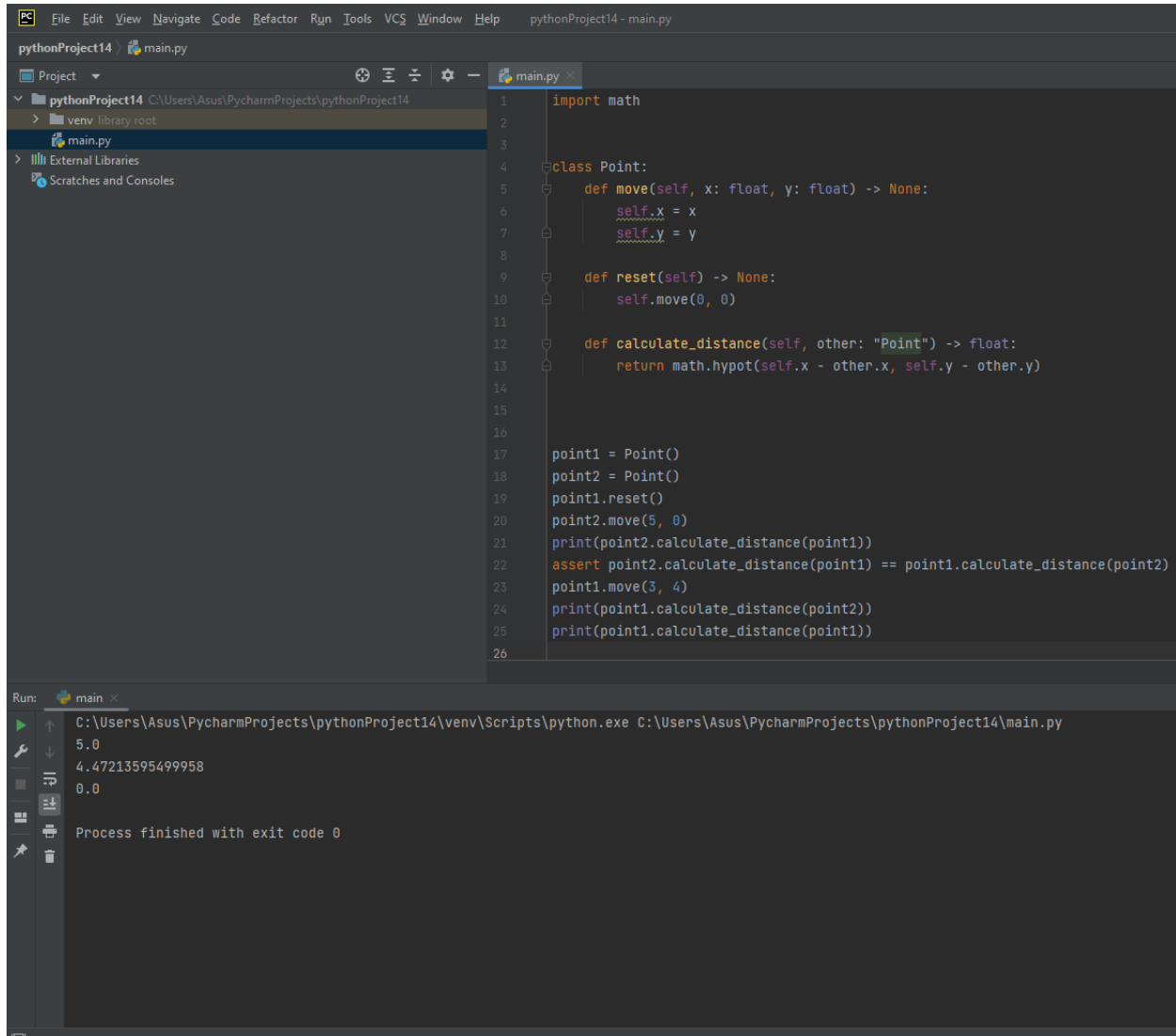


1. I wrote code in PyCharm and code was run.

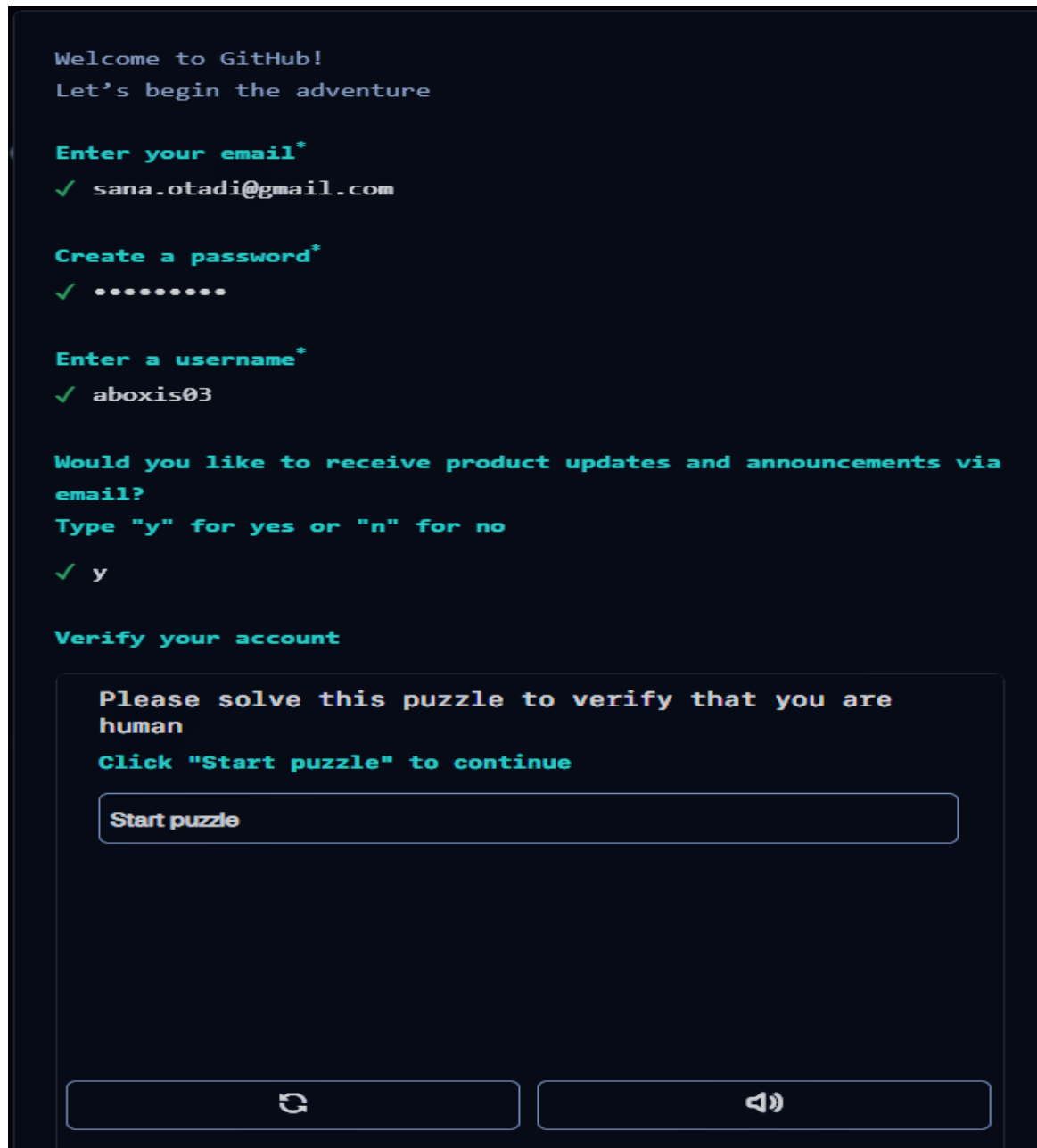


The screenshot displays the PyCharm IDE interface. The top toolbar includes menus for File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The project pane on the left shows the project structure for 'pythonProject14', including a 'venv' directory and the 'main.py' file. The main editor window shows the code in 'main.py' with line numbers 1 through 26. The code defines a 'Point' class with methods 'move', 'reset', and 'calculate_distance'. It then creates two 'Point' objects, 'point1' and 'point2', and performs several operations on them, including an assertion. The bottom pane shows the 'Run' output, indicating the command executed and the resulting output values: 5.0, 4.47213595499958, and 0.0. The process finished with exit code 0.

```
1 import math
2
3
4 class Point:
5     def move(self, x: float, y: float) -> None:
6         self.x = x
7         self.y = y
8
9     def reset(self) -> None:
10        self.move(0, 0)
11
12    def calculate_distance(self, other: "Point") -> float:
13        return math.hypot(self.x - other.x, self.y - other.y)
14
15
16
17 point1 = Point()
18 point2 = Point()
19 point1.reset()
20 point2.move(5, 0)
21 print(point2.calculate_distance(point1))
22 assert point2.calculate_distance(point1) == point1.calculate_distance(point2)
23 point1.move(3, 4)
24 print(point1.calculate_distance(point2))
25 print(point1.calculate_distance(point1))
26
```

Run: main x
C:\Users\Asus\PycharmProjects\pythonProject14\venv\Scripts\python.exe C:\Users\Asus\PycharmProjects\pythonProject14\main.py
5.0
4.47213595499958
0.0
Process finished with exit code 0

2.1. I go to GitHub.com and go to sign up:



Welcome to GitHub!
Let's begin the adventure

Enter your email*
✓ sana.otadi@gmail.com

Create a password*
✓

Enter a username*
✓ aboxis03

Would you like to receive product updates and announcements via email?
Type "y" for yes or "n" for no
✓ y

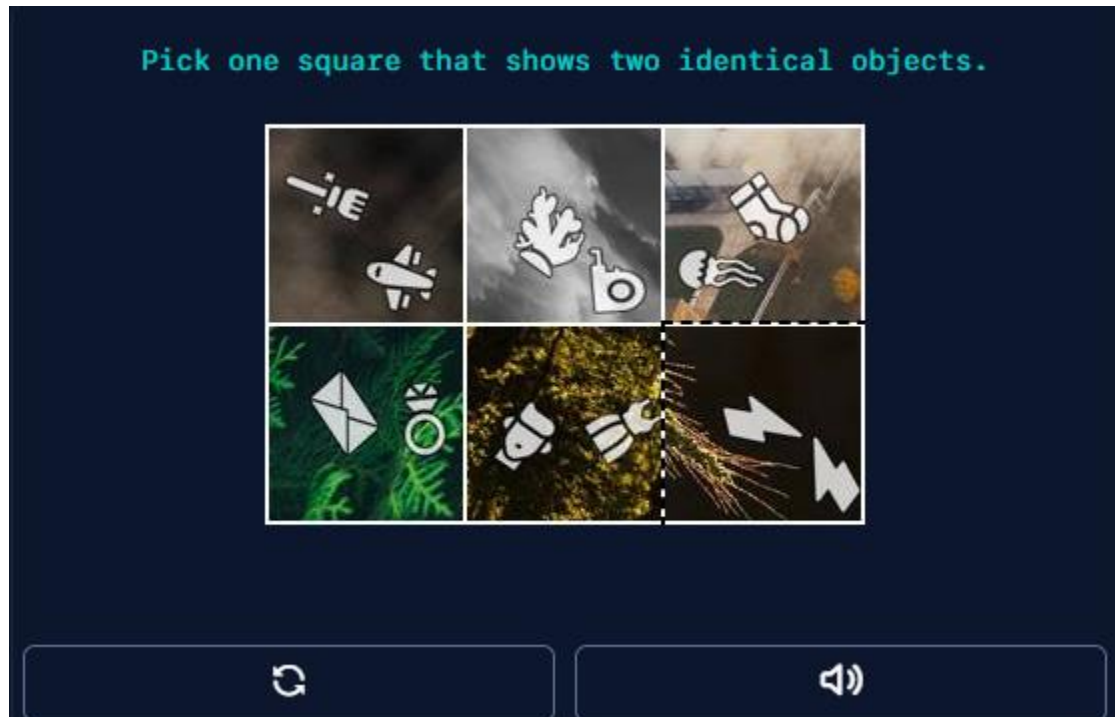
Verify your account

Please solve this puzzle to verify that you are human
Click "Start puzzle" to continue

Start puzzle

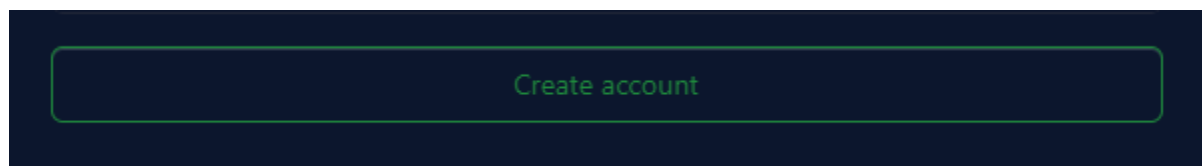
⏮ ⏭

Click on start puzzle

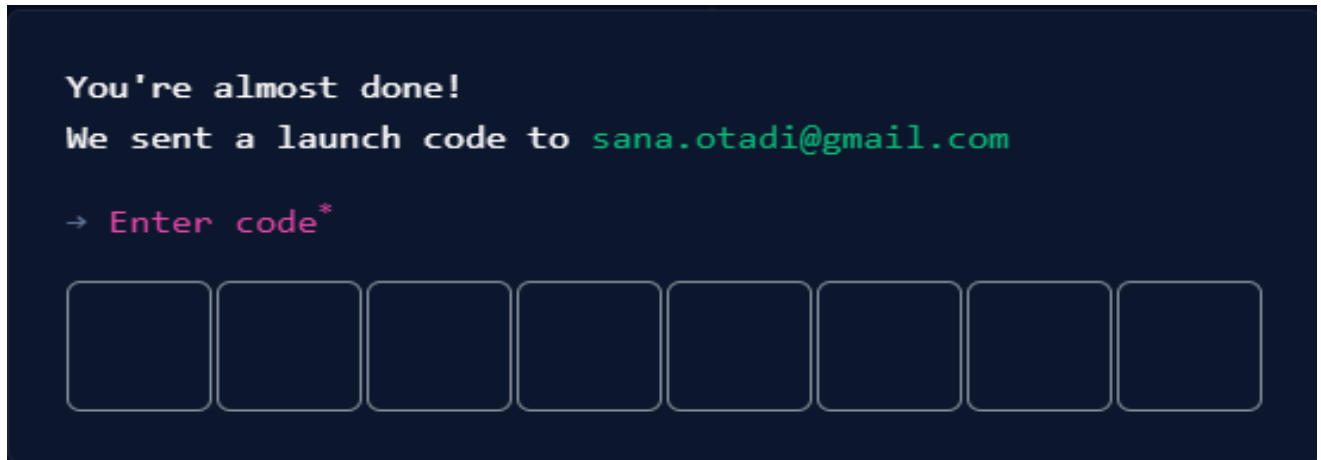


After clicking, according to the image, this puzzle started and had 2 stages.

After finishing the puzzle I clicked on **Create account**.



2.2. I entered the sent code inside the box



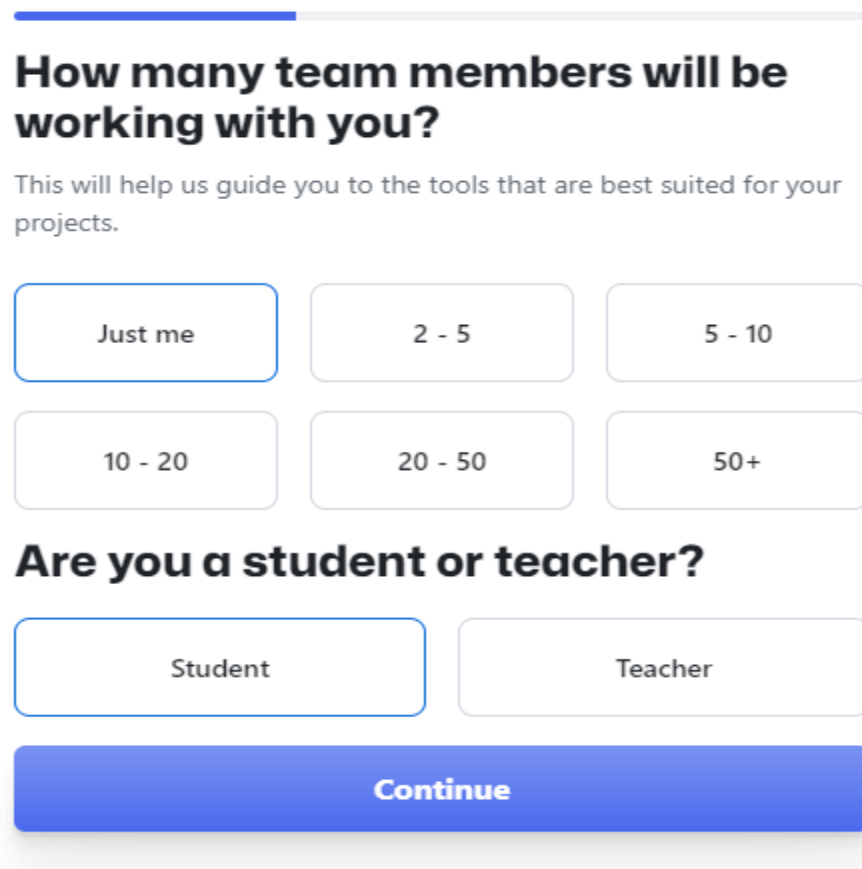
You're almost done!

We sent a launch code to `sana.otadi@gmail.com`

→ Enter code*

Eight empty input boxes for entering the code.

2.3. I selected **Just me** and **Student**.



How many team members will be working with you?

This will help us guide you to the tools that are best suited for your projects.

Just me 2 - 5 5 - 10

10 - 20 20 - 50 50+

Are you a student or teacher?








Student Teacher

Continue

2.4. I selected Project **Management**.

What specific features are you interested in using?

Select all that apply so we can point you to the right GitHub plan.

- ☐  Collaborative coding
Codespaces, Pull requests, Notifications, Code review, Code review assignments, Code owners, Draft pull requests, Protected branches, and more.
- ☐  Automation and CI/CD
Actions, Packages, APIs, GitHub Pages, GitHub Marketplace, Webhooks, Hosted runners, Self-hosted runners, Secrets management, and more.
- ☐  Security
Private repos, 2FA, Required reviews, Required status checks, Code scanning, Secret scanning, Dependency graph, Dependabot alerts, and more.
- ☐  Client Apps
GitHub Mobile, GitHub CLI, and GitHub Desktop.
- ☒  Project Management
Projects, Labels, Milestones, Issues, Unified Contribution Graph, Org activity graph, Org dependency insights, Repo insights, Wikis, and GitHub Insights.
- ☐  Team Administration
Organizations, Invitations, Team sync, Custom roles, Domain verification, Audit Log API, Repo creation restriction, and Notification restriction.
- ☐  Community
GitHub Marketplace, GitHub Sponsors, GitHub Skills, and Electron.

and the registration process is completed.

3.1. I clicked on plus in right corner and selected **new repository**.

The screenshot shows the GitHub 'Create a new repository' page. A red arrow labeled '1' points to the '+' icon in the top right corner. A blue arrow labeled '2' points to the 'Repository name' input field. A yellow arrow labeled '3' points to the 'Create repository' button at the bottom.

Pull requests Issues Codespaces Marketplace Explore

1

New repository
Import repository
New codespace
New gist
New organization
New project

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?
[Import a repository.](#)

Owner ^{*} Repository name ^{*}

Aboxis /

2

Great repository names are short and memorable. Need inspiration? How about [musical-broccoli?](#)

Description (optional)

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☐ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: **None**

Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)

License: **None**

You are creating a public repository in your personal account.

Create repository

3

3.2. I chose **Homework-1** for the name repository.

3.3. and clicked on **Create repository**.

3.4. I transferred the Python file to the repository with the help of git bash, according to the image below.

```
Asus@DESKTOP-RL8NDGT MINGW64 /e/Git (main)
$ git add myhomework.py

Asus@DESKTOP-RL8NDGT MINGW64 /e/Git (main)
$ git commit -m "first commit by Borhan"
[main (root-commit) f0e0e9a] first commit by Borhan
1 file changed, 27 insertions(+)
create mode 100644 myhomework.py

Asus@DESKTOP-RL8NDGT MINGW64 /e/Git (main)
$ git log
commit f0e0e9a6a0e59d0213bff0ec89592064190c39ed (HEAD -> main)
Author: Aboxis <Borhanbadri2002@gmail.com>
Date: Thu Apr 20 17:52:25 2023 +0430

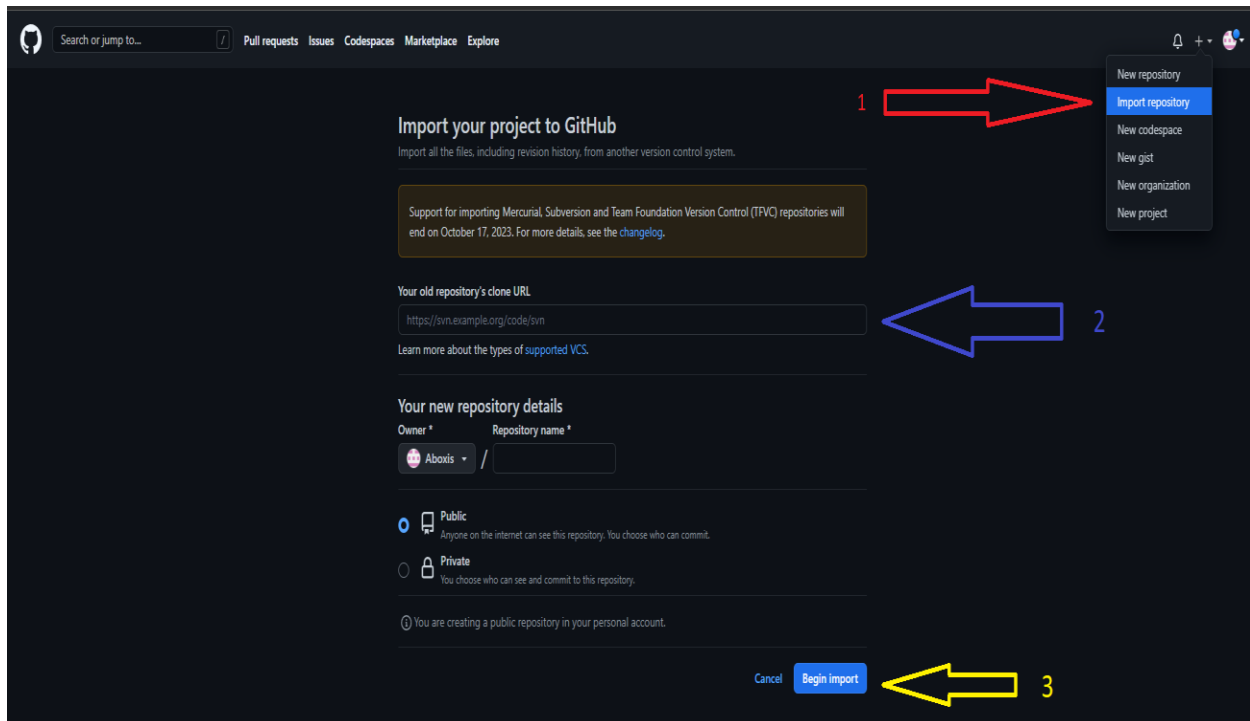
    first commit by Borhan

Asus@DESKTOP-RL8NDGT MINGW64 /e/Git (main)
$ git remote add origin https://github.com/Aboxis/Homework-1.git

Asus@DESKTOP-RL8NDGT MINGW64 /e/Git (main)
$ git branch -M main

Asus@DESKTOP-RL8NDGT MINGW64 /e/Git (main)
$ git push -u origin main
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% (3/3), 455 bytes | 455.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Aboxis/Homework-1.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

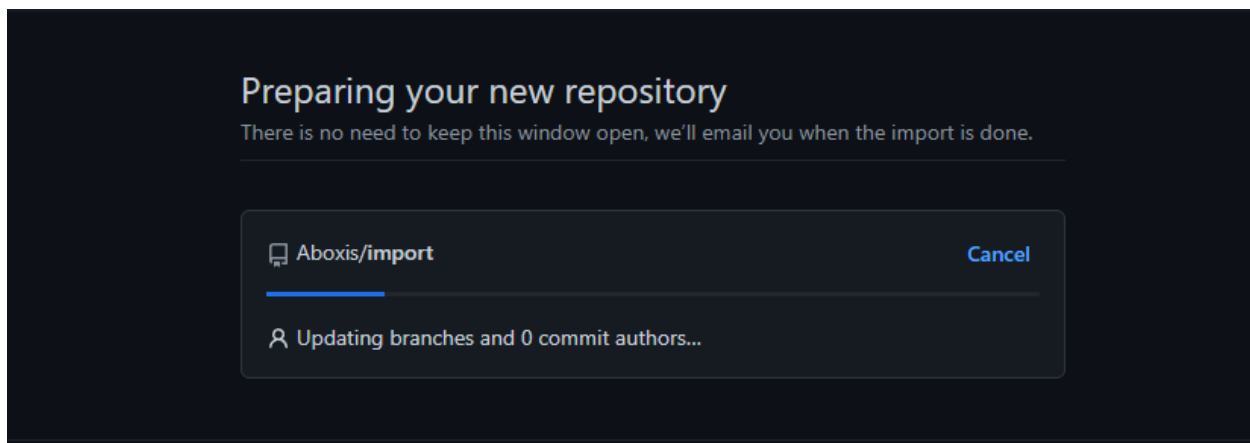
4. I copied the repository of book link from GitHub.



4.1. I clicked on plus in right corner and clicked **import repository**.

4.2. I pasted the copied link.

4.3. I click on **begin import**.



4.4. I waited for it to be imported.

4.5. I pull repository in my local repository according to image below.

```
Asus@DESKTOP-RL8NDGT MINGW64 ~
$ cd E:/import

Asus@DESKTOP-RL8NDGT MINGW64 /e/import (main)
$ ls -la
total 20
drwxr-xr-x 1 Asus 197121 0 Apr 20 18:35 ./
drwxr-xr-x 1 Asus 197121 0 Apr 20 18:33 ../
drwxr-xr-x 1 Asus 197121 0 Apr 20 18:35 .git/

Asus@DESKTOP-RL8NDGT MINGW64 /e/import (main)
$ git pull origin import
fatal: 'origin' does not appear to be a git repository
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.

Asus@DESKTOP-RL8NDGT MINGW64 /e/import (main)
$ git remote add origin https://github.com/Aboxis/import.git

Asus@DESKTOP-RL8NDGT MINGW64 /e/import (main)
$ git brnach -m main
git: 'brnach' is not a git command. See 'git --help'.

The most similar command is
    branch

Asus@DESKTOP-RL8NDGT MINGW64 /e/import (main)
$ git branch -m main

Asus@DESKTOP-RL8NDGT MINGW64 /e/import (main)
$ git pull add origin main
fatal: 'add' does not appear to be a git repository
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.

Asus@DESKTOP-RL8NDGT MINGW64 /e/import (main)
$ git log
fatal: your current branch 'main' does not have any commits yet

Asus@DESKTOP-RL8NDGT MINGW64 /e/import (main)
$ git commit
On branch main

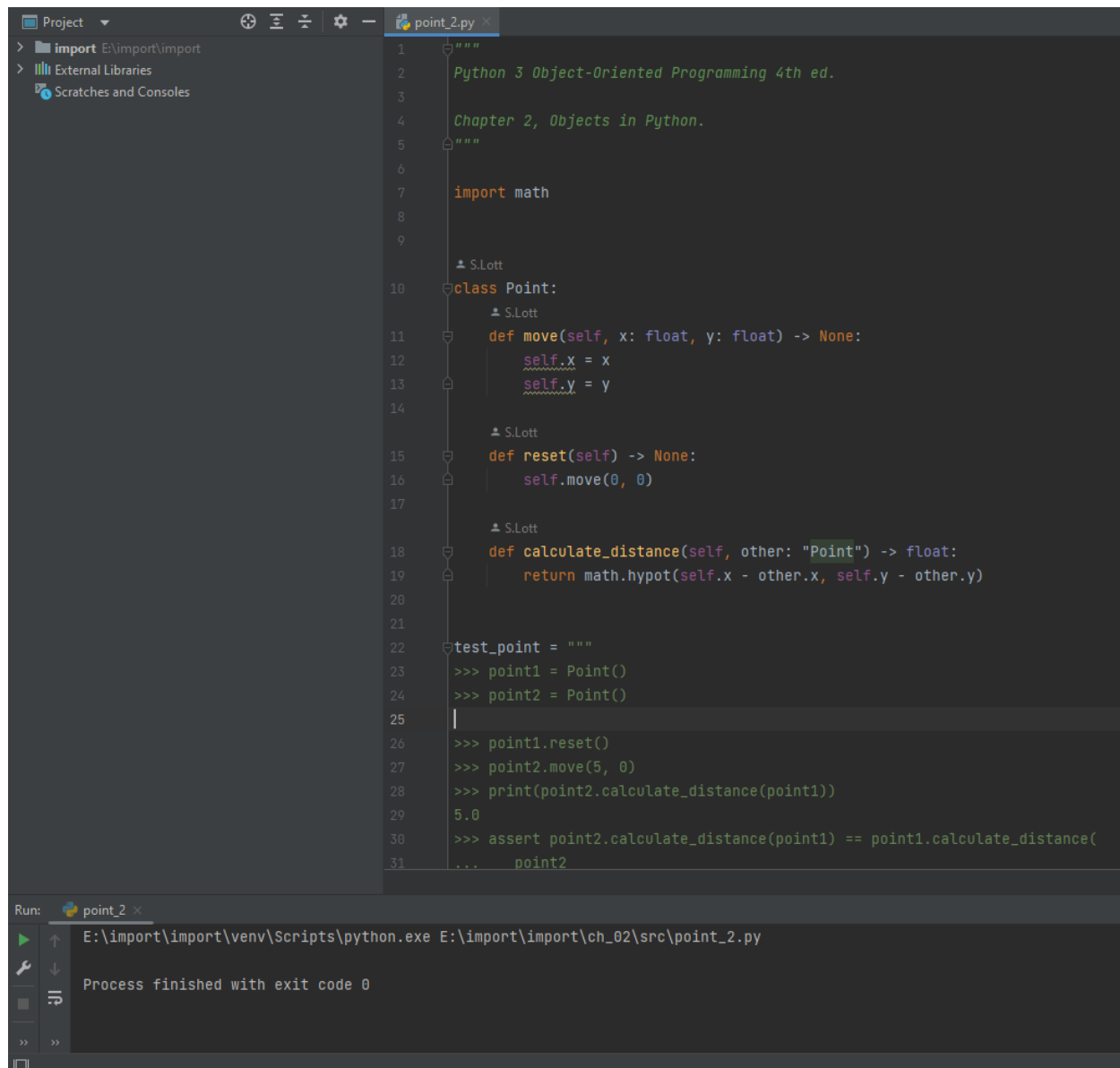
Initial commit

nothing to commit (create/copy files and use "git add" to track)

Asus@DESKTOP-RL8NDGT MINGW64 /e/import (main)
$ git clone https://github.com/Aboxis/import.git
Cloning into 'import'...
remote: Enumerating objects: 560, done.
remote: Counting objects: 100% (560/560), done.
remote: Compressing objects: 100% (431/431), done.
remote: Total 560 (delta 119), reused 560 (delta 119), pack-reused 0
Receiving objects: 100% (560/560), 18.59 MiB | 231.00 KiB/s, done.
Resolving deltas: 100% (119/119), done.
warning: the following paths have collided (e.g. case-sensitive paths
on a case-insensitive filesystem) and only one from the same
colliding group is in the working tree:

'bezdekIris.data'
'bezdekiris.data'
'bezdekIris.json'
'bezdekiris.json'
'bezdekIris.ndjson'
'bezdekiris.ndjson'
'bezdekIris.yaml'
'bezdekiris.yaml'
```

4.6. I opened project point_2 in my PyCharm and run it.



The image shows the PyCharm IDE interface. The left sidebar displays the project structure with folders 'import' and 'External Libraries'. The main editor window shows the file 'point_2.py' with the following code:

```
1  """
2  Python 3 Object-Oriented Programming 4th ed.
3
4  Chapter 2, Objects in Python.
5  """
6
7  import math
8
9
10 class Point:
11     def move(self, x: float, y: float) -> None:
12         self.x = x
13         self.y = y
14
15     def reset(self) -> None:
16         self.move(0, 0)
17
18     def calculate_distance(self, other: "Point") -> float:
19         return math.hypot(self.x - other.x, self.y - other.y)
20
21
22 test_point = """
23 >>> point1 = Point()
24 >>> point2 = Point()
25
26 >>> point1.reset()
27 >>> point2.move(5, 0)
28 >>> print(point2.calculate_distance(point1))
29 5.0
30 >>> assert point2.calculate_distance(point1) == point1.calculate_distance(
31 ...     point2
```

The bottom panel shows the 'Run' output for 'point_2'. The command executed is 'E:\import\import\venv\Scripts\python.exe E:\import\import\ch_02\src\point_2.py', and the output is 'Process finished with exit code 0'.