

**LAPORAN PRAKTIKUM WEEK 11**  
**PEMROGRAMAN BERORIENTASI OBJEK**  
**“MEMBUAT SISTEM MANAJEMEN PERPUSTAKAAN”**



**Oleh:**  
**Benony Gabriel**  
**105222002**

**PROGRAM STUDI ILMU KOMPUTER**  
**FAKULTAS SAINS DAN ILMU KOMPUTER**  
**UNIVERSITAS PERTAMINA**  
**2024**

## DAFTAR ISI

DAFTAR ISI .....	2
<u>BAB I : PENDAHULUAN</u> .....	3
1.1 Studi Kasus .....	3
1.2 Struktur Direktori.....	3
1.3 Variabel dan Tipe Data .....	4
1.4 Constructor dan Method .....	5
<u>BAB II : DOKUMENTASI DAN PEMBAHASAN</u> .....	7
2.1 Class Book .....	7
2.2 Class Ebook .....	8
2.3 Class Magazine .....	8
2.4 Class LibraryItem .....	9
2.5 Interface Loanable .....	10
2.6 Interface Downloadable .....	10
2.7 Class LibraryMember .....	11
2.8 Class RegularMember .....	12
2.9 Class PremiumMember .....	12
2.10 Output Program .....	14
<u>BAB III : KESIMPULAN</u> .....	16

# **BAB I**

## **PENDAHULUAN**

### **1.1 Studi Kasus**

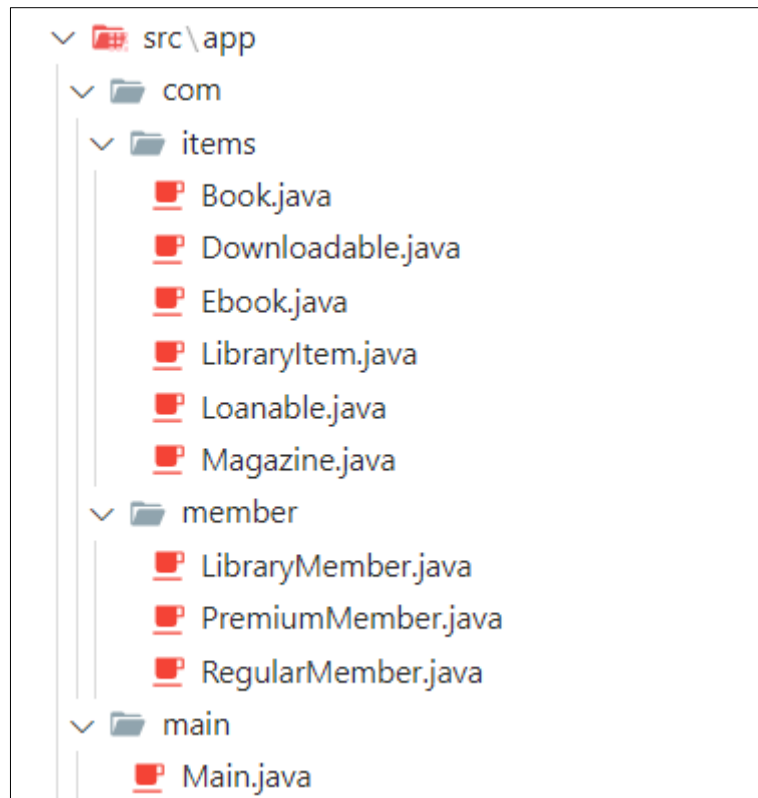
Anda diminta untuk mengembangkan sistem manajemen perpustakaan yang memiliki berbagai jenis bahan pustaka seperti buku, majalah, dan e-book. Setiap bahan pustaka memiliki atribut umum seperti judul, penulis, dan tahun publikasi, serta metode untuk menampilkan informasi. Buku fisik dan majalah dapat dipinjam, sedangkan e-book dapat diunduh. Selain itu, terdapat jenis anggota perpustakaan yang berbeda seperti anggota biasa dan anggota premium dengan hak akses yang berbeda, yaitu dapat mendownload e-book.

Tugas Anda adalah:

- Membuat interface `Loanable` yang mendefinisikan metode `loan()` dan `returnItem()`.
- Membuat abstract class `LibraryItem` yang mendefinisikan atribut dasar dan metode abstrak `displayInfo()`.
- Membuat kelas `Book`, `Magazine`, dan `Ebook` yang masing-masing mewarisi `LibraryItem` dan mengimplementasikan `Loanable` jika berlaku.
- Membuat interface `Downloadable` yang mendefinisikan metode `download()`.
- Membuat kelas `Ebook` yang mengimplementasikan `Downloadable`.
- Membuat abstract class `LibraryMember` yang mendefinisikan atribut dasar dan metode abstrak `borrowItem(LibraryItem item)`.
- Membuat kelas `RegularMember` dan `PremiumMember` yang mewarisi `LibraryMember`.
- Menyusun metode di `LibraryMember` untuk memanfaatkan konsep polymorphism, abstract class, interface, dan object casting.

### **1.2 Struktur Direktori**

Direktori `src/app` merupakan direktori utama yang berisi sub-package `com`. Di dalam `com` terdapat dua sub-package, yaitu `items` dan `member`.



Sub-package items berisi kelas-kelas yang merepresentasikan berbagai jenis bahan pustaka seperti Book, Ebook, dan Magazine. Kelas-kelas ini mewarisi abstract class LibraryItem dan mengimplementasikan interface Loanable jika bisa dipinjam atau Downloadable jika bisa diunduh. Sub-package member berisi kelas-kelas yang merepresentasikan anggota perpustakaan seperti LibraryMember, PremiumMember, dan RegularMember. LibraryMember adalah abstract class yang مخصوص untuk anggota perpustakaan dan diimplementasikan oleh PremiumMember dan RegularMember dengan hak akses yang berbeda. Package main berisi kelas Main yang menjadi titik masuk program.

### 1.3 Variabel dan Tipe Data

Untuk membuat sistem manajemen perpustakaan, kita akan membuat beberapa kelas di dalam package yang sesuai. Berikut adalah daftar variabel yang digunakan dalam program ini beserta tipe datanya dan keterangannya.

Nama Variabel	Tipe Data	Keterangan
Class LibraryItem		
title	String	Judul item perpustakaan
author	String	Penulis item perpustakaan
publicationYear	int	Tahun publikasi item perpustakaan
Class LibraryMember		
name	String	Nama anggota perpustakaan
memberId	String	ID anggota perpustakaan

Class Main		
book	Book	Objek buku yang diinstansiasi
magazine	Magazine	Objek majalah yang diinstansiasi
ebook	Ebook	Objek ebook yang diinstansiasi
regularMember	RegularMember	Objek anggota reguler yang diinstansiasi
premiumMember	PremiumMember	Objek anggota premium yang diinstansiasi

#### 1.4 Constructor dan Method

Berikut adalah daftar metode yang digunakan dalam pengembangan sistem manajemen perpustakaan:

Nama Method	Jenis	Keterangan
Class Book		
Book	Constructor	Konstruktor untuk inisialisasi objek Book
loan	Function	Implementasi metode loan dari interface Loanable
returnItem	Function	Implementasi metode returnItem dari interface Loanable
displayInfo	Function	Implementasi metode abstrak displayInfo
Class Ebook		
Ebook	Constructor	Konstruktor untuk inisialisasi objek Ebook
download	Function	Implementasi metode download dari interface Downloadable
displayInfo	Function	Implementasi metode abstrak displayInfo
Class Magazine		
Magazine	Constructor	Konstruktor untuk inisialisasi objek Magazine
loan	Function	Implementasi metode loan dari interface Loanable
returnItem	Function	Implementasi metode returnItem dari interface Loanable
displayInfo	Function	Implementasi metode abstrak displayInfo
Class LibraryItem		
LibraryItem	Constructor	Konstruktor untuk inisialisasi objek LibraryItem
displayInfo	Function	Metode abstrak untuk menampilkan informasi
Interface Loanable		
loan	Function	Metode untuk meminjam item
returnItem	Function	Metode untuk mengembalikan item
Interface Downloadable		
download	Function	Metode untuk mengunduh item
Class LibraryMember		
LibraryMember	Constructor	Konstruktor untuk inisialisasi objek LibraryMember
borrowItem	Function	Metode abstrak untuk meminjam item

Class RegularMember		
RegularMember	Constructor	Konstruktor untuk inisialisasi objek RegularMember
borrowItem	Function	Implementasi metode abstrak borrowItem
Class Premium Member		
PremiumMember	Constructor	Konstruktor untuk inisialisasi objek PremiumMember
borrowItem	Function	Implementasi metode abstrak borrowItem

## BAB II

### DOKUMENTASI DAN PEMBAHASAN

Pada bagian ini, kita akan membahas lebih jauh terkait setiap constructor dan method yang memainkan peran penting dalam mengelola data item perpustakaan, interaksi antara anggota perpustakaan dan item perpustakaan, serta fungsi tambahan yang mendukung fitur-fitur perpustakaan. Kita akan melihat bagaimana constructor dan method-method tersebut diimplementasikan untuk menangani pembuatan objek item perpustakaan dan anggota perpustakaan. Selain itu, kita akan memeriksa metode-metode yang mengatur bagaimana item perpustakaan dapat dipinjam, dikembalikan, dan diunduh oleh anggota perpustakaan.

#### 2.1 Class Book

A screenshot of a Java IDE window titled 'Book.java'. The code defines a class 'Book' that extends 'LibraryItem' and implements the 'Loanable' interface. The class has a constructor that takes 'title', 'author', and 'publicationYear' as parameters and calls 'super()'. It also has three methods: 'loan()' which prints 'The book [title] has been loaned', 'returnItem()' which prints 'The book [title] has been returned', and 'displayInfo()' which prints 'Book [Title : title Author : author Year: publicationYear]'.

```
package app.com.items;

public class Book extends LibraryItem implements Loanable {
    public Book(String title, String author, int publicationYear){
        super(title, author, publicationYear);
    }

    @Override
    public void loan(){
        System.out.println("The book \"" + title + "\" has been loaned");
    }

    @Override
    public void returnItem(){
        System.out.println("The book \"" + title + "\" has been returned");
    }

    @Override
    public void displayInfo() {
        System.out.println("Book [Title : " + title + " Author : " + author + "Year: " + publicationYear + "]" );
    }
}
```

Kode program di atas mendefinisikan kelas 'Book' yang berada dalam package 'app.com.items'. Kelas 'Book' ini merupakan subclass dari 'LibraryItem' dan mengimplementasikan interface 'Loanable'. Konstruktor kelas 'Book' menerima tiga parameter: 'title', 'author', dan 'publicationYear', yang kemudian diteruskan ke konstruktor superclass 'LibraryItem'. Kelas ini mengimplementasikan tiga metode: 'loan()', 'returnItem()', dan 'displayInfo()'. Metode 'loan()' menampilkan pesan bahwa buku telah dipinjam, metode 'returnItem()' menampilkan pesan bahwa buku telah dikembalikan, dan metode 'displayInfo()' menampilkan informasi tentang buku, termasuk judul, penulis, dan tahun publikasi.

## 2.2 Class Ebook

A screenshot of a code editor window titled 'Ebook.java'. The code defines a class 'Ebook' that extends 'LibraryItem' and implements the 'Downloadable' interface. It includes a constructor and two methods: 'download()' and 'displayInfo()'.

```
package app.com.items;

public class Ebook extends LibraryItem implements Downloadable{
    public Ebook(String title, String author, int publicationYear){
        super(title, author, publicationYear);
    }

    @Override
    public void download() {
        System.out.println("The e-book \" + title + "\" has been downloaded.");
    }

    @Override
    public void displayInfo() {
        System.out.println("Ebook [Title: \" + title + \", Author: \" + author + \", Year: \" + publicationYear + \"]");
    }
}
```

Kode program di atas mendefinisikan kelas 'Ebook' yang berada dalam package 'app.com.items'. Kelas 'Ebook' ini merupakan subclass dari 'LibraryItem' dan mengimplementasikan interface 'Downloadable'. Konstruktor kelas 'Ebook' menerima tiga parameter: 'title', 'author', dan 'publicationYear', yang kemudian diteruskan ke konstruktor superclass 'LibraryItem'. Kelas ini mengimplementasikan dua metode: 'download()' dan 'displayInfo()'. Metode 'download()' menampilkan pesan bahwa e-book telah diunduh, dan metode 'displayInfo()' menampilkan informasi tentang e-book, termasuk judul, penulis, dan tahun publikasi.

## 2.3 Class Magazine

A screenshot of a code editor window titled 'Magazine.java'. The code defines a class 'Magazine' that extends 'LibraryItem' and implements the 'Loanable' interface. It includes a constructor and three methods: 'loan()', 'returnItem()', and 'displayInfo()'.

```
package app.com.items;

public class Magazine extends LibraryItem implements Loanable {
    public Magazine(String title, String author, int publicationYear){
        super(title, author, publicationYear);
    }

    @Override
    public void loan() {
        System.out.println("The magazine \" + title + "\" has been loaned.");
    }

    @Override
    public void returnItem() {
        System.out.println("The magazine \" + title + "\" has been returned");
    }

    @Override
    public void displayInfo() {
        System.out.println("Magazine [Title: \" + title + \", Author: \" + author + \", Year: \" + publicationYear + \"]");
    }
}
```

Kode program di atas mendefinisikan kelas 'Magazine' yang berada dalam package 'app.com.items'. Kelas 'Magazine' ini merupakan subclass dari



`LibraryItem` dan mengimplementasikan interface `Loanable`. Konstruktor kelas `Magazine` menerima tiga parameter: `title`, `author`, dan `publicationYear`, yang kemudian diteruskan ke konstruktor superclass `LibraryItem`. Kelas ini mengimplementasikan tiga metode: `loan()`, `returnItem()`, dan `displayInfo()`. Metode `loan()` menampilkan pesan bahwa majalah telah dipinjam, metode `returnItem()` menampilkan pesan bahwa majalah telah dikembalikan, dan metode `displayInfo()` menampilkan informasi tentang majalah, termasuk judul, penulis, dan tahun publikasi.

## 2.4 Class LibraryItem

```
package app.com.items;

public abstract class LibraryItem {
    public String title;
    public String author;
    public int publicationYear;

    public LibraryItem(String title, String author, int publicationYear) {
        this.title = title;
        this.author = author;
        this.publicationYear = publicationYear;
    }

    public abstract void displayInfo();
}
```

Kode program di atas mendefinisikan kelas abstrak bernama `LibraryItem` yang berada dalam package `app.com.items`. Kelas ini berfungsi sebagai superclass untuk berbagai jenis item perpustakaan seperti buku, ebook, dan majalah. Kelas `LibraryItem` memiliki tiga variabel instance: `title`, `author`, dan `publicationYear`, yang berturut-turut merupakan judul item, penulis item, dan tahun publikasi item. Konstruktornya menerima tiga parameter (`title`, `author`, dan `publicationYear`) dan menginisialisasi variabel instance dengan nilai-nilai yang diterima.

Selain itu, kelas ini mendeklarasikan satu metode abstrak `displayInfo()`, yang berarti setiap subclass dari `LibraryItem` harus menyediakan implementasi konkret untuk metode ini. Metode `displayInfo()` bertujuan untuk menampilkan informasi tentang item perpustakaan. Karena `LibraryItem` adalah kelas abstrak, ia tidak dapat diinstansiasi langsung dan hanya dapat digunakan sebagai superclass untuk kelas-kelas lain yang lebih spesifik.

## 2.5 Interface Loanable



```
package app.com.items;

// Membuat interface Loanable
public interface Loanable {

    void loan();
    void returnItem();

}
```

Kode program di atas mendefinisikan interface bernama 'Loanable' yang berada dalam package 'app.com.items'. Interface ini mendeklarasikan dua metode: 'loan()' dan 'returnItem()'.

Metode 'loan()' bertujuan untuk menangani proses peminjaman item, sedangkan metode 'returnItem()' bertujuan untuk menangani proses pengembalian item.

Interface 'Loanable' tidak memberikan implementasi konkret untuk kedua metode tersebut. Sebaliknya, setiap kelas yang mengimplementasikan interface 'Loanable' harus menyediakan implementasi konkret untuk kedua metode tersebut. Interface ini digunakan untuk menentukan kontrak atau blueprint yang harus dipenuhi oleh kelas-kelas yang mengimplementasikannya, memastikan bahwa kelas-kelas tersebut dapat dipinjam dan dikembalikan.

## 2.6 Interface Downloadable



```
package app.com.items;

// Membuat interface Downloadable
public interface Downloadable {

    void download();

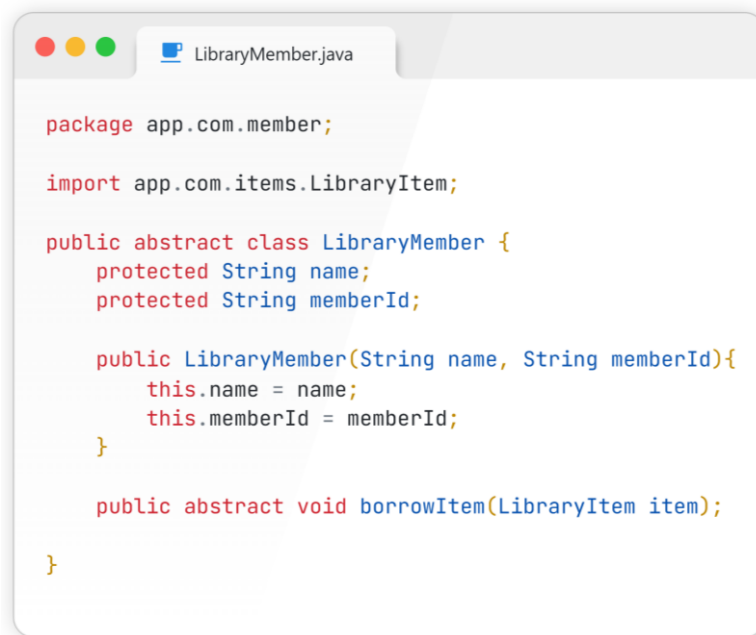
}
```

Kode program di atas mendefinisikan interface 'Downloadable' yang berada dalam package 'app.com.items'. Interface ini mendeklarasikan satu metode: 'download()'.

Metode 'download()' bertujuan untuk menangani proses pengunduhan item. Interface 'Downloadable' tidak memberikan implementasi konkret untuk metode ini. Sebaliknya, setiap kelas yang mengimplementasikan interface 'Downloadable' harus menyediakan implementasi konkret untuk metode 'download()'.

Interface ini digunakan untuk menentukan kontrak atau blueprint yang harus dipenuhi oleh kelas-kelas yang mengimplementasikannya, memastikan bahwa kelas-kelas tersebut mendukung fitur pengunduhan. Contoh kelas yang mungkin mengimplementasikan interface ini adalah kelas `Ebook`, yang memungkinkan e-book diunduh oleh pengguna.

## 2.7 Class LibraryMember



```
package app.com.member;

import app.com.items.LibraryItem;

public abstract class LibraryMember {
    protected String name;
    protected String memberId;

    public LibraryMember(String name, String memberId){
        this.name = name;
        this.memberId = memberId;
    }

    public abstract void borrowItem(LibraryItem item);
}
```

Kode program di atas mendefinisikan kelas abstrak `LibraryMember` yang berada dalam package `app.com.member`. Kelas ini berfungsi sebagai superclass untuk berbagai jenis anggota perpustakaan. Kelas `LibraryMember` memiliki dua variabel instance: `name` dan `memberId`, yang masing-masing merupakan nama anggota perpustakaan dan ID anggota perpustakaan.

Konstruktornya menerima dua parameter (`name` dan `memberId`) dan menginisialisasi variabel instance dengan nilai-nilai yang diterima.

Kelas ini juga mendeklarasikan satu metode abstrak `borrowItem(LibraryItem item)`, yang berarti setiap subclass dari `LibraryMember` harus menyediakan implementasi konkret untuk metode ini. Metode `borrowItem()` bertujuan untuk menangani proses peminjaman item perpustakaan oleh anggota perpustakaan. Karena `LibraryMember` adalah kelas abstrak, ia tidak dapat diinstansiasi langsung dan hanya dapat digunakan sebagai superclass untuk kelas-kelas lain yang lebih spesifik seperti `RegularMember` dan `PremiumMember`.

## 2.8 Class RegularMember



```
package app.com.member;

import app.com.items.LibraryItem;
import app.com.items.Loanable;

public class RegularMember extends LibraryMember {
    public RegularMember(String name, String memberId) {
        super(name, memberId);
    }

    @Override
    public void borrowItem(LibraryItem item) {
        if (item instanceof Loanable) {
            ((Loanable) item).loan();
            System.out.println("Regular member \"" + name + "\" borrowed: " + item.title);
        } else {
            System.out.println("Regular members cannot borrow this item.");
        }
    }
}
```

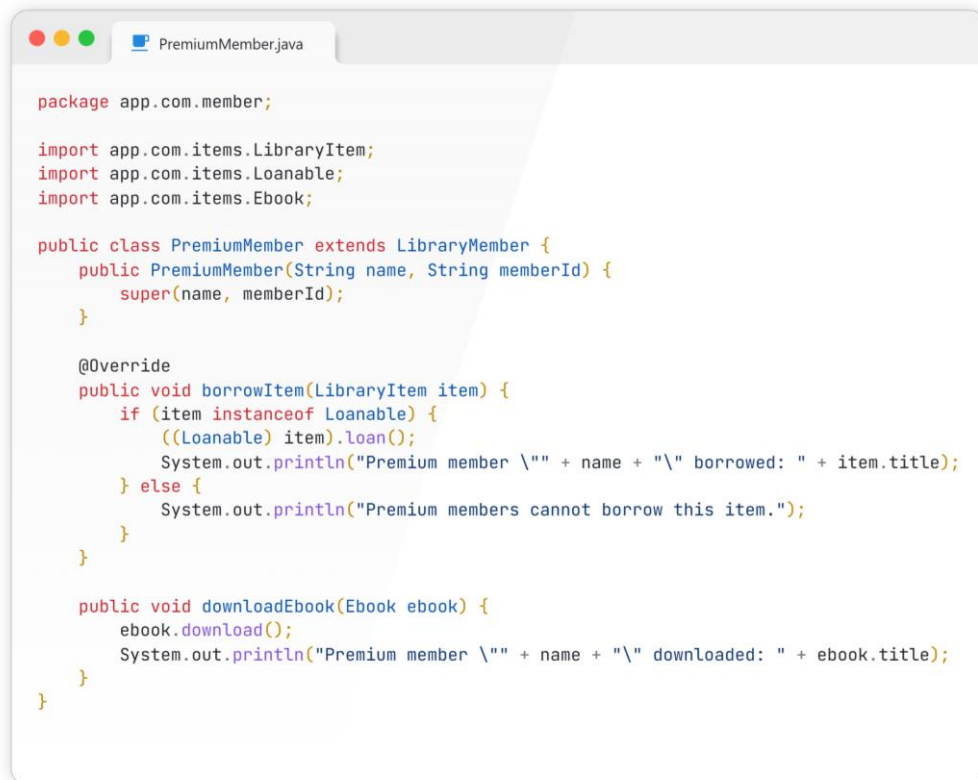
Kode program di atas mendefinisikan kelas bernama `RegularMember` yang berada dalam package `app.com.member`. Kelas ini adalah subclass dari `LibraryMember` dan mengimplementasikan metode abstrak `borrowItem()` yang dideklarasikan dalam kelas `LibraryMember`.

Konstruktornya menerima dua parameter (`name` dan `memberId`) dan meneruskan nilai-nilai ini ke konstruktor superclass `LibraryMember` untuk menginisialisasi variabel instance `name` dan `memberId`.

Metode `borrowItem(LibraryItem item)` diimplementasikan untuk menangani proses peminjaman item perpustakaan oleh anggota reguler. Metode ini memeriksa apakah item yang akan dipinjam merupakan instance dari interface `Loanable`. Jika iya, metode `loan()` dari interface `Loanable` dipanggil untuk meminjam item tersebut, dan sebuah pesan dicetak ke layar yang menunjukkan bahwa anggota reguler dengan nama tertentu telah meminjam item tersebut. Jika item tidak bisa dipinjam (bukan instance dari `Loanable`), pesan akan dicetak yang menyatakan bahwa anggota reguler tidak dapat meminjam item tersebut.

## 2.9 Class Premium Member

Kelas ini adalah subclass dari `LibraryMember` dan mengimplementasikan metode abstrak `borrowItem()` yang dideklarasikan dalam kelas `LibraryMember`.



```
package app.com.member;

import app.com.items.LibraryItem;
import app.com.items.Loanable;
import app.com.items.Ebook;

public class PremiumMember extends LibraryMember {
    public PremiumMember(String name, String memberId) {
        super(name, memberId);
    }

    @Override
    public void borrowItem(LibraryItem item) {
        if (item instanceof Loanable) {
            ((Loanable) item).loan();
            System.out.println("Premium member \"" + name + "\" borrowed: " + item.title);
        } else {
            System.out.println("Premium members cannot borrow this item.");
        }
    }

    public void downloadEbook(Ebook ebook) {
        ebook.download();
        System.out.println("Premium member \"" + name + "\" downloaded: " + ebook.title);
    }
}
```

Konstruktornya menerima dua parameter (`name` dan `memberId`) dan meneruskan nilai-nilai ini ke konstruktor superclass `LibraryMember` untuk menginisialisasi variabel instance `name` dan `memberId`.

Metode `borrowItem(LibraryItem item)` diimplementasikan untuk menangani proses peminjaman item perpustakaan oleh anggota premium. Metode ini mirip dengan metode yang didefinisikan dalam kelas `RegularMember`, namun juga menambahkan kemampuan untuk memeriksa apakah item yang akan dipinjam merupakan instance dari interface `Loanable`. Jika iya, metode `loan()` dari interface `Loanable` dipanggil untuk meminjam item tersebut, dan sebuah pesan dicetak ke layar yang menunjukkan bahwa anggota premium dengan nama tertentu telah meminjam item tersebut. Jika item tidak bisa dipinjam (bukan instance dari `Loanable`), pesan akan dicetak yang menyatakan bahwa anggota premium tidak dapat meminjam item tersebut.

Selain metode `borrowItem()`, kelas `PremiumMember` juga memiliki metode tambahan `downloadEbook(Ebook ebook)`. Metode ini menerima objek `Ebook` sebagai parameter dan memanggil metode `download()` dari objek tersebut untuk mengunduh e-book. Setelah itu, pesan akan dicetak ke layar yang menunjukkan bahwa anggota premium dengan nama tertentu telah mengunduh e-book tersebut.

## 2.10 Output Program

```
Book [Title : Effective Java Author : Joshua BlochYear: 2008]
Magazine [Title: National Geographic, Author: Various, Year: 2021]
Ebook [Title: Clean Code, Author: Robert C. Martin, Year: 2008]
The book "Effective Java" has been loaned
Regular member "Alice" borrowed: Effective Java
The magazine "National Geographic" has been loaned.
Regular member "Alice" borrowed: National Geographic
Regular members cannot borrow this item.
The book "Effective Java" has been loaned
Premium member "Bob" borrowed: Effective Java
The magazine "National Geographic" has been loaned.
Premium member "Bob" borrowed: National Geographic
The e-book "Clean Code" has been downloaded.
Premium member "Bob" downloaded: Clean Code
```

Output program ini memberikan informasi tentang item-item perpustakaan yang telah dibuat dan interaksi yang terjadi antara anggota perpustakaan dengan item-item tersebut. Berikut penjelasan dari setiap baris output:

- Book [Title : Effective Java Author : Joshua BlochYear: 2008]: Informasi tentang buku yang telah dibuat, berjudul "Effective Java", ditulis oleh "Joshua Bloch", dan diterbitkan pada tahun 2008.
- Magazine [Title: National Geographic, Author: Various, Year: 2021]: Informasi tentang majalah yang telah dibuat, berjudul "National Geographic", memiliki berbagai penulis, dan diterbitkan pada tahun 2021.
- Ebook [Title: Clean Code, Author: Robert C. Martin, Year: 2008]: Informasi tentang e-book yang telah dibuat, berjudul "Clean Code", ditulis oleh "Robert C. Martin", dan diterbitkan pada tahun 2008.
- The book "Effective Java" has been loaned: Pesan bahwa buku "Effective Java" telah dipinjam.
- Regular member "Alice" borrowed: Effective Java: Informasi bahwa anggota reguler dengan nama "Alice" telah meminjam buku "Effective Java".
- The magazine "National Geographic" has been loaned.: Pesan bahwa majalah "National Geographic" telah dipinjam.
- Regular member "Alice" borrowed: National Geographic: Informasi bahwa anggota reguler dengan nama "Alice" telah meminjam majalah "National Geographic".
- Regular members cannot borrow this item.: Pesan bahwa anggota reguler tidak dapat meminjam item tersebut (e-book).
- The book "Effective Java" has been loaned: Pesan bahwa buku "Effective Java" telah dipinjam.

- Premium member "Bob" borrowed: Effective Java: Informasi bahwa anggota premium dengan nama "Bob" telah meminjam buku "Effective Java".
- The magazine "National Geographic" has been loaned.: Pesan bahwa majalah "National Geographic" telah dipinjam.
- Premium member "Bob" borrowed: National Geographic: Informasi bahwa anggota premium dengan nama "Bob" telah meminjam majalah "National Geographic".
- The e-book "Clean Code" has been downloaded.: Pesan bahwa e-book "Clean Code" telah diunduh.
- Premium member "Bob" downloaded: Clean Code: Informasi bahwa anggota premium dengan nama "Bob" telah mengunduh e-book "Clean Code".

### **BAB III**

### **KESIMPULAN**

Kesimpulan dari keseluruhan sistem manajemen perpustakaan ini adalah sebagai berikut:

- **Struktur Kelas dan Interface:** Sistem ini menggunakan konsep pemrograman berorientasi objek dengan baik, dimana terdapat kelas-kelas yang mewakili berbagai jenis item perpustakaan (seperti buku, majalah, dan e-book) serta kelas-kelas anggota perpustakaan (reguler dan premium). Penggunaan interface juga memungkinkan untuk mendefinisikan perilaku tertentu, seperti kemampuan meminjam item atau mengunduh item.
- **Polimorfisme:** Adanya penggunaan polimorfisme memungkinkan penggunaan metode-metode yang sama pada berbagai jenis item perpustakaan, meskipun perilaku yang spesifik dapat bervariasi antara kelas-kelas tersebut. Contohnya adalah metode `displayInfo()` yang diimplementasikan dalam setiap kelas item perpustakaan, meskipun perilaku penampilannya berbeda tergantung pada jenis itemnya.
- **Inheritance:** Penggunaan inheritance memungkinkan untuk mengelompokkan fitur-fitur bersama dalam superclass dan mewarisi fitur-fitur tersebut ke subclass yang lebih spesifik. Misalnya, kelas `Book`, `Ebook`, dan `Magazine` mewarisi fitur-fitur dasar dari kelas `LibraryItem`.
- **Abstraksi dan Encapsulation:** Penggunaan kelas abstrak dan variabel instance yang bersifat `protected` memberikan tingkat abstraksi yang baik, dimana detail implementasi dari kelas-kelas tersebut dapat disembunyikan dan hanya fitur-fitur yang relevan untuk penggunaan luar yang terlihat.
- **Interaksi Antara Kelas-Kelas:** Sistem ini memungkinkan interaksi yang baik antara anggota perpustakaan dengan item-item perpustakaan. Fitur peminjaman dan pengunduhan diimplementasikan dengan baik, memperhitungkan jenis anggota perpustakaan (reguler atau premium) serta jenis item yang dapat dipinjam atau diunduh.

Dengan demikian, keseluruhan sistem manajemen perpustakaan ini memberikan struktur yang kokoh dan fleksibel untuk mengelola item-item perpustakaan dan interaksi antara anggota perpustakaan dengan item-item tersebut.