

Tugas Praktikum 4

Relasi Kelas



Rafi Ody Prasetyo
(2341720180)

D-IV Teknik Informatika
Politeknik Negeri Malang
Semester 3
2024

Percobaan 1

Code:

Processor.java

```
package ac.id.polinema;

public class Processor {
    private String merk;
    private double cache;
    public Processor() {}
    public Processor(String merk, double cache) {
        this.merk = merk;
        this.cache = cache;
    }
    public void setMerk(String merk) {
        this.merk = merk;
    }
    public void setCache(double cache) {
        this.cache = cache;
    }
    public String getMerk() {
        return merk;
    }
    public double getCache() {
        return cache;
    }
    public void info() {
        System.out.printf("Merk Processor = %s\n", merk);
        System.out.printf("Cache Memory = %.2f\n", cache);
    }
}
```

Laptop.java

```
package ac.id.polinema;

public class Laptop {
    private String merk;
    private Processor proc;
    public Laptop() {}
    public Laptop(String merk, Processor proc) {
        this.merk = merk;
        this.proc = proc;
    }
    public void setMerk(String merk) {
        this.merk = merk;
    }
    public void setProc(Processor proc) {
        this.proc = proc;
    }
    public String getMerk() {
        return merk;
    }
    public Processor getProc() {
        return proc;
    }
    public void info() {
        System.out.println("Merk Laptop = " + merk);
        proc.info();
    }
}
```

MainPercobaan1.java

```
package ac.id.polinema;

public class MainPercobaan1 {
    public static void main(String[] args) {
        Processor p = new Processor("Intel i5", 3);
        Laptop l = new Laptop("Thinkpad", p);
        l.info();
        Processor p1 = new Processor();
        p1.setMerk("Intel i5");
        p1.setCache(4);
        Laptop L1 = new Laptop();
        L1.setMerk("Thinkpad");
        L1.setProc(p1);
        L1.info();
    }
}
```

Output:

```
Merk Laptop = Thinkpad
Merk Processor = Intel i5
Cache Memory = 3.00
Merk Laptop = Thinkpad
Merk Processor = Intel i5
Cache Memory = 4.00
```

Pertanyaan Percobaan 1

1. Di dalam class `Processor` dan class `Laptop` , terdapat method setter dan getter untuk masing-masing atributnya. Apakah gunanya *method setter* dan *getter* tersebut ?

Jawab:

Setter digunakan untuk memasukkan nilai ke dalam variabel, sedangkan getter digunakan untuk mengambil nilai dari variabel.

2. Di dalam class `Processor` dan class `Laptop`, masing-masing terdapat konstruktor default dan konstruktor berparameter. Bagaimanakah beda penggunaan dari kedua jenis konstruktor tersebut ?

Jawab:

Konstruktor default berfungsi untuk memberikan nilai awal kepada properti objek secara default ketika objek tersebut dibuat. Sedangkan konstruktor berparameter digunakan untuk memberikan nilai yang spesifik untuk properti saat objek dibuat.

3. Perhatikan class `Laptop`, di antara 2 atribut yang dimiliki (merk dan proc), atribut manakah yang bertipe object ?

Jawab:

Proc merupakan atribut yang bertipe object. Atribut ini digunakan untuk mengambil nilai dari class `Processor`.

4. Perhatikan class `Laptop`, pada baris manakah yang menunjukkan bahwa class `Laptop` memiliki relasi dengan class `Processor` ?

Jawab:

Pada method `info()`, karena pada method tersebut data dari class `processor` ditampilkan pada method `info()` yang terdapat pada class `laptop`.

5. Perhatikan pada class `Laptop` , Apakah guna dari sintaks `proc.info()` ?

Jawab:

Untuk menampilkan merek laptop dan juga informasi dari processor.

6. Pada class `MainPercobaan1`, terdapat baris kode:

```
Laptop l = new Laptop("Thinkpad", p);
```

Apakah `p` tersebut ? Dan apakah yang terjadi jika baris kode tersebut diubah menjadi:

```
Laptop l = new Laptop("Thinkpad", new Processor("Intel i5", 3));
```

Bagaimanakah hasil program saat dijalankan, apakah ada perubahan ?

Jawab:

P merupakan variabel yang digunakan untuk memanggil konstruktor processor, dan variabel tersebut digunakan untuk menampilkan informasi processor pada konstruktor laptop. Jika kode diubah seperti diatas, hasilnya akan sama seperti menggunakan variabel p.

Percobaan 2

Code:

Sopir.java

```
public class Sopir {  
    private String nama;  
    private int biaya;  
    public Sopir() {}  
    public Sopir(String nama, int biaya) {  
        this.nama = nama;  
        this.biaya = biaya;  
    }  
    public String getNama() {  
        return nama;  
    }  
    public int getBiaya() {  
        return biaya;  
    }  
    public void setNama(String nama) {  
        this.nama = nama;  
    }  
    public void setBiaya(int biaya) {  
        this.biaya = biaya;  
    }  
    public int hitungBiayaSopir(int hari) {  
        return biaya * hari;  
    }  
}
```

Mobil.java

```
public class Mobil {  
    private String merk;  
    private int biaya;  
    public Mobil() {}  
    public Mobil(String merk, int biaya) {  
        this.merk = merk;  
        this.biaya = biaya;  
    }  
    public void setMerk(String merk) {  
        this.merk = merk;  
    }  
    public String getMerk() {  
        return merk;  
    }  
    public void setBiaya(int biaya) {  
        this.biaya = biaya;  
    }  
    public int getBiaya() {  
        return biaya;  
    }  
    public int hitungBiayaMobil(int hari) {  
        return biaya * hari;  
    }  
}
```

Pelanggan.java

```
public class Pelanggan {  
    private String nama;  
    private Mobil mobil;  
    private Sopir sopir;  
    private int hari;  
    public Pelanggan() {}  
    public Pelanggan(String nama, Mobil mobil, Sopir sopir, int  
hari) {  
        this.nama = nama;  
        this.mobil = mobil;  
        this.sopir = sopir;  
        this.hari = hari;  
    }  
    public String getNama() {  
        return nama;  
    }  
    public void setNama(String nama) {  
        this.nama = nama;  
    }  
    public int getHari() {  
        return hari;  
    }  
    public void setHari(int hari) {  
        this.hari = hari;  
    }  
    public Sopir getSopir() {  
        return sopir;  
    }  
    public void setSopir(Sopir sopir) {  
        this.sopir = sopir;  
    }  
    public Mobil getMobil() {  
        return mobil;  
    }  
    public void setMobil(Mobil mobil) {  
        this.mobil = mobil;  
    }  
    public int hitungBiayaTotal() {  
        return mobil.hitungBiayaMobil(hari) +  
            sopir.hitungBiayaSopir(hari);  
    }  
}
```

MainPercobaan2.java

```
public class MainPercobaan2 {  
    public static void main(String[] args) {  
        Mobil m = new Mobil();  
        m.setMerk("Avanza");  
        m.setBiaya(350000);  
        Sopir s = new Sopir();  
        s.setNama("John Doe");  
        s.setBiaya(200000);  
        Pelanggan p = new Pelanggan();  
        p.setNama("Jane Doe");  
        p.setMobil(m);  
        p.setSopir(s);  
        p.setHari(2);  
        System.out.println("Biaya Total = " +  
            p.hitungBiayaTotal());  
    }  
}
```

Output:

```
Biaya Total = 1100000
```

Pertanyaan Percobaan 2

1. Perhatikan class Pelanggan. Pada baris program manakah yang menunjukkan bahwa class Pelanggan memiliki relasi dengan class Mobil dan class Sopir ?

Jawab:

Pada method hitungBiayaTotal(), karena disana menjumlahkan nilai hari pada class mobil dan sopir.

```
public int hitungBiayaTotal() {  
    return mobil.hitungBiayaMobil(hari) +  
        sopir.hitungBiayaSopir(hari);  
}
```

2. Perhatikan method hitungBiayaSopir pada class Sopir, serta method hitungBiayaMobil pada class Mobil. Mengapa menurut Anda method tersebut harus memiliki argument hari ?

Jawab:

Karena biaya sopir dan mobil dihitung per hari.

3. Perhatikan kode dari class Pelanggan. Untuk apakah perintah mobil.hitungBiayaMobil(hari) dan sopir.hitungBiayaSopir(hari) ?

Jawab:

Untuk menghitung biaya total.

4. Perhatikan class MainPercobaan2. Untuk apakah sintaks `p.setMobil(m)` dan `p.setSopir(s)` ?

Jawab:

Digunakan untuk memanggil data yang ada di class mobil dan sopir.

5. Perhatikan class MainPercobaan2. Untuk apakah proses `p.hitungBiayaTotal()` tersebut ?

Jawab:

Untuk menampilkan hasil dari method `hitungBiayaTotal()`.

6. Perhatikan class MainPercobaan2, coba tambahkan pada baris terakhir dari method main dan amati perubahan saat di-run!

`System.out.println(p.getMobil().getMerk());`

Jadi untuk apakah sintaks `p.getMobil().getMerk()` yang ada di dalam method main tersebut?

Jawab:

```
Biaya Total = 1100000  
Avanza
```

Digunakan untuk menampilkan nilai merek pada class Mobil.

Percobaan 3

Code:

Pegawai.java

```
public class Pegawai {  
    private String nip;  
    private String nama;  
    public Pegawai() {}  
    public Pegawai(String nip, String nama) {  
        this.nip = nip;  
        this.nama = nama;  
    }  
    public String getNip() {  
        return nip;  
    }  
    public void setNip(String nip) {  
        this.nip = nip;  
    }  
    public String getNama() {  
        return nama;  
    }  
    public void setNama(String nama) {  
        this.nama = nama;  
    }  
    public String info() {  
        String info = "";  
        info += "Nip: " + this.nip + "\n";  
        info += "Nama: " + this.nama + "\n";  
        return info;  
    }  
}
```

KeretaApi.java

```
public class KeretaApi {  
    private String nama;  
    private String kelas;  
    private Pegawai masinis;  
    private Pegawai asisten;  
    public KeretaApi(String nama, String kelas, Pegawai masinis)  
    {  
        this.nama = nama;  
        this.kelas = kelas;  
        this.masinis = masinis;  
    }  
    public KeretaApi(String nama, String kelas, Pegawai masinis,  
    Pegawai asisten) {  
        this.nama = nama;  
        this.kelas = kelas;  
        this.masinis = masinis;  
        this.asisten = asisten;  
    }  
    public String getNama() {  
        return this.nama;  
    }  
    public String getMasinis() {  
        return this.masinis.getNama();  
    }  
    public String getKelas() {  
        return this.kelas;  
    }  
    public String getAsisten() {  
        return this.asisten;  
    }  
    public void setAsisten(Pegawai asisten) {  
        this.asisten = asisten;  
    }  
    public void setMasinis(Pegawai masinis) {  
        this.masinis = masinis;  
    }  
    public void setKelas(String kelas) {  
        this.kelas = kelas;  
    }  
    public void setNama(String nama) {  
        this.nama = nama;  
    }  
    public String info() {  
        String info = "";  
        info += "Nama: " + this.nama + "\n";  
        info += "Kelas: " + this.kelas + "\n";  
        info += "Masinis: " + this.masinis.info() + "\n";  
        info += "Asisten: " + this.asisten.info() + "\n";  
        return info;  
    }  
}
```

MainPercobaan3.java

```
public class MainPercobaan3 {  
    public static void main(String[] args) {  
        Pegawai masinis = new Pegawai("1234", "Spongebob  
Squarepants");  
        Pegawai asisten = new Pegawai("4567", "Patrick Star");  
        KeretaApi keretaApi = new KeretaApi("Gaya Baru",  
"Bisnis",  
        masinis, asisten);  
        System.out.println(keretaApi.info());  
    }  
}
```

Pertanyaan Percobaan 3

1. Di dalam method info() pada class KeretaApi, baris this.masinis.info() dan this.asisten.info() digunakan untuk apa ?

Jawab:

Untuk menampilkan data nip dan nama masinis dan asisten.

2. Buatlah main program baru dengan nama class MainPertanyaan pada package yang sama. Tambahkan kode berikut pada method main() !
3. Apa hasil output dari main program tersebut ? Mengapa hal tersebut dapat terjadi ?

Jawab:

Hasilnya akan error, karena pada kode diatas kurang menambahkan data dari asisten agar method info dapat dijalankan.

4. Perbaiki class KeretaApi sehingga program dapat berjalan !

Jawab:

Menambahkan data asisten.

Percobaan 4

Code:

Pernumpang.java

```
public class Pernumpang {
    private String nama;
    private String ktp;
    public Pernumpang() {}
    public Pernumpang(String nama, String ktp) {
        this.nama = nama;
        this.ktp = ktp;
    }
    public String getNama() {
        return this.nama;
    }
    public void setNama(String nama) {
        this.nama = nama;
    }
    public String getKtp() {
        return this.ktp;
    }
    public void setKtp(String ktp) {
        this.ktp = ktp;
    }
    public String info() {
        String info = "";
        info += "Ktp: " + ktp + "\n";
        info += "Nama: " + nama + "\n";
        return info;
    }
}
```

Kursi.java

```
public class Kursi {  
    private String nomor;  
    private Penumpang penumpang;  
    public Kursi() {}  
    public Kursi(String nomor) {  
        this.nomor = nomor;  
    }  
    public void setNomor(String nomor) {  
        this.nomor = nomor;  
    }  
    public String getNomor() {  
        return this.nomor;  
    }  
    public void setPenumpang(Penumpang penumpang) {  
        this.penumpang = penumpang;  
    }  
    public Penumpang getPenumpang() {  
        return this.penumpang;  
    }  
    public String info() {  
        String info = "";  
        info += "Nomor: " + nomor + "\n";  
        if (this.penumpang != null) {  
            info += "Penumpang: " + penumpang.info() + "\n";  
        }  
        return info;  
    }  
}
```

Gerbong.java

```
public class Gerbong {  
    private String kode;  
    private Kursi[] arrayKursi;  
    public Gerbong() {}  
    private void initKursi() {  
        for (int i = 0; i < arrayKursi.length; i++) {  
            this.arrayKursi[i] = new Kursi(String.valueOf(i +  
1));  
        }  
    }  
    public Gerbong(String kode, int kapasitas) {  
        this.kode = kode;  
        this.arrayKursi = new Kursi[kapasitas];  
        this.initKursi();  
    }  
    public void setKode(String kode) {  
        this.kode = kode;  
    }  
    public String getKode() {  
        return this.kode;  
    }  
    public String info() {  
        String info = "";  
        info += "Kode: " + kode + "\n";  
        for (Kursi kursi : arrayKursi) {  
            info += kursi.info();  
        }  
        return info;  
    }  
    public void setPenumpang(Penumpang penumpang, int nomor) {  
        this.arrayKursi[nomor - 1].setPenumpang(penumpang);  
    }  
}
```

MainPercobaan4.java

```
public class MainPercobaan4 {  
    public static void main(String[] args) {  
        Penumpang p = new Penumpang("12345", "Mr. Krab");  
        Gerbong gerbong = new Gerbong("A", 10);  
        gerbong.setPenumpang(p, 1);  
        System.out.println(gerbong.info());  
    }  
}
```

Output:

```
Kode: A
Nomor: 1
Penumpang: Ktp: Mr. Krab
Nama: 12345

Nomor: 2
Nomor: 3
Nomor: 4
Nomor: 5
Nomor: 6
Nomor: 7
Nomor: 8
Nomor: 9
Nomor: 10
```

Pertanyaan Percobaan 4

1. Pada main program dalam class MainPercobaan4, berapakah jumlah kursi dalam Gerbong A ?

Jawab:

Terdapat total 10 kursi.

2. Perhatikan potongan kode pada method info() dalam class Kursi. Apa maksud kode tersebut ?

```
...
if (this.penumpang != null) {
    info += "Penumpang: " + penumpang.info() + "\n";
}
...
```

Jawab:

Jika nilai variabel penumpang tidak null, maka program akan menampilkan nama dan no ktp penumpang.

3. Mengapa pada method setPenumpang() dalam class Gerbong, nilai nomor dikurangi dengan angka 1 ?

Jawab:

Apabila terdapat penumpang, maka total kursi akan berkurang 1 karena kursi tersebut dipakai oleh penumpang yang baru ditambahkan.

4. Instansiasi objek baru budi dengan tipe Penumpang, kemudian masukkan objek baru tersebut pada gerbong dengan gerbong.setPenumpang(budi, 1). Apakah yang terjadi?

Jawab:

Kursi nomor 1 akan di duduki budi.


```
Kode: A
Nomor: 1
Penumpang: Ktp: 121212
Nama: budi

Nomor: 2
Nomor: 3
Nomor: 4
Nomor: 5
Nomor: 6
Nomor: 7
Nomor: 8
Nomor: 9
Nomor: 10
```

5. Modifikasi program sehingga tidak diperkenankan untuk menduduki kursi yang sudah ada penumpang lain !

Jawab:

Code:

```
public void setPenumpang(Penumpang penumpang, int nomor) {
    if (this.arrayKursi[nomor - 1].getPenumpang() != null) {
        System.out.println("Kursi nomor " + nomor + " sudah terisi.");
    } else {
        this.arrayKursi[nomor - 1].setPenumpang(penumpang);
    }
}
```

Output:

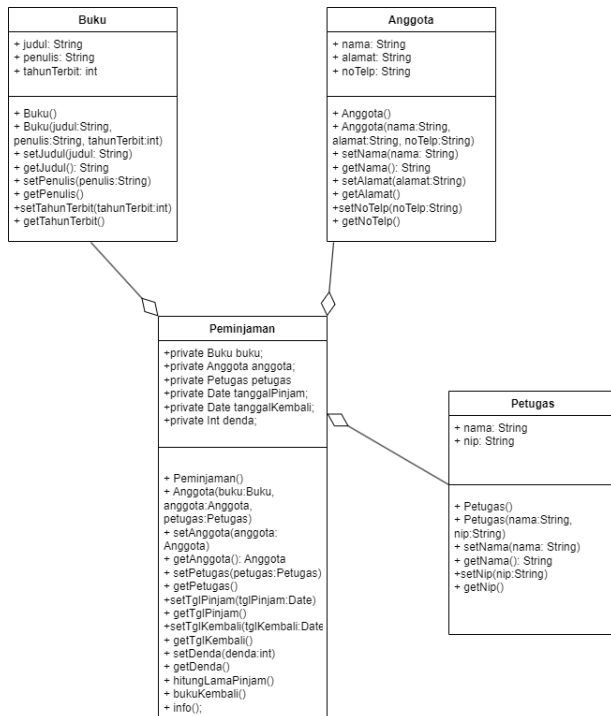
```
Kursi nomor 1 sudah terisi.
Kode: A
Nomor: 1
Penumpang: Ktp: Mr. Krab
Nama: 12345

Nomor: 2
Nomor: 3
Nomor: 4
Nomor: 5
Nomor: 6
Nomor: 7
Nomor: 8
Nomor: 9
Nomor: 10
```

Tugas

Berdasarkan latihan di pertemuan teori, rancang dengan class diagram, kemudian implementasikan ke dalam program! Studi kasus harus mewakili relasi class dari percobaan-percobaan yang telah dilakukan pada materi ini, setidaknya melibatkan minimal 4 class (class yang berisi main tidak dihitung).

Jawab:



Code:

```
public class Anggota {  
    private String nama;  
    private String alamat;  
    private String telepon;  
    public Anggota() {}  
    public Anggota(String nama, String alamat, String telepon) {  
        this.nama = nama;  
        this.alamat = alamat;  
        this.telepon = telepon;  
    }  
    public String getNama() {  
        return nama;  
    }  
    public void setNama(String nama) {  
        this.nama = nama;  
    }  
    public String getAlamat() {  
        return alamat;  
    }  
    public void setAlamat(String alamat) {  
        this.alamat = alamat;  
    }  
    public String getTelepon() {  
        return telepon;  
    }  
    public void setTelepon(String telepon) {  
        this.telepon = telepon;  
    }  
  
    public void info() {  
        System.out.println("Nama: " + getNama());  
        System.out.println("Alamat: " + getAlamat());  
        System.out.println("Telepon: " + getTelepon());  
    }  
}
```

```
public class Buku {

    private String judul;
    private String penulis;
    private int tahunTerbit;

    public Buku() {}

    public Buku(String judul, String penulis, int tahunTerbit) {
        this.judul = judul;
        this.penulis = penulis;
        this.tahunTerbit = tahunTerbit;
    }

    public String getJudul() {
        return judul;
    }

    public void setJudul(String judul) {
        this.judul = judul;
    }

    public String getPenulis() {
        return penulis;
    }

    public void setPenulis(String penulis) {
        this.penulis = penulis;
    }

    public int getTahunTerbit() {
        return tahunTerbit;
    }

    public void setTahunTerbit(int tahunTerbit) {
        this.tahunTerbit = tahunTerbit;
    }

    public void info() {
        System.out.println("Judul : " + judul);
        System.out.println("Penulis : " + penulis);
        System.out.println("Tahun Terbit : " + tahunTerbit);
    }

}
```

```
public class Petugas {

    private String nama;
    private String nip;

    public Petugas() {}

    public Petugas(String nama, String nip) {
        this.nama = nama;
        this.nip = nip;
    }

    public String getNama() {
        return this.nama;
    }

    public void setNama(String nama) {
        this.nama = nama;
    }

    public String getNip() {
        return this.nip;
    }

    public void setNip(String nip) {
        this.nip = nip;
    }

    public void info() {
        System.out.println("Nama: " + this.nama);
        System.out.println("NIP: " + this.nip);
    }

}
```

```

import java.time.Duration;

import java.time.LocalDate;
import java.time.format.DateTimeFormatter;

public class Peminjaman {

    private Buku buku;
    private Anggota anggota;
    private Petugas petugas;
    private LocalDate tglPinjam;
    private LocalDate tglKembali;
    private int denda;

    public Peminjaman() {}

    public Peminjaman(Buku buku, Anggota anggota, Petugas petugas, String
tglPinjam, String tglKembali, int denda) {
        this.buku = buku;
        this.anggota = anggota;
        this.petugas = petugas;
        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd");
        this.tglPinjam = LocalDate.parse(tglPinjam, formatter);
        this.tglKembali = LocalDate.parse(tglKembali, formatter);
        this.denda = denda;
    }

    public Buku getBuku() {
        return buku;
    }

    public void setBuku(Buku buku) {
        this.buku = buku;
    }

    public Anggota getAnggota() {
        return anggota;
    }

    public Petugas getPetugas() {
        return petugas;
    }

    public void setPetugas(Petugas petugas) {
        this.petugas = petugas;
    }

    public void setAnggota(Anggota anggota) {
        this.anggota = anggota;
    }

    public LocalDate getTglPinjam() {
        return tglPinjam;
    }

    public LocalDate getTglKembali() {
        return tglKembali;
    }

    public int getDenda() {
        return denda;
    }

    public void setTglPinjam(String tglPinjam) {

```

```

public void setTglPinjam(String tglPinjam) {
    DateTimeFormatter formatter =
DateTimeFormatter.ofPattern("yyyy-MM-dd");
    this.tglPinjam = LocalDate.parse(tglPinjam, formatter);
}

    public void setTglKembali(String tglKembali) {
        DateTimeFormatter formatter =
DateTimeFormatter.ofPattern("yyyy-MM-dd");
        this.tglKembali = LocalDate.parse(tglKembali, formatter);
    }
    public void setDenda(int denda) {
        this.denda = denda;
    }
    public int hitungLamaPinjam() {
        Duration diff =
Duration.between(tglPinjam.atStartOfDay(),
tglKembali.atStartOfDay());
        return (int) diff.toDays();
    }
    public int bukuKembali(String bukuKembali) {
        DateTimeFormatter formatter =
DateTimeFormatter.ofPattern("yyyy-MM-dd");
        LocalDate tglPengembalian = LocalDate.parse(bukuKembali,
formatter);
        if (tglPengembalian.isAfter(tglKembali)) {
            int keterlambatan = hitungLamaPinjam();
            return (int) (denda * keterlambatan);
        } else {
            return 0;
        }
    }
    public void cetak() {
        System.out.println("Buku : " + buku.getJudul());
        System.out.println("Anggota : " + anggota.getNama());
        System.out.println("Petugas : " + petugas.getNama());
        System.out.println("Tgl Pinjam : " + tglPinjam);
        System.out.println("Tgl Kembali : " + tglKembali);
        System.out.println("Lama Pinjam : " + hitungLamaPinjam()
+ " hari");
        System.out.println("Denda : " +
bukuKembali(tglKembali.toString()));
    }
}

```

Output:

```

Buku : Koala Kumal
Anggota : Nafis
Petugas : Aryanto
Tgl Pinjam : 2024-01-10
Tgl Kembali : 2024-01-11
Lama Pinjam : 1 hari
Denda : 0

```