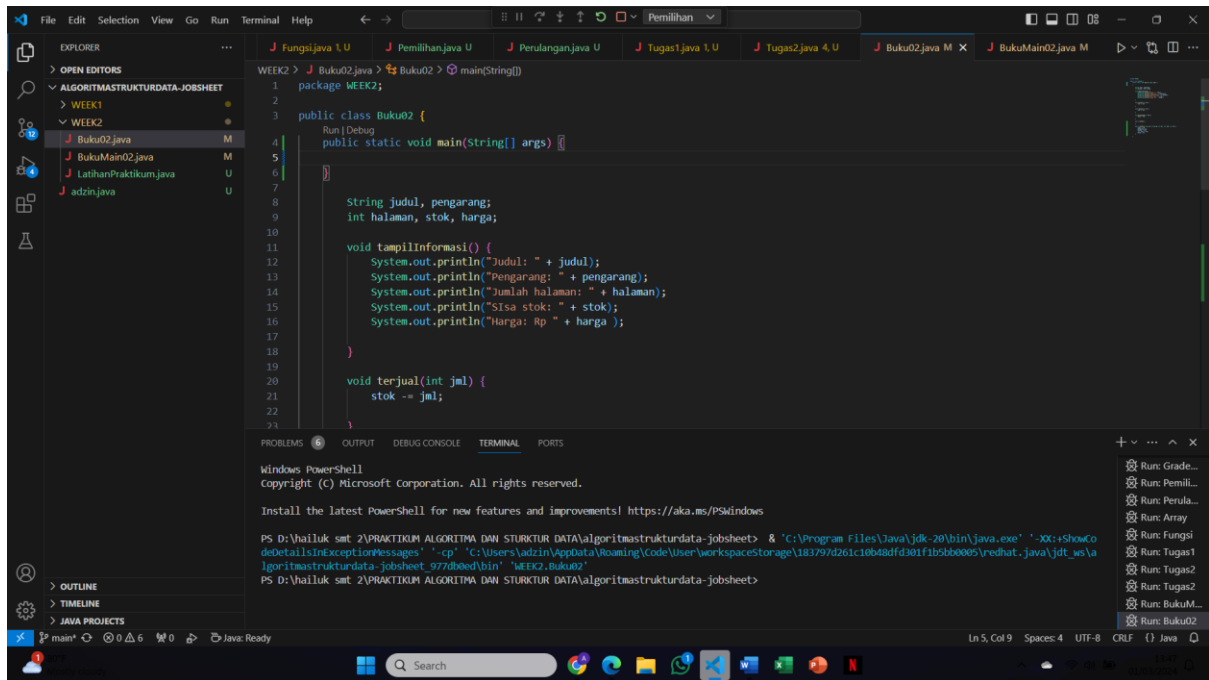


JOBSHEET 2

Verifikasi Hasil Percobaan 1



2.1.3 Pertanyaan

1. Sebutkan dua karakteristik class atau object!
2. Perhatikan class **Buku** pada Praktikum 1 tersebut, ada berapa atribut yang dimiliki oleh class Buku? Sebutkan apa saja atributnya!
3. Ada berapa method yang dimiliki oleh class tersebut? Sebutkan apa saja methodnya!
4. Perhatikan method **terjual()** yang terdapat di dalam class **Buku**. Modifikasi isi method tersebut sehingga proses pengurangan hanya dapat dilakukan jika stok masih ada (lebih besar dari 0)!
5. Menurut Anda, mengapa method **restock()** mempunyai satu parameter berupa bilangan int?
6. **Commit dan push code program ke Github**

Answer

1. Object memiliki 2 karakteristik, yaitu mempunyai sesuatu (istilahnya data, properti, variable, state, dan atribut) dan melakukan sesuatu (istilahnya tingkah laku, behaviour, fungsi, dan method). Class masih berupa rancangan yang bersifat umum. Sedangkan object merupakan bentuk nyata yang terbentuk dari suatu class dan memiliki sifat spesifik.
2. Buku memiliki 5 atribut. Atributnya adalah judul, pengarang, halaman, stok, dan harga.
3. Ada 4 method, yaitu tampilInformasi, terjual, restock, dan gantiHarga.

```

void terjual(int jml) {
    if (stok > 0) {
        stok -= jml;
    }
}

```

- 4.
5. Method restock() memiliki 1 parameter karena kita membutuhkan return stok akhir buku setelah melewati method restock

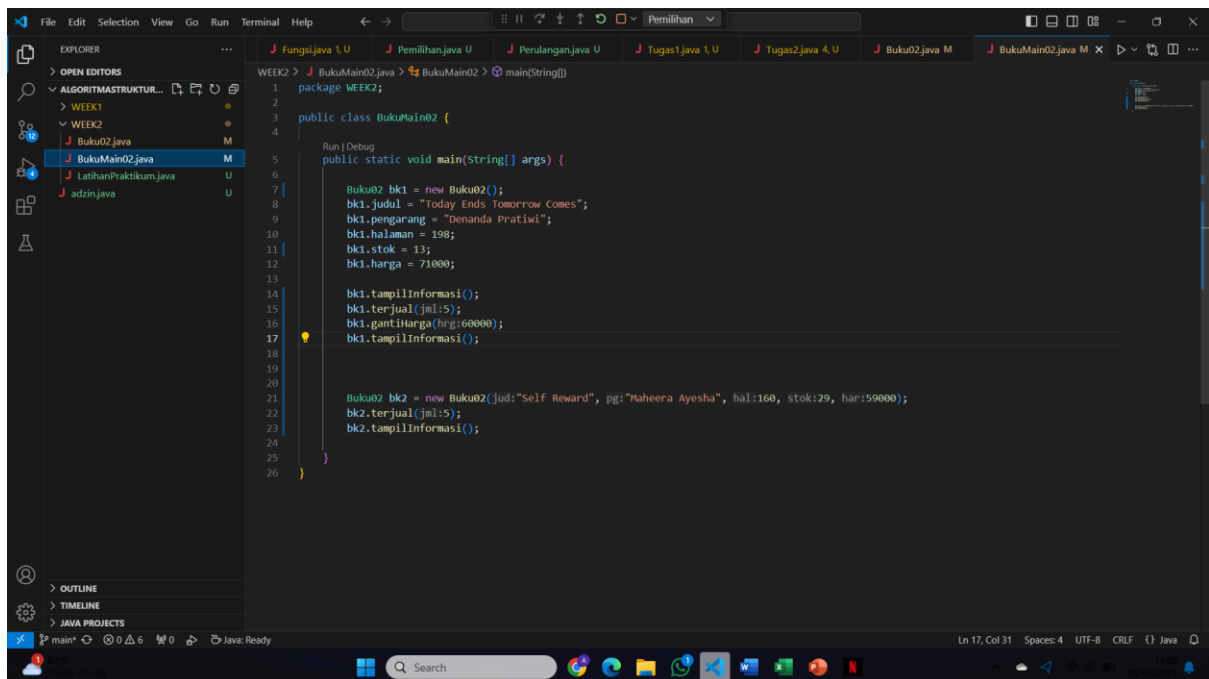
```

1 package WEEK2;
2
3 public class Buku02 {
4
5
6     String judul, pengarang;
7     int halaman, stok, harga;
8
9     void tampilInformasi() {
10         System.out.println("Judul: " + judul);
11         System.out.println("Pengarang: " + pengarang);
12         System.out.println("Jumlah halaman: " + halaman);
13         System.out.println("Sisa stok: " + stok);
14         System.out.println("Harga: Rp " + harga);
15     }
16
17     void terjual(int jml) {
18         stok -= jml;
19     }
20
21     void restock(int jml) {
22         stok += jml;
23     }
24
25     void gantiHarga(int hrg) {
26         harga = hrg;
27     }
28
29
30
31
32 }

```

- 6.

Verifikasi Hasil Percobaan 2



2.2.3 Pertanyaan

1. Pada class **BukuMain**, tunjukkan baris kode program yang digunakan untuk proses instansiasi!
Apa nama object yang dihasilkan?
2. Bagaimana cara mengakses atribut dan method dari suatu objek?
3. Mengapa hasil output pemanggilan method **tampilInformasi()** pertama dan kedua berbeda?

Answer

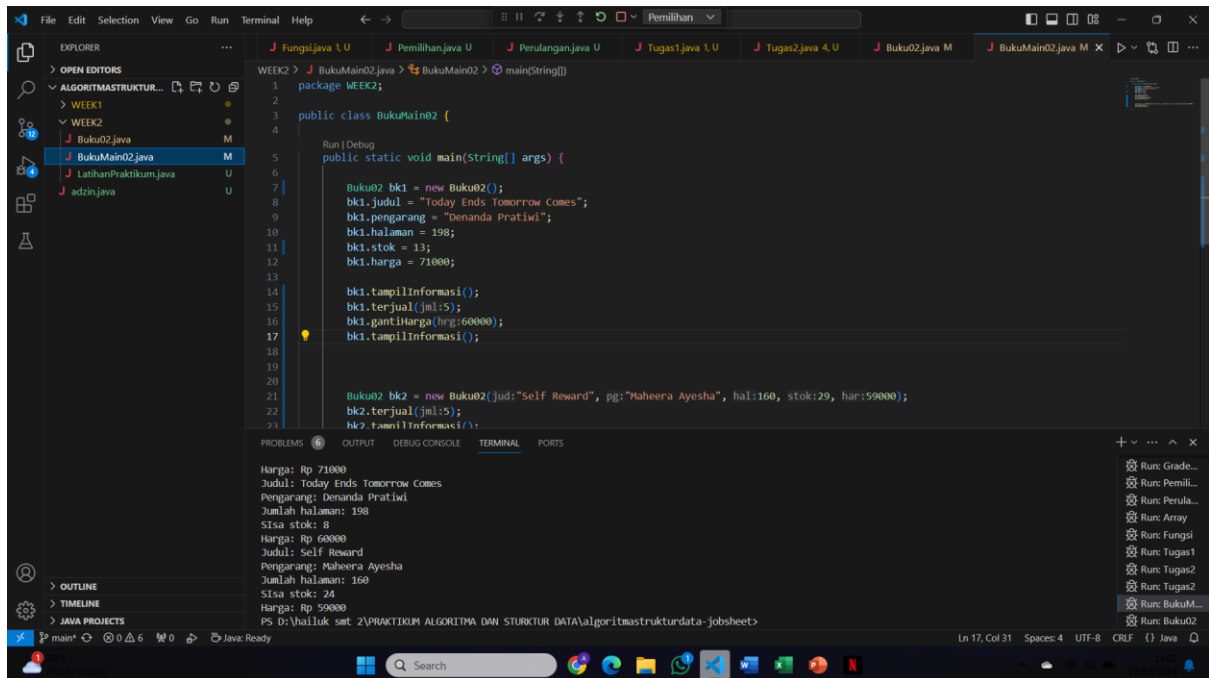
1.

```
Buku02 bk1 = new Buku02();
```
2. Cara mengakses atribut

```
bk1.judul = "Today Ends Tomorrow Comes";  
bk1.pengarang = "Denanda Pratiwi";  
bk1.halaman = 198;  
bk1.stok = 13;  
bk1.harga = 71000;
```
3. Cara mengakses method

```
bk1.tampilInformasi();  
bk1.terjual(jml:5);  
bk1.gantiHarga(hrg:60000);  
bk1.tampilInformasi();
```
4. Karena terdapat perubahan nilai pada method **terjual()** dan method **gantiHarga()**.

Verifikasi Hasil Percobaan 3



2.3.3 Pertanyaan

1. Pada class **Buku** di Percobaan 3, tunjukkan baris kode program yang digunakan untuk mendeklarasikan konstruktor berparameter!
2. Perhatikan class **BukuMain**. Apa sebenarnya yang dilakukan pada baris program berikut?
`Buku bk2 = new Buku(jud:"Self Reward", pg:"Maheera Ayesha", hal:160, stok:29, har:59000);`
3. Hapus konstruktor default pada class **Buku**, kemudian compile dan run program. Bagaimana hasilnya? Jelaskan mengapa hasilnya demikian!
4. Setelah melakukan instansiasi object, apakah method di dalam class **Buku** harus diakses secara berurutan? Jelaskan alasannya!
5. Buat object baru dengan nama **buku<NamaMahasiswa>** menggunakan konstruktor berparameter dari class **Buku**!
6. **Commit dan push kode program ke Github**

Answer

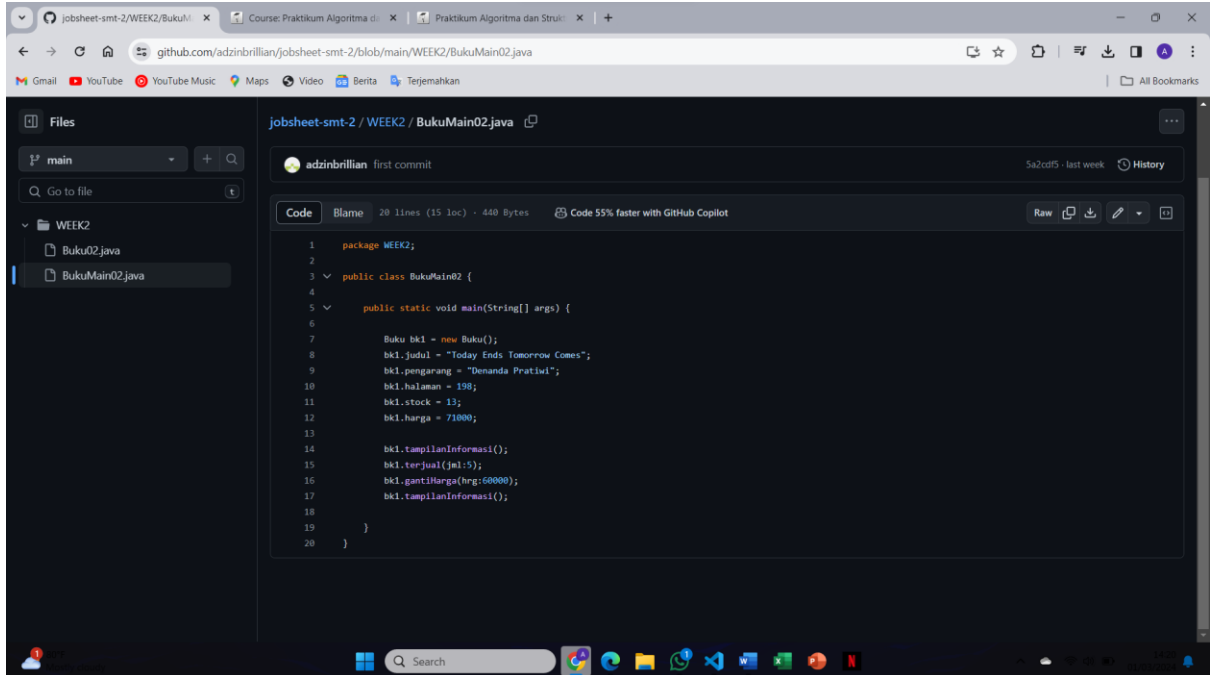
- ```
public Buku02(String jud, String pg, int hal, int stok, int har) {
 judul = jud;
 pengarang = pg;
 halaman = hal;
 this.stok = stok;
 harga = har;
}
```
- 1.
  2. `Buku02 bk2 = new Buku02(jud:"Self Reward", pg:"Maheera Ayesha", hal:160, stok:29, har:59000);`

Membuat object baru bernama bk2 dengan menggunakan konstruktor berparameter.

- Hasilnya akan error, karena pada object bk1 menggunakan konstruktor default. Jika dihapus konstruktor pada object bk1 tidak terdefinisi.
- Method pada class Buku tidak harus diakses secara berurutan karena class tersebut hanya template . Yang diakses secara berurutan adalah class BukuMain

```
Buku02 bukuAdzin = new Buku02(jud:"Self Reward", pg:"Maheera Ayesha", hal:160, stok:29, har:59000);
bk2.terjual(5);
bk2.tampilInformasi();
```

5.



```
package WEEK2;

public class BukuMain02 {
 public static void main(String[] args) {
 Buku bk1 = new Buku();
 bk1.judul = "Today Ends Tomorrow Comes";
 bk1.pengarang = "Denanda Pratiwi";
 bk1.halaman = 198;
 bk1.stok = 13;
 bk1.harga = 71000;

 bk1.tampilInformasi();
 bk1.terjual(jml:5);
 bk1.gantiHarga(hrg:60000);
 bk1.tampilInformasi();
 }
}
```

6.

## 2.4 Latihan Praktikum

Waktu : 150 Menit

1. Pada class Buku yang telah dibuat, tambahkan tiga method yaitu **hitungHargaTotal()**, **hitungDiskon()**, dan **hitungHargaBayar()** dengan penjelasan sebagai berikut:
  - o Method **hitungHargaTotal()** digunakan untuk menghitung harga total yang merupakan perkalian antara harga dengan jumlah buku yang terjual
  - o Method **hitungDiskon()** digunakan untuk menghitung diskon dengan aturan berikut:
    - Jika harga total lebih dari 150000, maka harga didiskon sebesar 12%
    - Jika harga total antara 75000 sampai 150000, maka harga didiskon sebesar 5%

Tim Ajar Algoritma dan Struktur Data 2023-2024

Jurusan Teknologi Informasi-Politeknik Negeri Malang

5

Algoritma dan Struktur Data 2023-2024



- Jika harga total kurang dari 75000, maka harga tidak didiskon
- o Method **hitungHargaBayar()** digunakan untuk menghitung harga total setelah dikurangi diskon

Class diagram **Buku** setelah penambahan ketiga method tersebut adalah sebagai berikut.

| Buku                                                                                                                                                                                                       |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| judul: String<br>pengarang: String<br>halaman: int<br>stok: int<br>harga: int                                                                                                                              |
| tampilInformasi(): void<br>terjual(jml: int): void<br>restock(n: int): void<br>gantiHarga(hrg: int): int<br><b>hitungHargaTotal(): int</b><br><b>hitungDiskon(): int</b><br><b>hitungHargaBayar(): int</b> |

Answer

```
int hitungDiskon(int jml) {
 int diskon = 0;
 int hargaTotal = hitunganHargaTotal(jml);
 if (hargaTotal > 150000)
 diskon = (int) (0.12*hargaTotal);
 else if (hargaTotal < 150000 && hargaTotal > 75000)
 diskon = (int)(0.05*hargaTotal);
 else
 diskon = 0;
 return diskon;
}

int hitunganHargaBayar(int jml) {
 return hitungHargaBayar(int jml) - hitungDiskon(jml);
}
```

2. Buat program berdasarkan class diagram berikut ini!

| Dragon                                                                                                                                        |
|-----------------------------------------------------------------------------------------------------------------------------------------------|
| x: int<br>y: int<br>width: int<br>height: int                                                                                                 |
| moveLeft(): void<br>moveRight(): void<br>moveUp(): void<br>moveDown(): void<br>printPosition(): void<br>detectCollision(x: int, y: int): void |

Penjelasan dari atribut dan method pada class Dragon tersebut adalah sebagai berikut:

- Atribut **x** digunakan untuk menyimpan posisi koordinat x (mendatar) dari dragon, sedangkan atribut **y** untuk posisi koordinat y (vertikal)
- Atribut **width** digunakan untuk menyimpan lebar dari area permainan, sedangkan **height** untuk menyimpan panjang area
- Method **moveLeft()** digunakan untuk mengubah posisi dragon ke kiri (koordinat x akan berkurang 1), sedangkan **moveRight()** untuk bergerak ke kanan (koordinat x akan bertambah 1). Perlu diperhatikan bahwa koordinat x tidak boleh lebih kecil dari 0 atau lebih besar dari nilai width. Jika koordinat  $x < 0$  atau  $x > \text{width}$  maka panggil method **detectCollision()**
- Method **moveUp()** digunakan untuk mengubah posisi dragon ke atas (koordinat y akan berkurang 1), sedangkan **moveDown()** untuk bergerak ke bawah (koordinat y akan bertambah 1)

Tim Ajar Algoritma dan Struktur Data 2023-2024

Jurusan Teknologi Informasi-Politeknik Negeri Malang

6

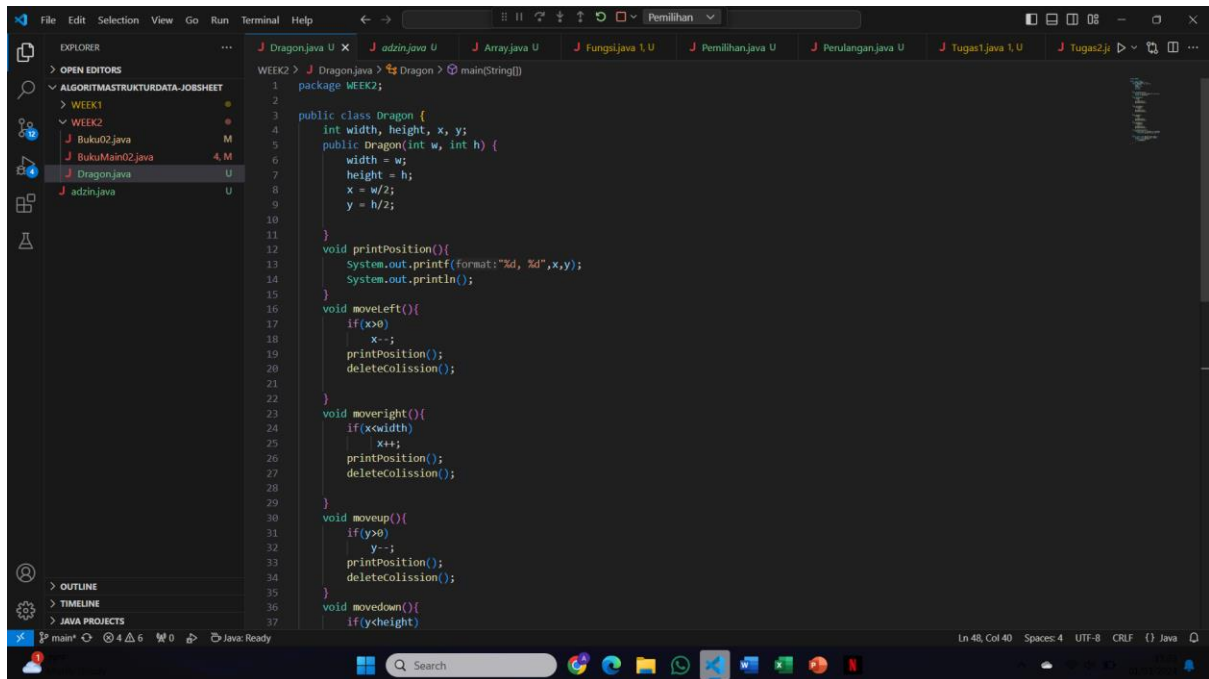
Algoritma dan Struktur Data 2023-2024



- 1). Perlu diperhatikan bahwa koordinat y tidak boleh lebih kecil dari 0 atau lebih besar dari nilai height. Jika koordinat  $y < 0$  atau  $y > \text{height}$  maka panggil method **detectCollision()**
- Method **detectCollision()** akan mencetak pesan "Game Over" apabila dragon menyentuh ujung area permainan.

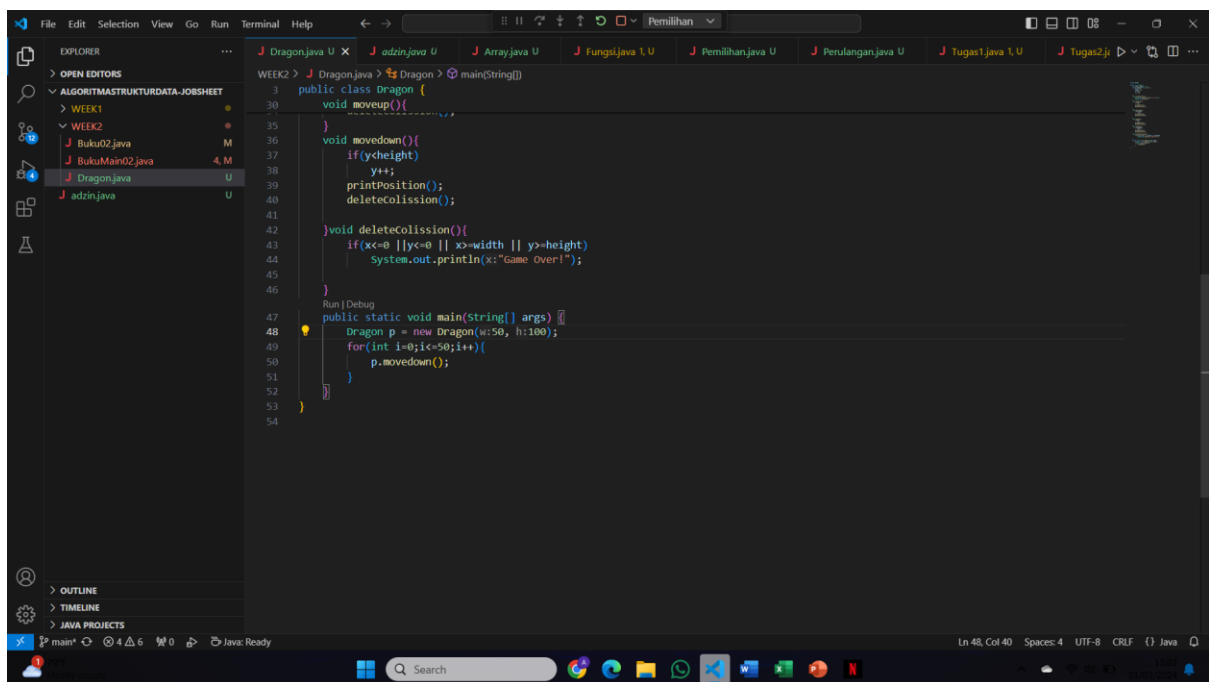


## Answer



This screenshot shows the first part of the Java code in the Dragon.java file. The Explorer on the left shows the project structure with files like Dragon.java, adzin.java, and others. The main editor displays the following code:

```
1 package WEEK2;
2
3 public class Dragon {
4 int width, height, x, y;
5 public Dragon(int w, int h) {
6 width = w;
7 height = h;
8 x = w/2;
9 y = h/2;
10 }
11
12 void printPosition(){
13 System.out.printf(format:"%d", %d",x,y);
14 System.out.println();
15 }
16 void moveLeft(){
17 if(x>0)
18 x--;
19 printPosition();
20 deleteColission();
21 }
22
23 void moveright(){
24 if(x<width)
25 x++;
26 printPosition();
27 deleteColission();
28 }
29
30 void moveup(){
31 if(y>0)
32 y--;
33 printPosition();
34 deleteColission();
35 }
36
37 void movedown(){
38 if(y<height)
```



This screenshot shows the continuation of the Java code in the Dragon.java file. The Explorer on the left is the same. The main editor displays the following code:

```
39 y++;
40 printPosition();
41 deleteColission();
42 }
43 void deleteColission(){
44 if(x<=0 || y<=0 || x>=width || y>=height)
45 System.out.println(x:"Game Over!");
46 }
47
48 Run | Debug
49 public static void main(String[] args) {
50 Dragon p = new Dragon(w:50, h:100);
51 for(int i=0;i<50;i++){
52 p.movedown();
53 }
54 }
```

