**LAPORAN PRAKTIKUM OOP DASAR**

**MODUL 13**

**CLASS, OBJECT, AND OVERLOADING**



**Adam Arthur Faizal**

**M3119001**

**TI A**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**SEKOLAH VOKASI**

**UNIVERSITAS SEBELAS MARET**

**SURAKARTA**

**2020**

**MODUL 13. CLASS, OBJECT, OVERLOADING**

**Capaian Pembelajaran Praktikum:**

* Menerapkan konsep class, object & method
* Menerapkan overloading konstruktor/method

**Tools:**

* Java Development Kit (JDK)
* Eclipse

**Terminologi:**

Isikan terminology yang sesuai untuk definisi dibawah ini:

[this] An optional keyword used to access the members and methods of a class.

[garbage collector] A built-in function of the Java VM that frees memory as objects are no longer needed or referenced.

[Finalizers] An optional method that is called just before an object is removed by the garbage collector.

[Overloading Constructors] Having more than one constructor or method with the same name but different arguments.

[Access modifiers] Used to specify accessibility for variables, methods, and classes.

[default constructor] A constructor that does not have any parameters

[variable arguments method] A way to call a method with a variable number of arguments.

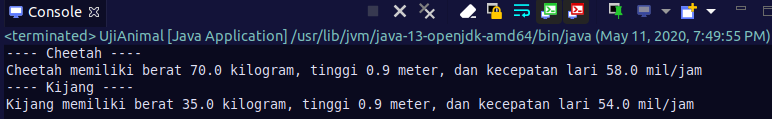
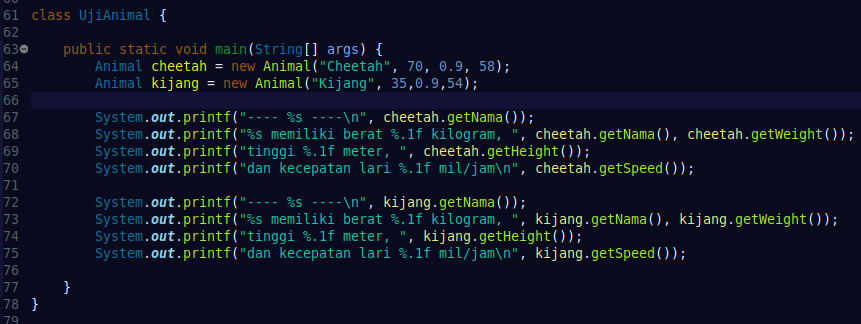
**TRY IT / SOLVE IT:**

1. Identifikas bagian dari class java berikut. Beri tanda asterisk (\*) untuk semua instance variabel. Tandai dengan box pada masing-masing konstruktor. Lingkari signature method nya. Tandai dengan segitiga di sekitar parameternya. Garis bawahi tipe nilai kembalian dari method.





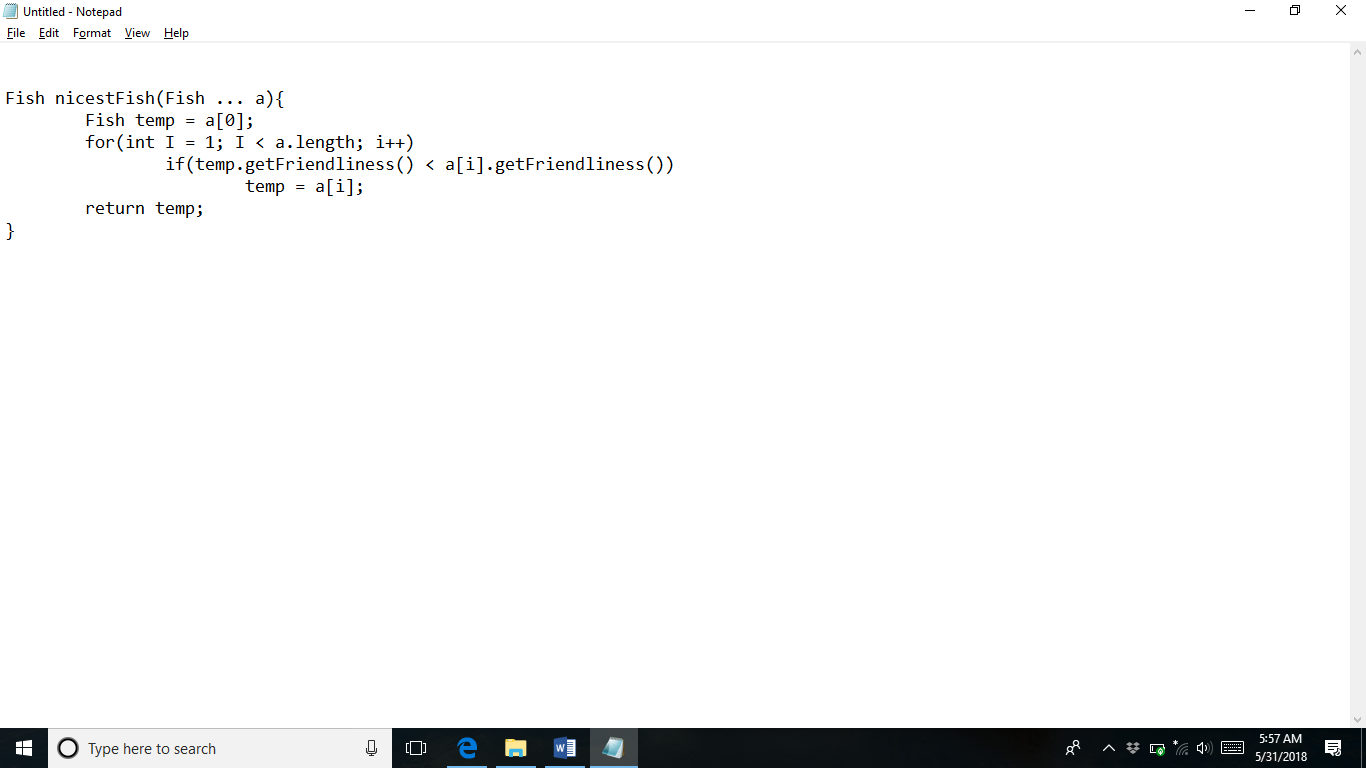
1. Tambahkan method main pada program pada nomor 1 untuk membuat 2 objek dengan menggunakan semua konstruktor yang disediakan (nilai parameter sembarang). Kemudian tambahkan kode untuk mencetak ke layar sbb:
   * Animal #1 memiliki kecepatan …. (isian sesuai dengan nilai parameter)
   * Animal #2 memiliki kecepatan …. (isian sesuai dengan nilai parameter)



1. Buatlah sebuah class public dengan nama Fish dengan fitur sbb:
   * Class *Fish* memiliki 2 instance variabel yakni *typeOfFish* bertipe String dan *friendliness* bertipe integer.
   * Buatlah konstruktor default, dimana body konstruktor akan memberikan nilai “Unknown” untuk variabel *typeOfFish* dan nilai 3 untuk variabel *friendliness.*
   * Tambahkan konstruktor yang lain dengan 2 parameter yakni String t dan integer f. Dalam body konstruktor nilai parameter t akan mengisi variabel *typeofFish* dan parameter f akan mengisi variabel *friendliness.*
   * Tambahkan method setter dan getter untuk masing-masing instance variable.
   * Tulis segment code untuk membuat 2 objek fish yakni Amber dan James dengan definisi sbb:
     + Amber -- Type: AngelFish, Frindliness level: 5
     + James – Type: Guppy, Friendliness level: 3
   * Buat functional method *nicestFish* dengan parameter berupa 2 objek fish yang membandingkan level *friendliness* dari 2 fish tersebut dan akan mengembalikan objek fish yang memiliki level *friendliness* paling tinggi. Uji method ini dengan objek fish pada poin e. (*Friendliness scale*: 1 mean, 2 not friendly, 3 neutral, 4 friendly, 5 very friendly)

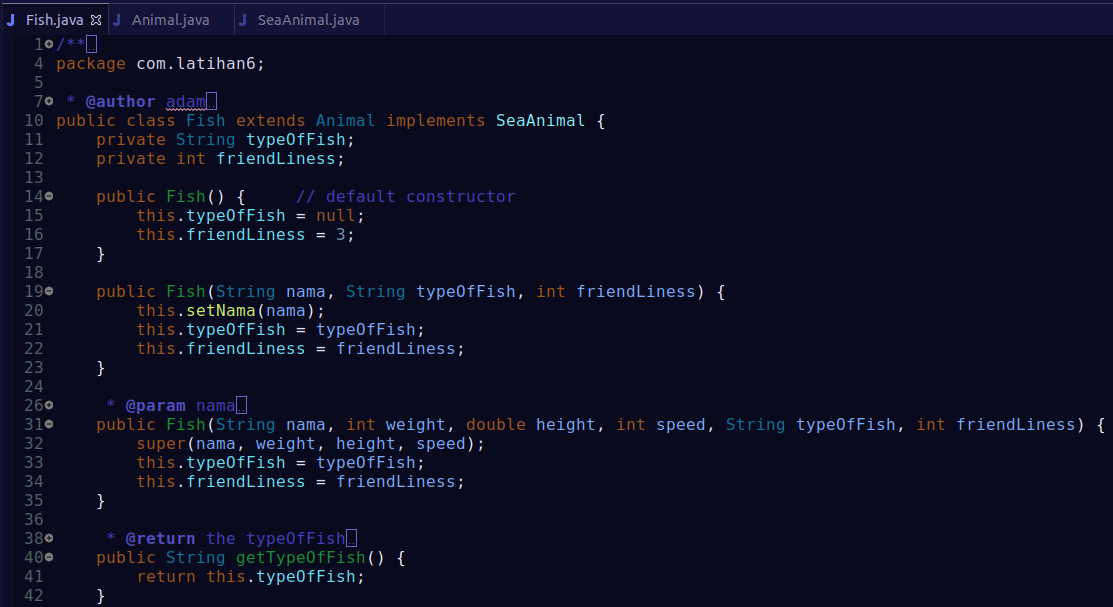
Hint: *fishName.getFriendliness()* akan mengambil nilai *friendliness* dari *fishName*.

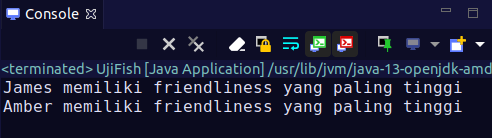
* + Tambahkan method *nicestFish* yang lain untuk membandingkan lebih dari 2 objek fish sebagai berikut:



* + Uji method pada poin g dengan membandingkan 3 objek. (Sebelumnya anda tambahkan satu objek fish dengan parameter yang berbeda dari objek amber dan james). Jelaskan cara kerjanya.



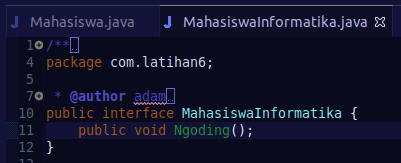


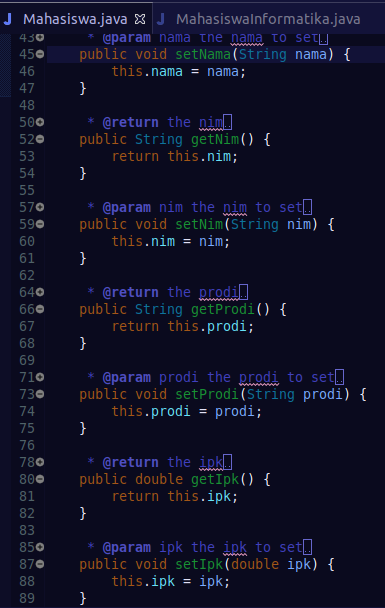


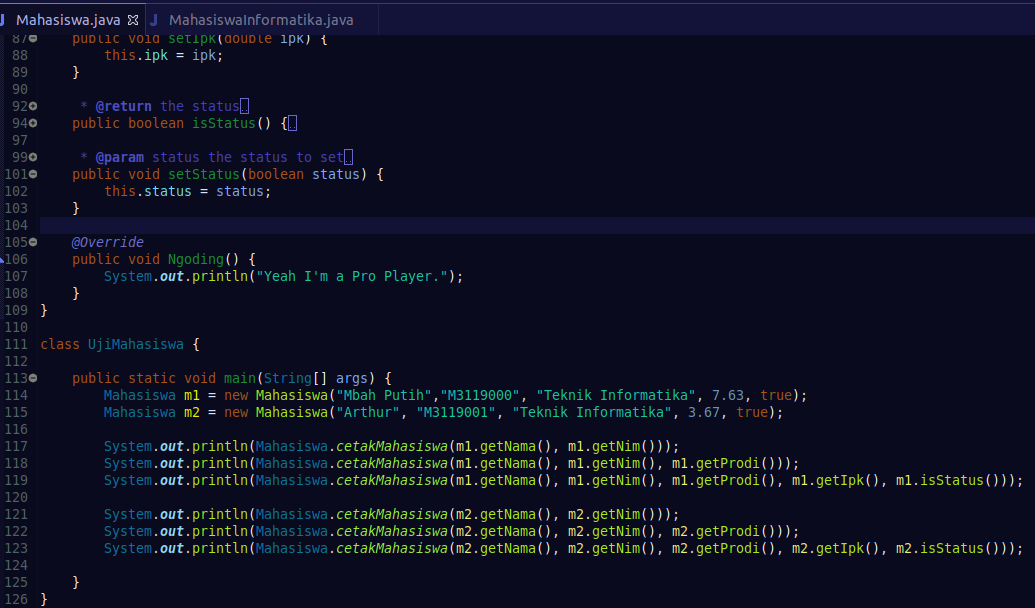
**LATIHAN:**

1. Buat program class Mahasiswa dengan atribut NIM, Nama, Prodi, IPK, Status. Kemudian buatlah 3 method dengan nama yang sama *cetakMahasiswa()* untuk menerapkan overloading, yakni:
   * Cetak nama dan NIM
   * Cetak nama, NIM dan prodi
   * Cetak nama, NIM, prodi, IPK dan status.

Anda uji method yang anda buat dengan membuat beberapa objek Mahasiswa.





***Setelah sesi praktikum SELESAI, laporan praktikum harus dikirim/diupload ke google classroom pada hari yang ditentukan.***