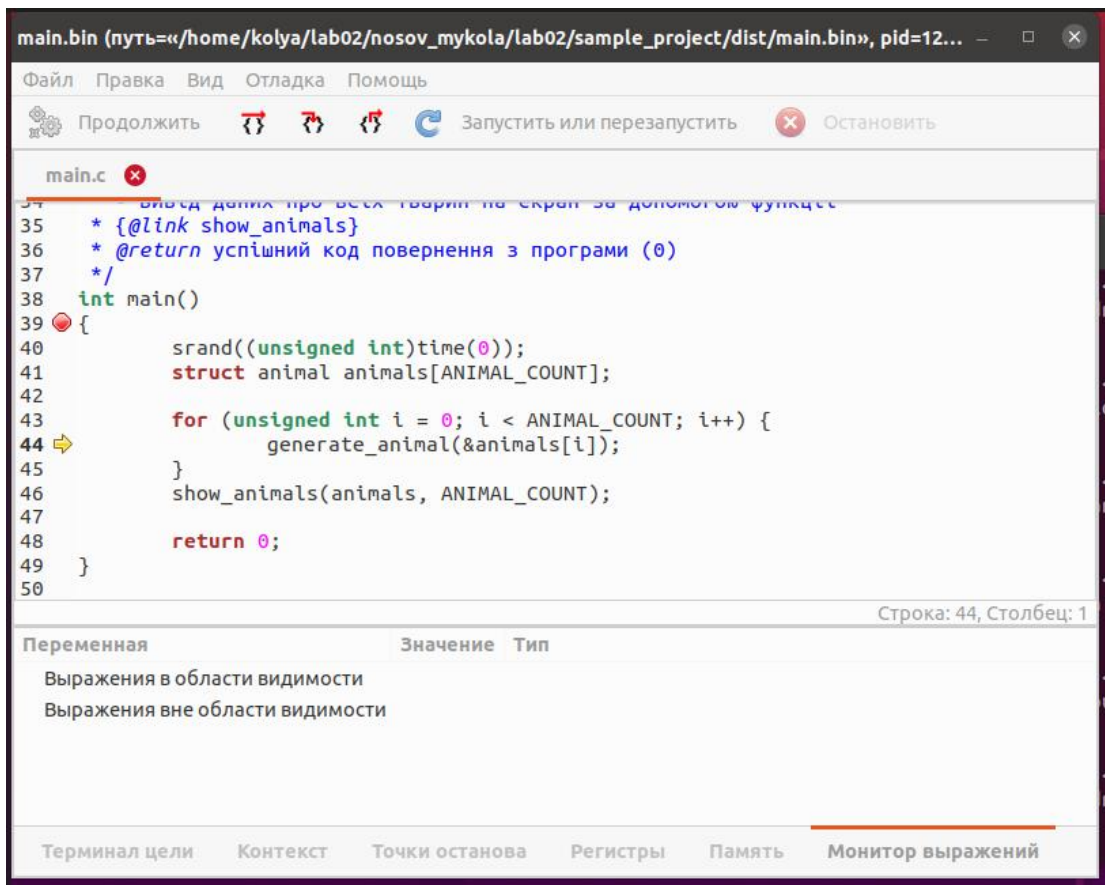
```
test.c
1  /**
2   * @file test.c
3   * @brief Файл з тестами на реалізаці функції оперування тваринами
4   *
5   * @author Davydov V.
6   * @date 14-apr-2020
7   * @version 1.3
8   */
9
10 #include "lib.h"
11 #include <string.h>
12
13 /**
14  * Верифікація роботи функції (getline_get_animal_type_name)
15  * на основі набору ксисів анімал та очікуваних даних
16  *
17  * @return стан проходження тестів: 1 - тести пройшли успішно, 0 - частина
18  * тестів не пройшла
19  */
20 unsigned short test_get_animal_by_name()
21 {
22     printf("Running test №s ... \n", __FUNCTION__);
23     #define DATA_SIZE 6
24     unsigned short ts_success = 1;
25     enum animal_type input_data[] = { HUMAN, CAT, DOG, COW, PIG, ANIMAL_TYPE_COUNT,
26     ANIMAL_TYPE_COUNT + 1 };
27     char *expected_values[] = { "Людина", "Кіт", "Собака", "Корова",
28     "Свиня", "N/A", "N/A" };
29
30     for (int t = 0; t < DATA_SIZE; t++) {
31         char *actual_value = get_animal_type_name(input_data[t]);
32         if (strcmp(expected_values[t], actual_value) != 0) {
33             printf("Test №s failed: expected: '%s', actual: '%s'\n",
34                 __FUNCTION__, expected_values[t], actual_value);
35             ts_success = 0;
36             break;
37         }
38     }
39     return ts_success;
40 }
```

(рис 1.5) - код файла test.c

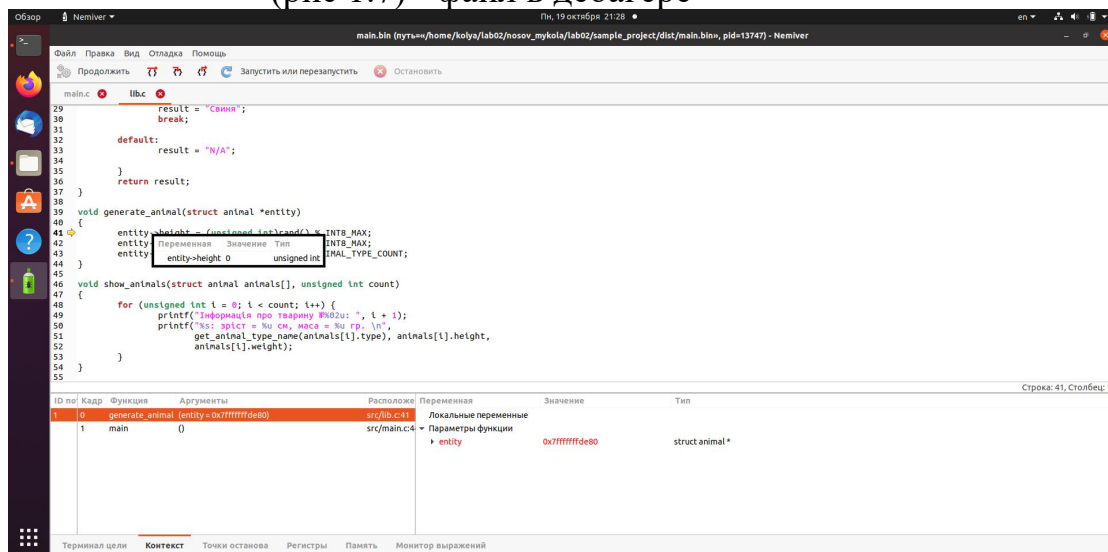


```
kolya@kolya:~/lab02$ cd nosov_mykola/
kolya@kolya:~/lab02/nosov_mykola$ ./home/kolya/lab02/nosov_mykola/Initial copy of sample_project/dist/main.bin
Інформація про тварину №01: Свиня: зріст = 96 см, маса = 78 гр.
Інформація про тварину №02: Корова: зріст = 116 см, маса = 118 гр.
Інформація про тварину №03: Корова: зріст = 31 см, маса = 54 гр.
Інформація про тварину №04: Свиня: зріст = 33 см, маса = 79 гр.
Інформація про тварину №05: Кіт: зріст = 87 см, маса = 123 гр.
Інформація про тварину №06: Кіт: зріст = 23 см, маса = 103 гр.
Інформація про тварину №07: Кіт: зріст = 11 см, маса = 32 гр.
Інформація про тварину №08: Людина: зріст = 85 см, маса = 50 гр.
Інформація про тварину №09: Собака: зріст = 32 см, маса = 72 гр.
Інформація про тварину №10: Собака: зріст = 114 см, маса = 29 гр.
Інформація про тварину №11: Кіт: зріст = 103 см, маса = 126 гр.
kolya@kolya:~/lab02/nosov_mykola$ ./home/kolya/lab02/nosov_mykola/Initial copy of sample_project/dist/test.bin
Running test test_get_animal_by_name ...
Running test test_generate_animal ...
Congratulations! All tests succeed!
kolya@kolya:~/lab02/nosov_mykola$
```

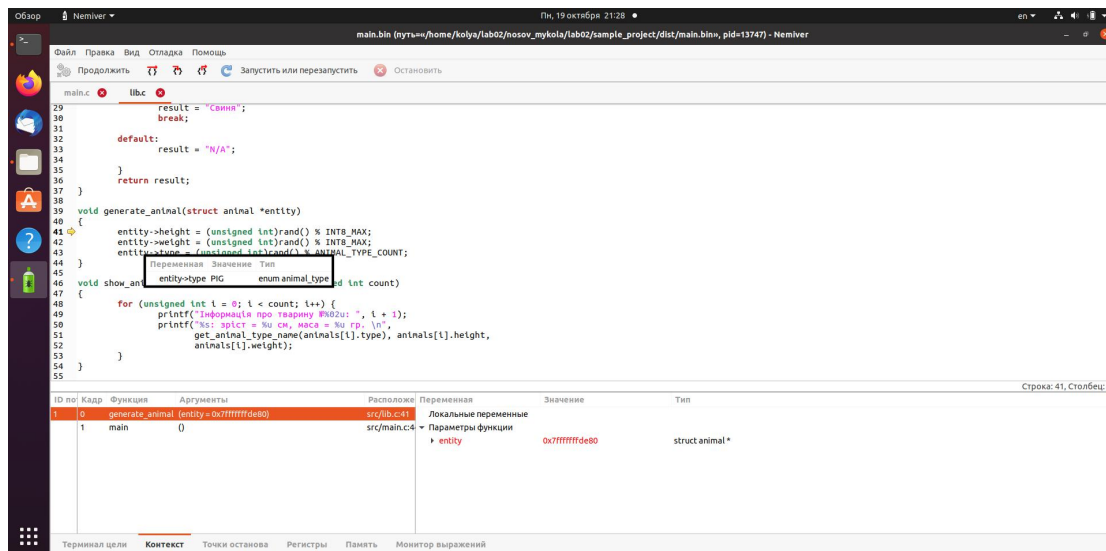
(рис 1.6)-тест main.bin, test.bin



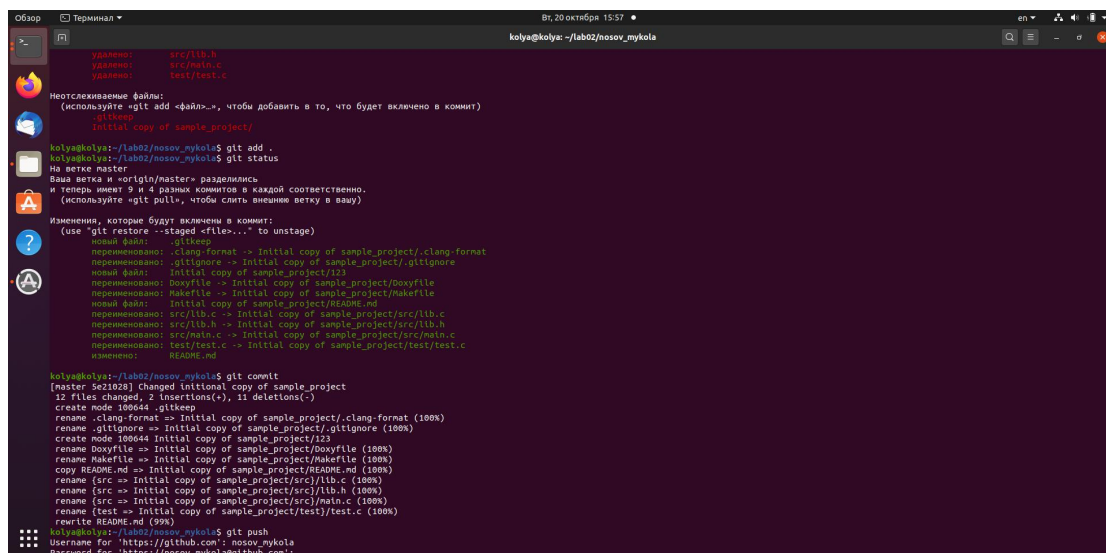
(рис 1.7) - файл в дебагере



(рис 1.8) - фіксації змін змінних



(рис 1.9) - фіксації змін



(рис1.10) - git push

Опис різниці між режимами Debug та Release:

Debug використовується для відладки програми. Цей режим вимикає настройки оптимізації та вмикає настройки відладки, що робить програму повільнішою, але полегшує процес відладки.

Release використовується для оптимізації програми для подальшого її випуску. Цей режим оптимізує програму по розміру та продуктивності.

Висновок:

у ході даної лабораторної роботи ми познайомились з особливостями роботи дебагера, а також додали новий клас до проєкту та загрузили його на зовнішній репозиторій