**LAPORAN PRAKTIKUM OOP DASAR**

**MODUL 6**

**ABSTRACTION**



**Adam Arthur Faizal**

**M3119001**

**TI A**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**SEKOLAH VOKASI**

**UNIVERSITAS SEBELAS MARET**

**SURAKARTA**

**2020**

**MODUL 6. Abstraksi**

**Capaian Pembelajaran Praktikum:**

* Menerapkan konsep abstraksi dalam program
* Menerapkan konstruktor berparameter
* Menerapkan teknik casting
* Accessing method at another class

**Tools:**

1. Java Development Kit (JDK)
2. Greenfoot IDE

**Terminologi:**

Isikan terminology yang sesuai untuk definisi dibawah ini:

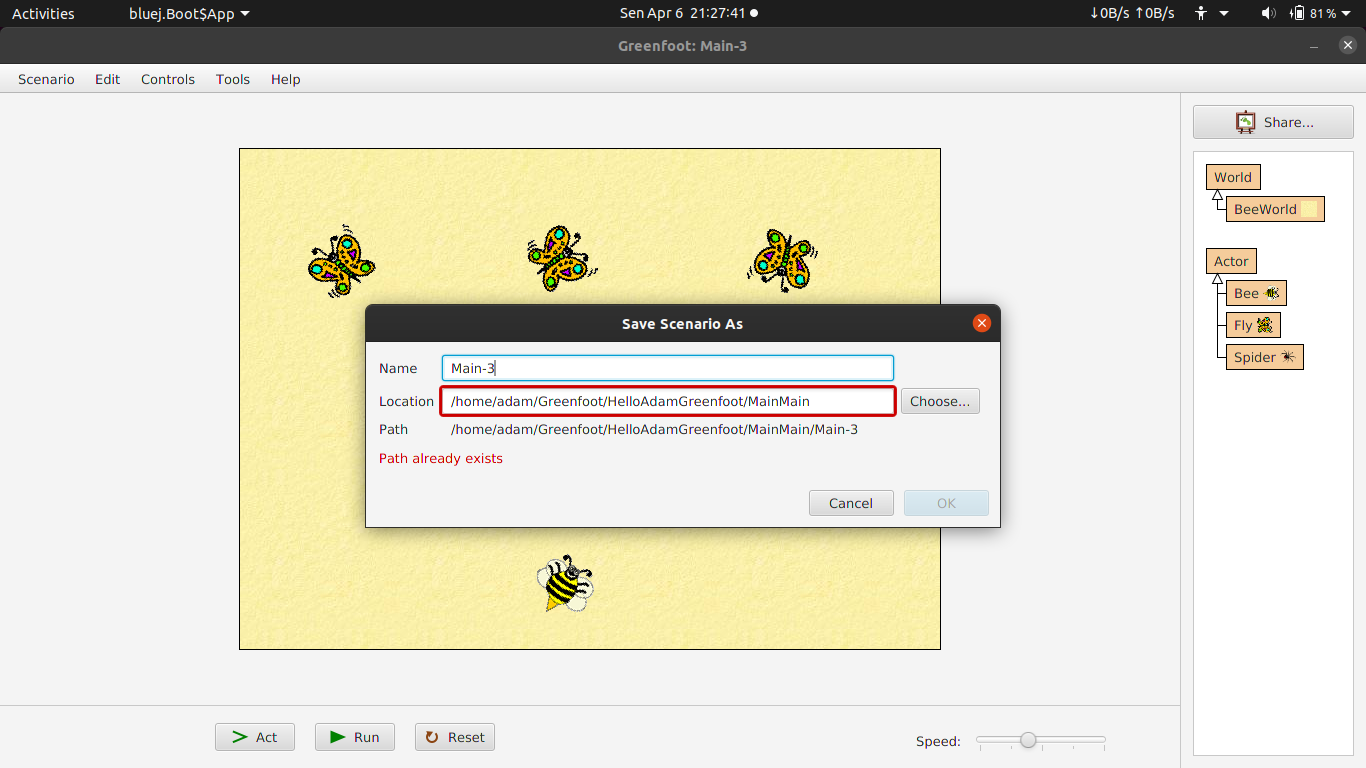
[Abstraction] A technique used to command newly-created instances to perform different actions.

[Casting] A technique used to tell java that a class is to be considered as another class

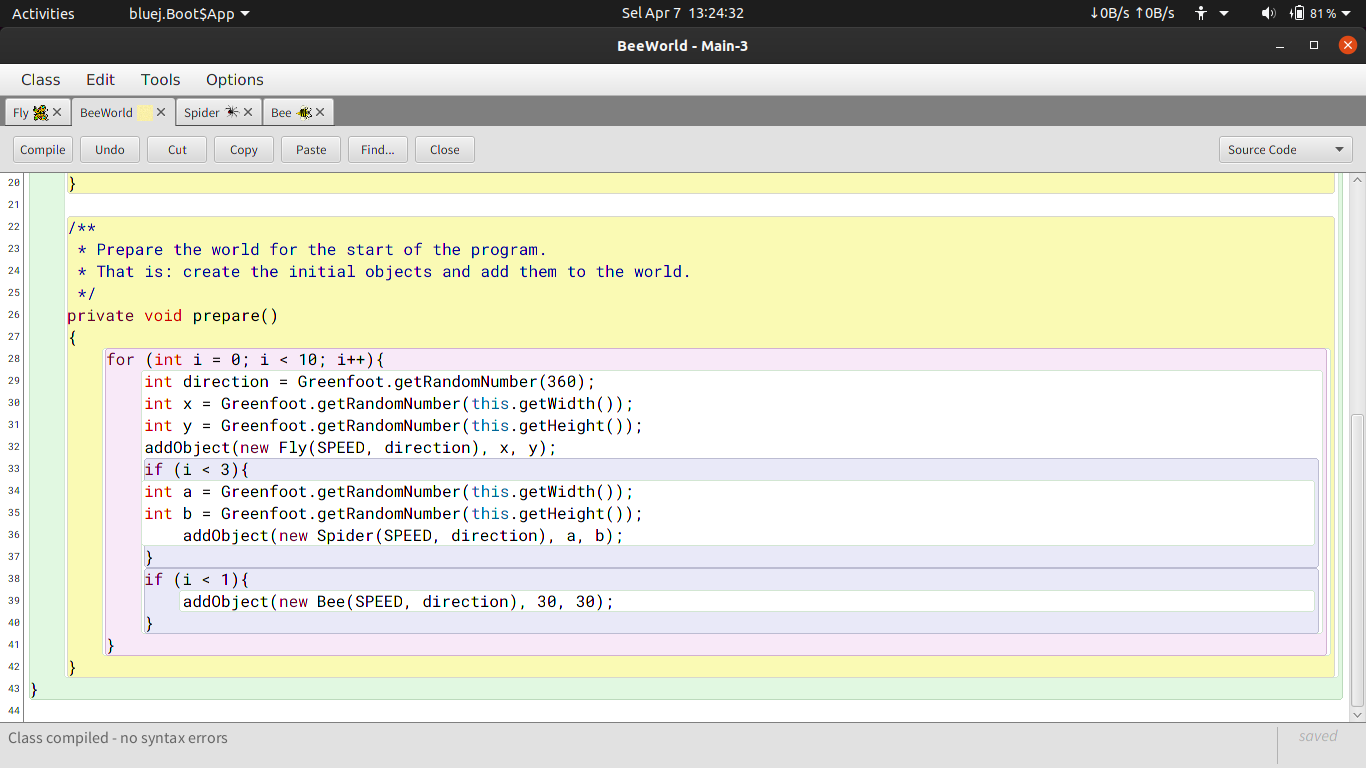
[Constructor] A special method that is executed automatically whenever a new instance of the class is created.

**TRY IT / SOLVE IT:**

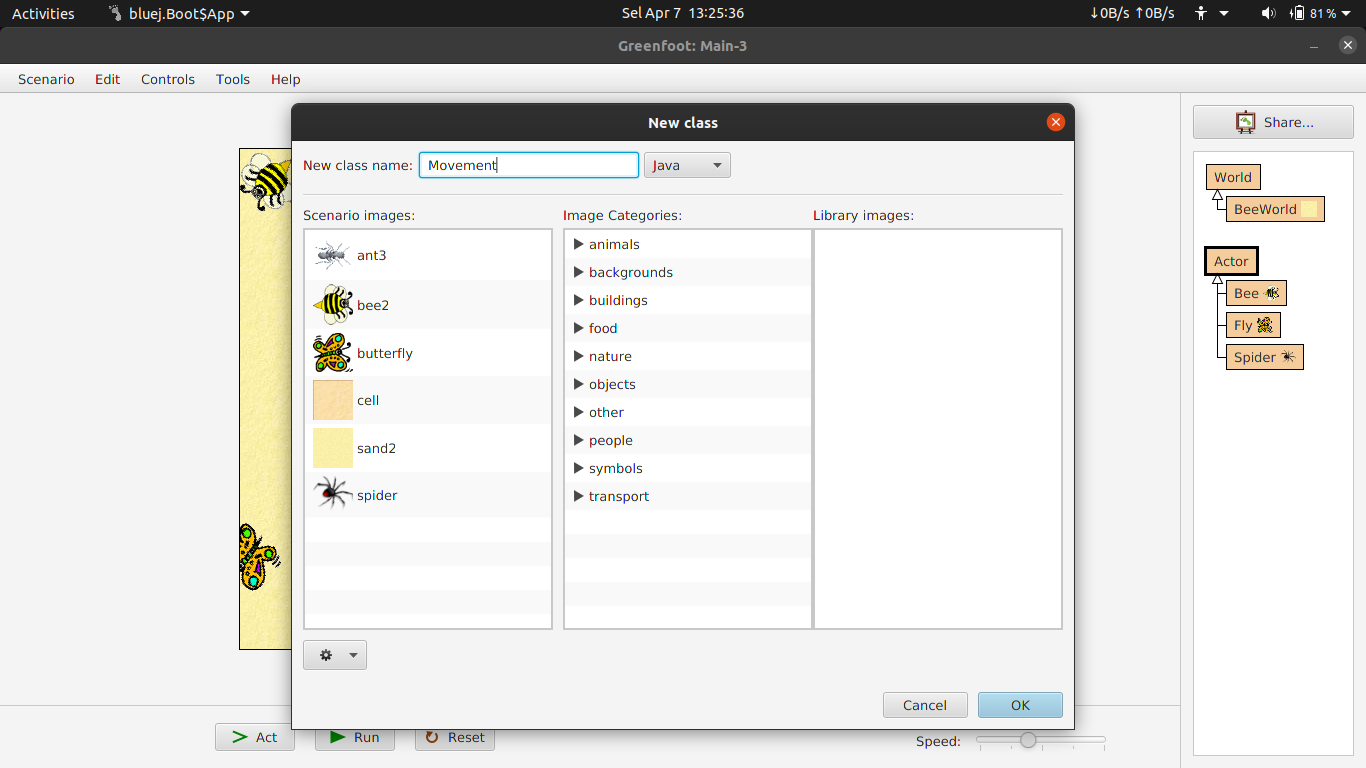
1. Buka scenario modul sebelumnya, kemudian lakukan versioning dengan melakukan save as. Beri nama folder yang baru untuk menyimpan versi scenario yang baru.



1. Dengan menggunakan abstraksi, modifikasi class BeeWorld dengan membuat method *prepare()* untuk menambahkan 10 objek *Fly()* ke dunia secara acak (X dan Y nya) dengan menggunakan perulangan, 1 objek *Bee()* pada lokasi 30,30, dan 3 objek *Spider()* secara acak dengan perulangan. Kemudian panggil method tersebut di konstruktor BeeWorld.



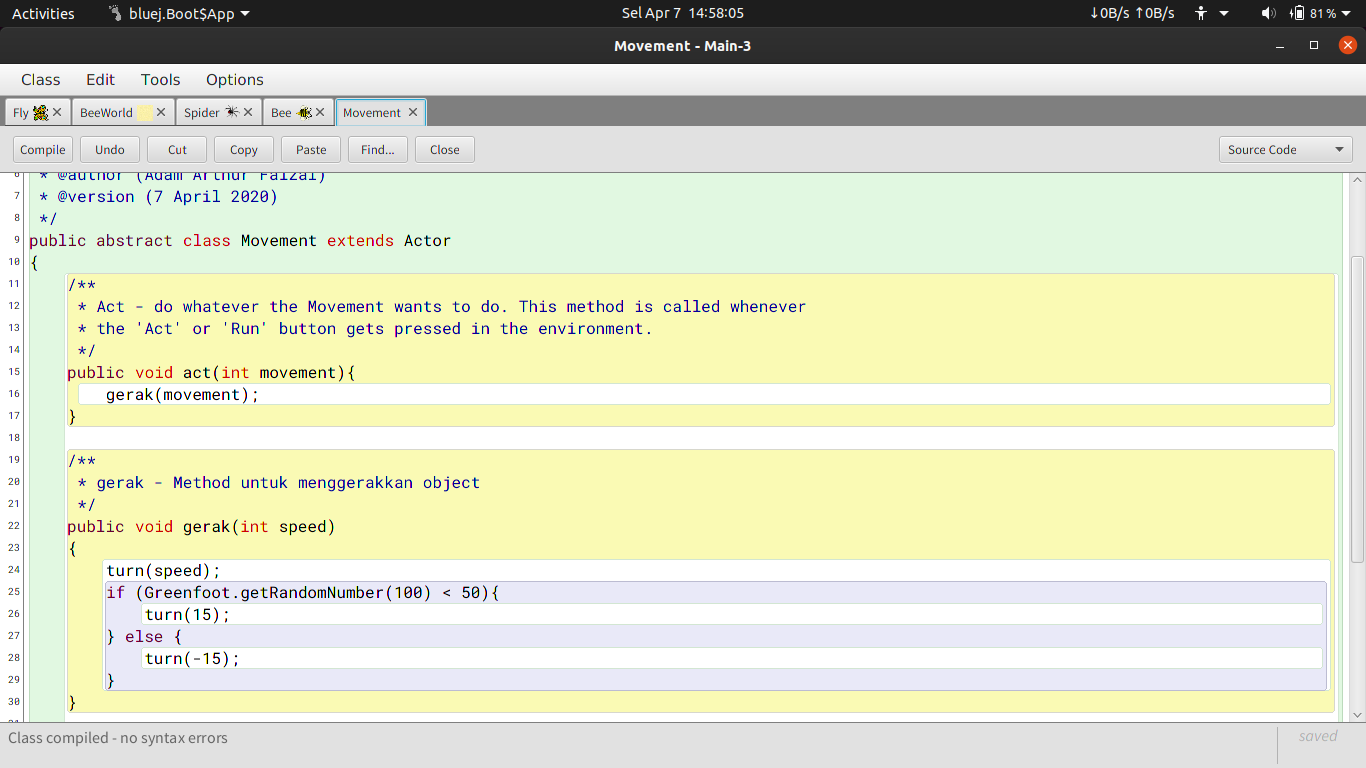
1. Buat class Movement sebagai sub class Actor. (tanpa image)



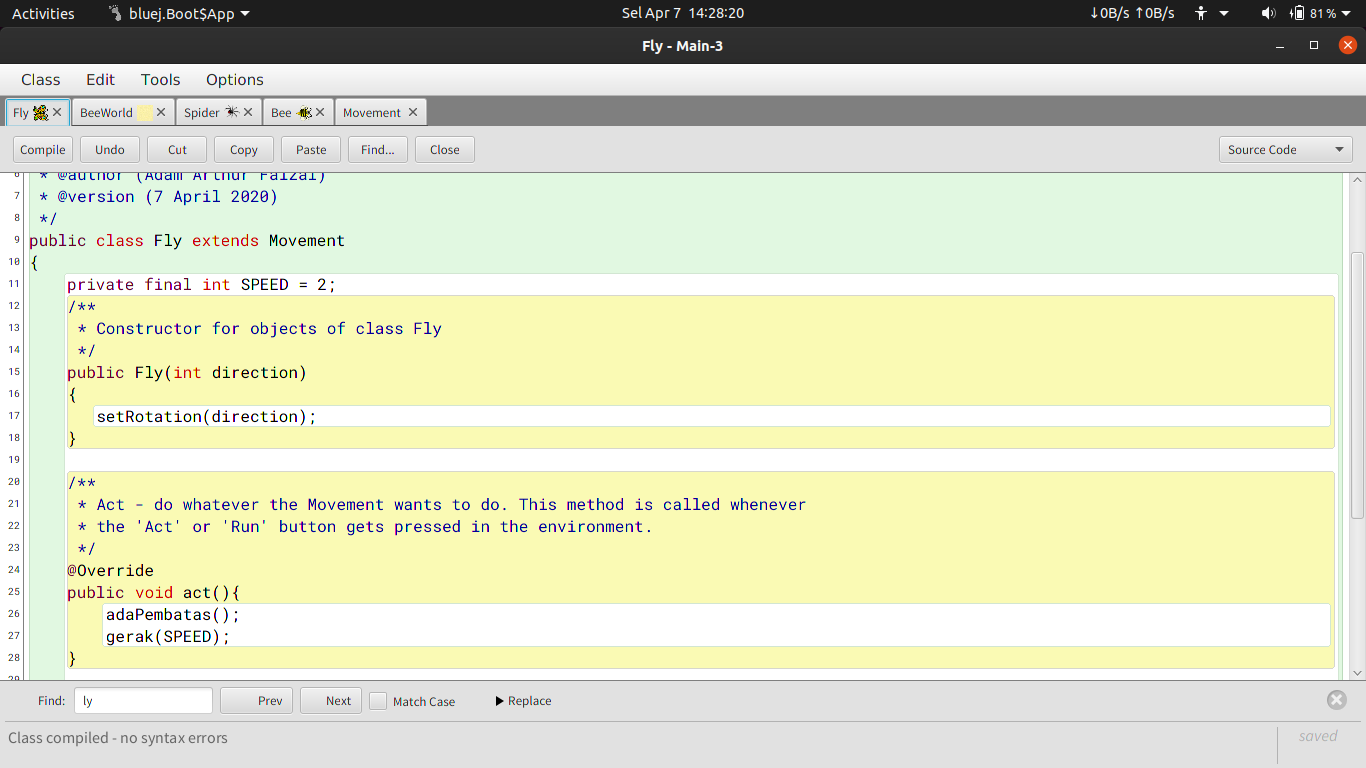
1. Modifikasi kelas Spider dan Fly, yang awalnya menginduk (extends) ke kelas Aktor, anda pindahkan menginduk pada kelas Movement.



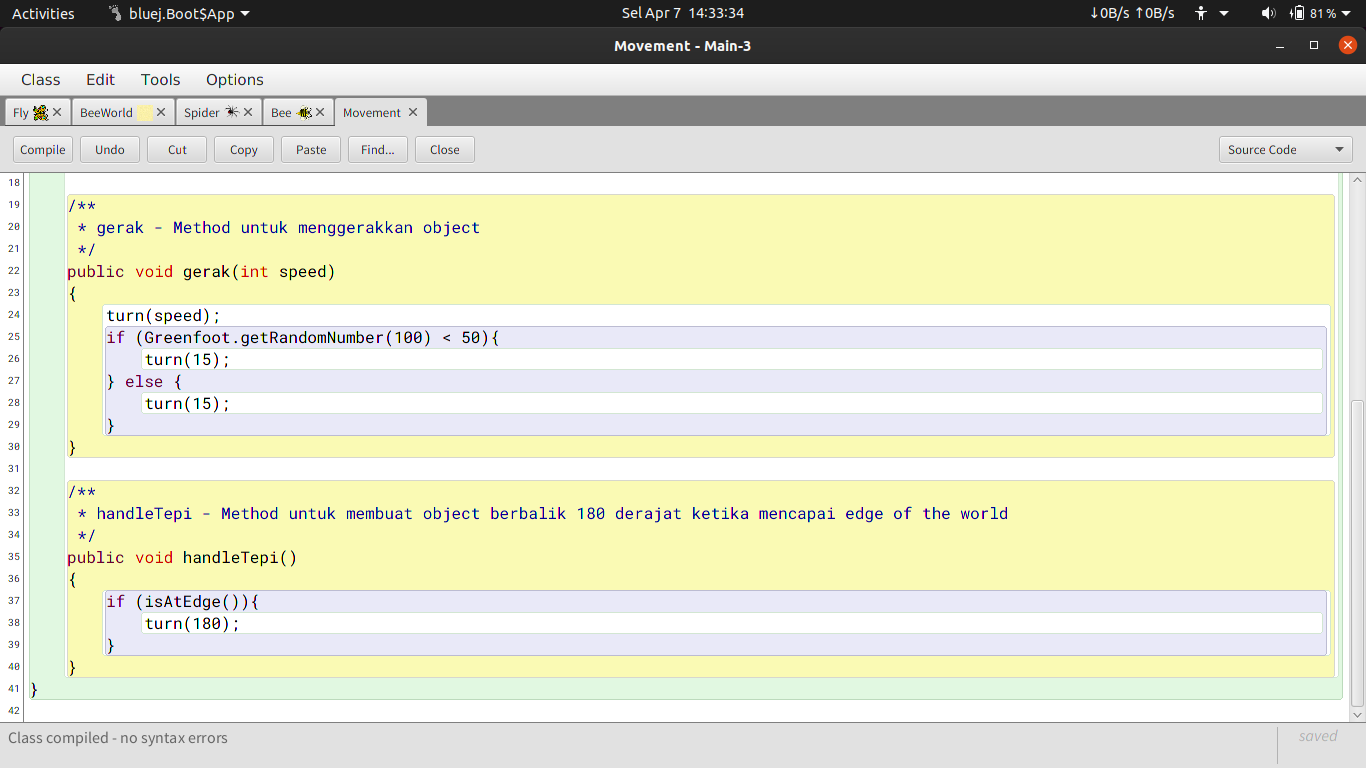
1. Buatlah method gerak() di kelas Movement yang akan mengganti gerak Spider dan Fly, dimana pergerakan dari objek Spider dan Fly adalah bergerak maju dengan kecepatan masing-masing (speed) dan berbelok ke kanan dan ke kiri sebesar 10 secara random dengan probabilitas 50:50.

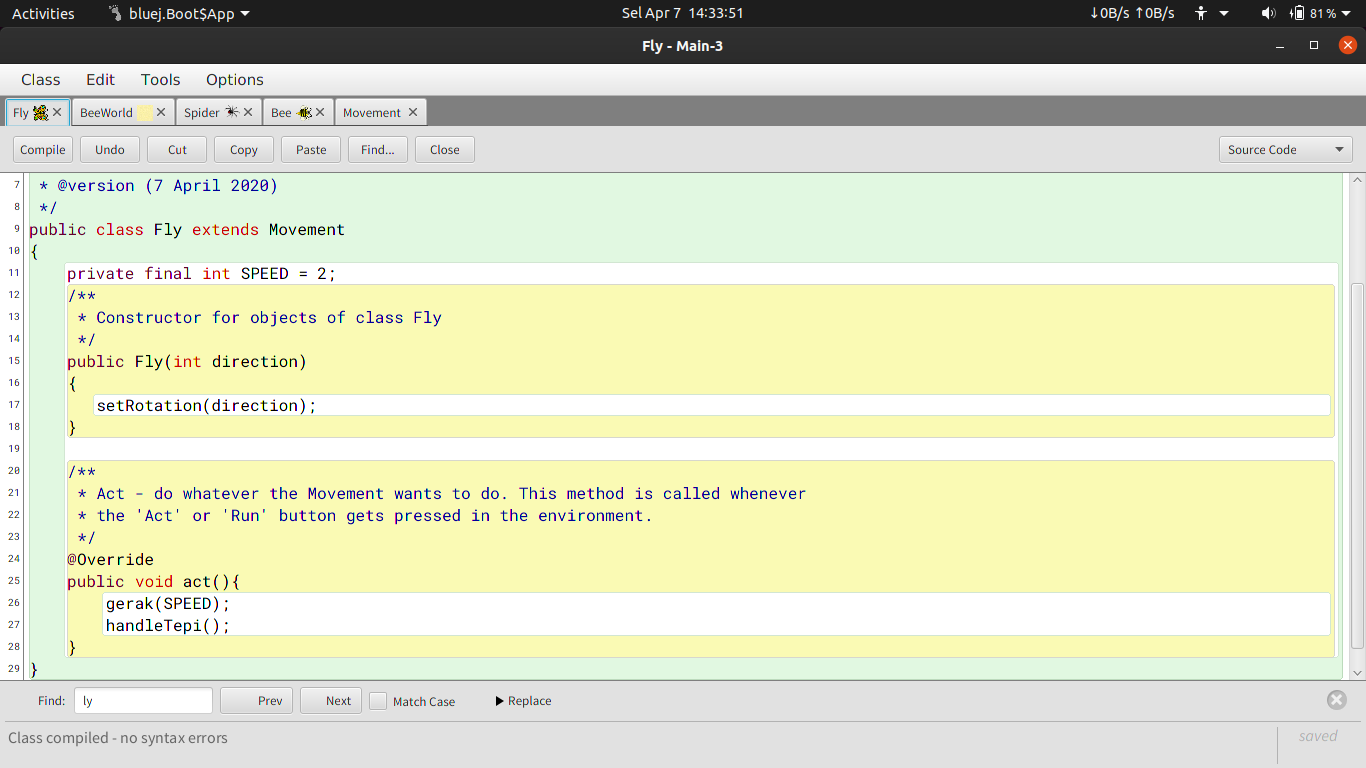
****

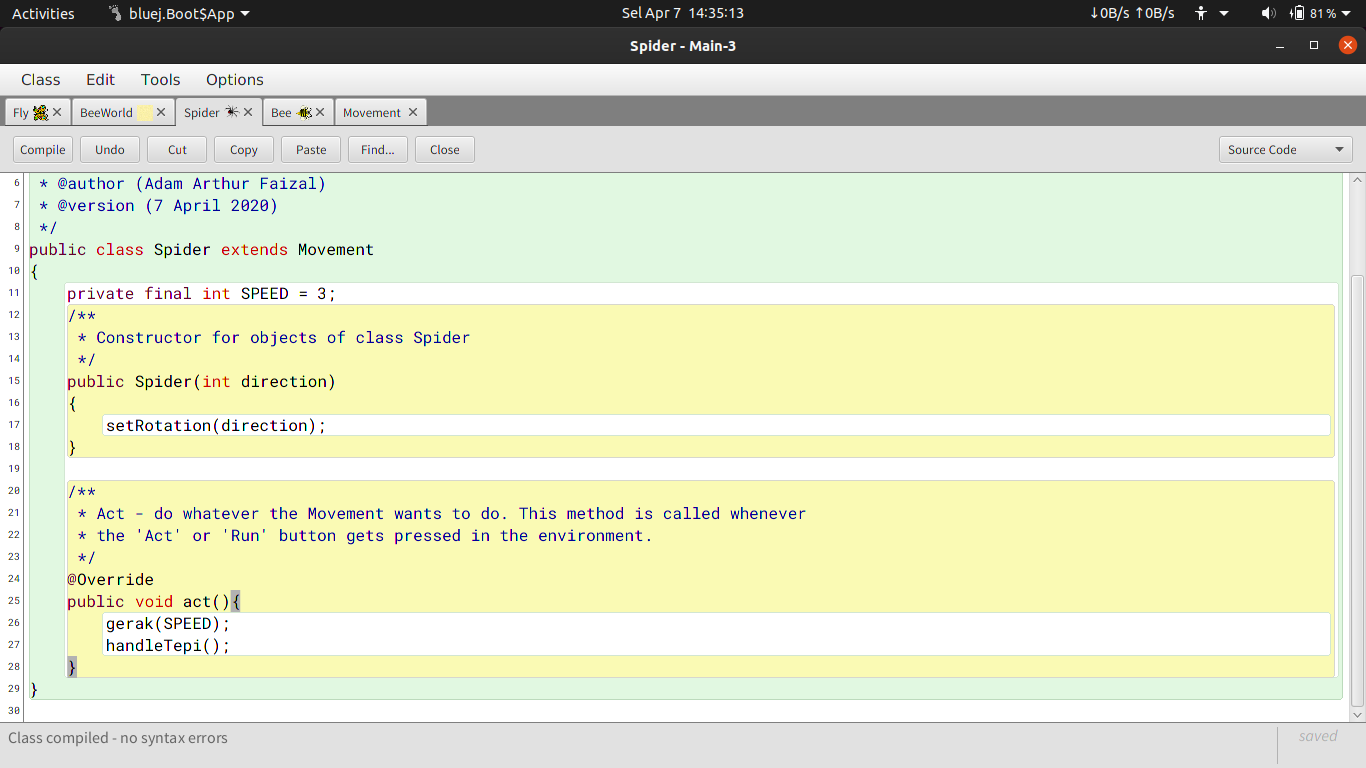
1. **Panggil / invoke** method gerak() di method act Spider maupun Fly. (Speed untuk Spider adalah 3 dan untuk Fly sebesar 2) – **Gunakan variabel**



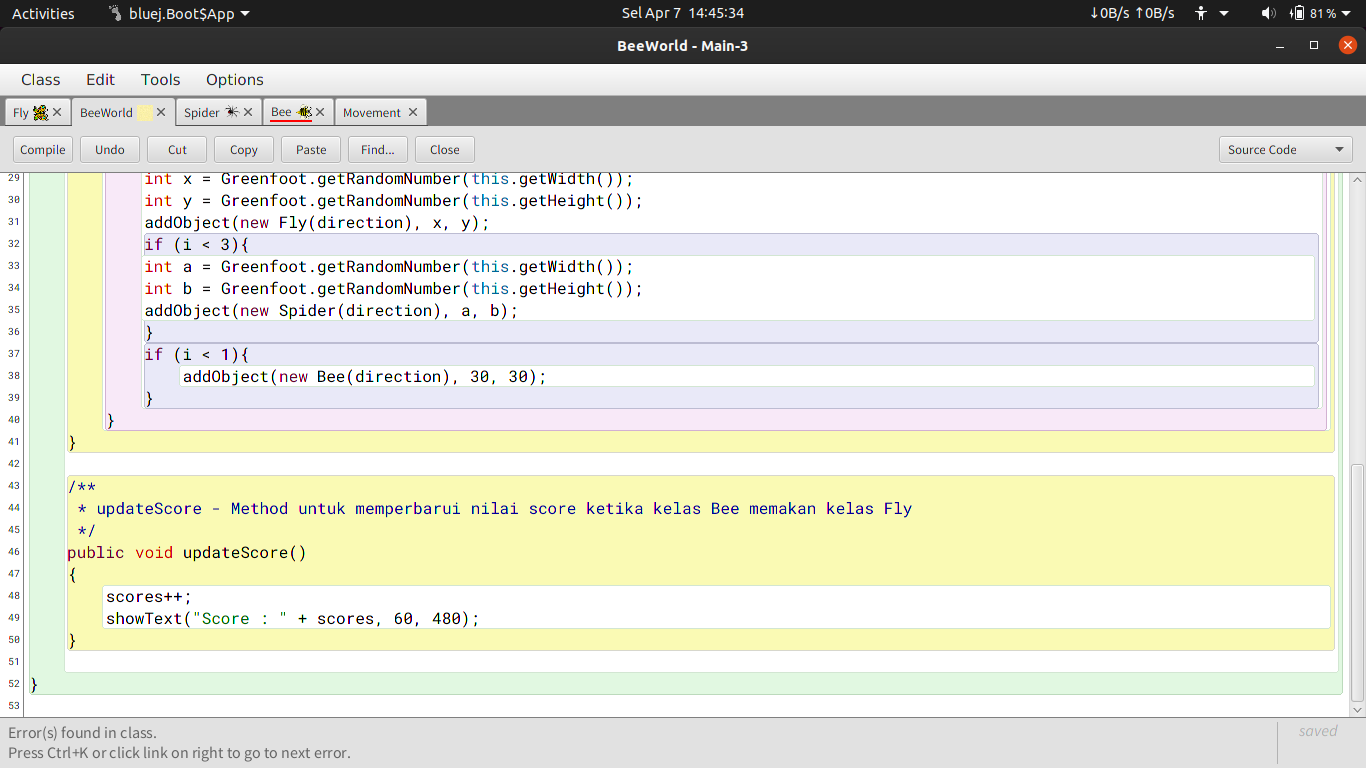
1. Modifikasi program anda, **Tarik/pindahkan** method *handleTepi()* pada class Spider ke class Movement. Kemudian panggil method handleTepi() di method *act()* pada kelas Fly dan kelas Spider.



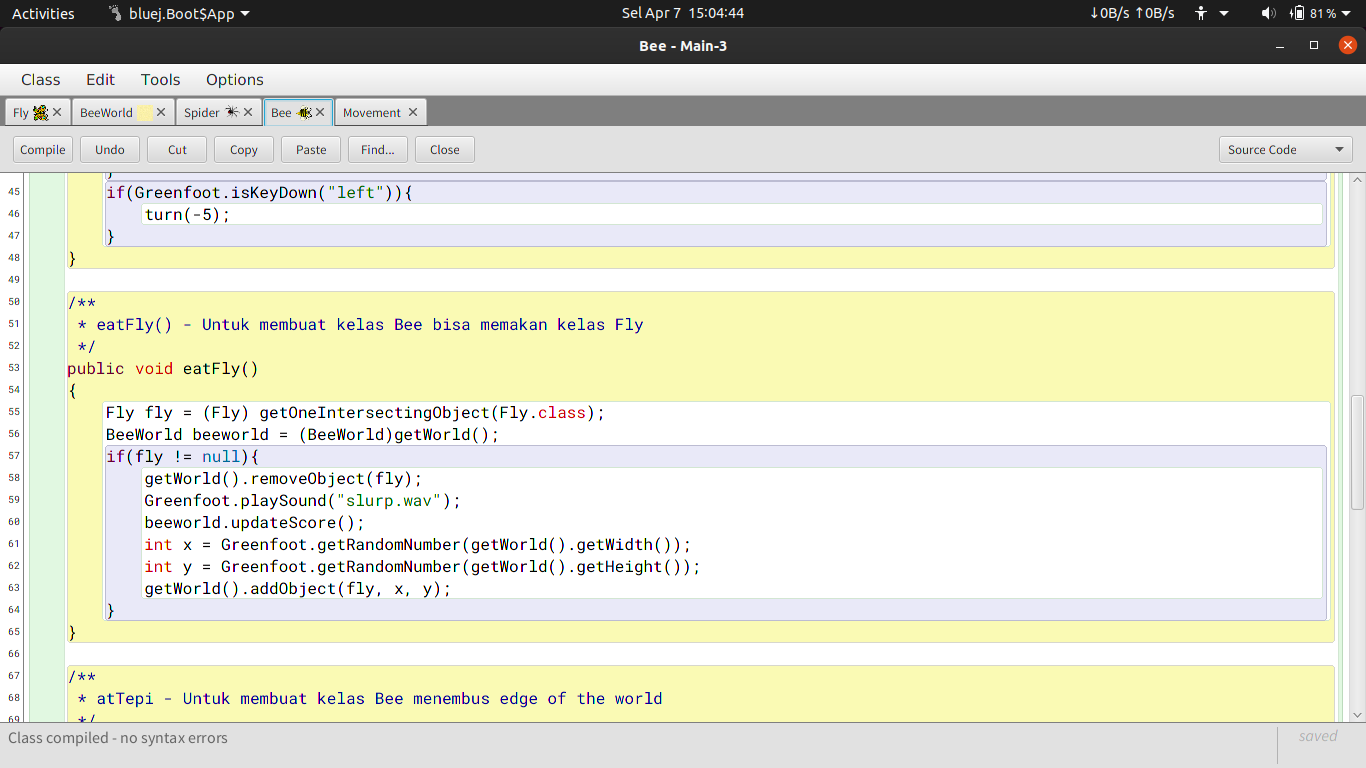




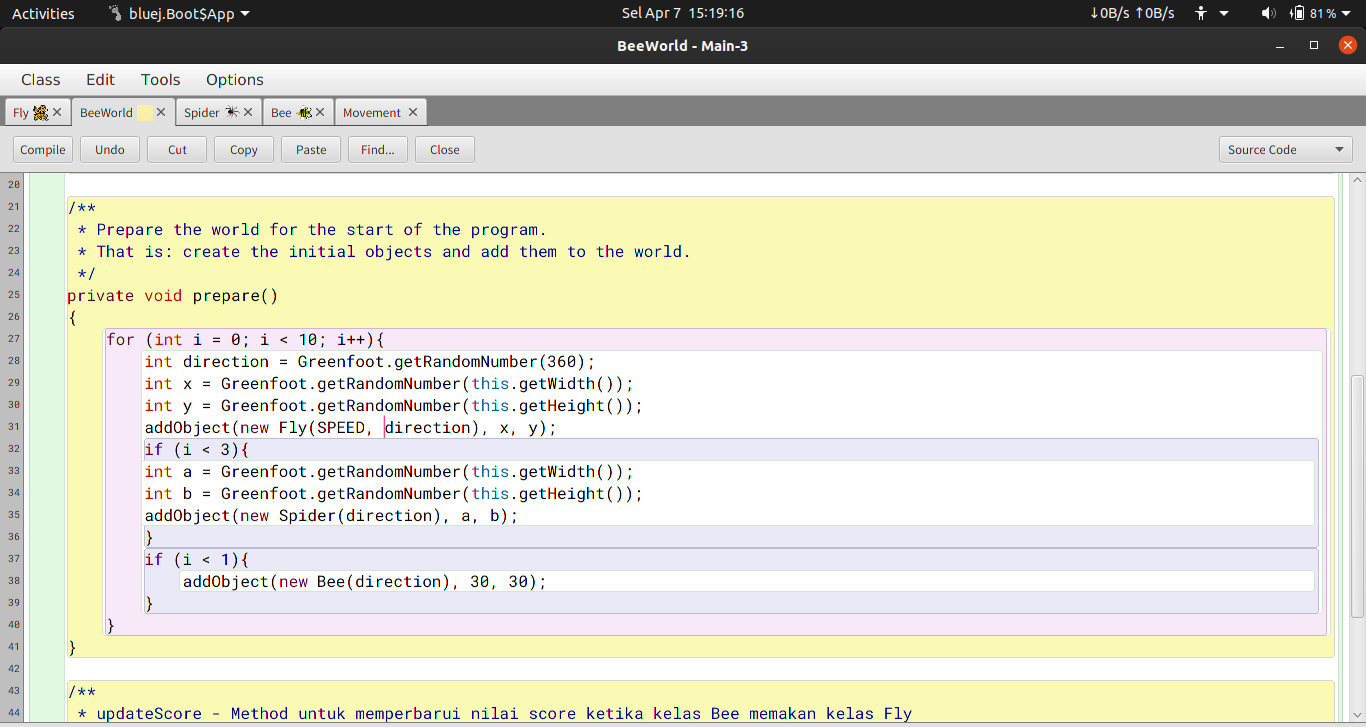
1. Tambahkan variabel score pada kelas BeeWorld. Beri nilai awal score dengan Nol. Kemudian buatlah method *updateScore()* di kelas BeeWorld yang berfungsi untuk menambahkan score ketika Bee memakan objek Fly serta menampilkan nilai Score pada layar.



1. Panggil method *updateScore()* di kelas Bee pada lokasi yang sesuai untuk menambahkan score ketika Bee memakan Fly. (Gunakan Teknik casting)



Tantangan:

1. Setiap objek Fly yang diciptakan akan memiliki kecepatannya masing-masing (minimal 1 dan max 4). Modifikasi program anda sehingga requirement ini bisa terpenuhi. (Gunakan konsep konstruktor berparameter)

***Setelah sesi praktikum SELESAI, laporan praktikum dan source code (zip) harus dikirim/diupload ke google classroom sebelum pertemuan berikutnya.***