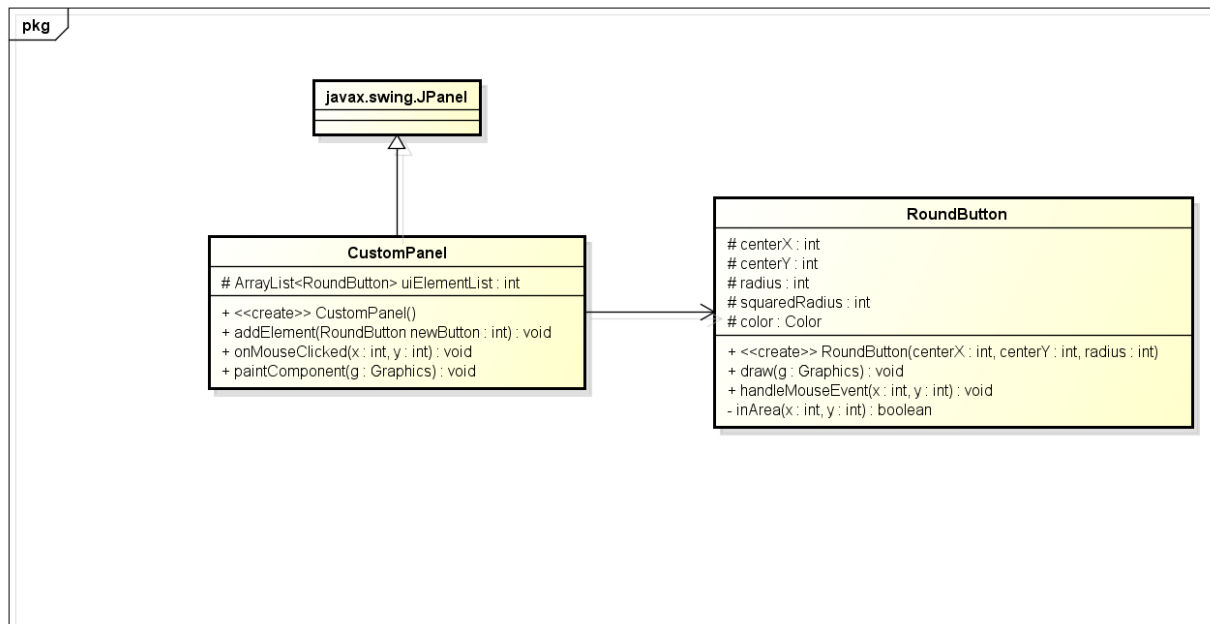


# Open-Closed Principle

Pada responsi kali ini kita akan mempelajari sebuah contoh desain kelas untuk menjelaskan prinsip Open-Closed. Salah satu contoh dalam Java yang menerapkan prinsip ini dengan cukup baik adalah API GUI milik Java. Oleh karena itu pada responsi ini kita akan membuat versi sederhana dari elemen-elemen GUI Java. Karena contoh pada responsi kali ini berfokus pada elemen UI, maka pada tugas kali ini kita hanya akan memiliki bagian View dari pemodelan MVC. **Anda akan perlu untuk membaca Javadoc dari kelas Graphics untuk dapat mengerjakan soal-soal pada responsi ini!**

Pada responsi ini kita akan membuat sebuah Panel dengan nama CustomPanel yang dapat kita isi dengan beberapa jenis elemen UI seperti tombol dan label. Pada awal pembuatan, kita hanya akan memiliki satu buah elemen UI yaitu tombol berbentuk lingkaran (RoundButton). Kita akan mengasumsikan bahwa pada pengembangan selanjutnya akan ada elemen-elemen UI lainnya. Untuk memudahkan pembuatan, kita akan mengasumsikan bahwa tombol-tombol yang dibuat hanya akan berubah warna pada saat di-*click*.

Perhatikanlah diagram kelas di bawah ini! Dengan asumsi bahwa kita akan menambahkan elemen UI baru, apakah desain kelas ini memenuhi prinsip Open-Closed? Pada saat kita menambahkan kelas baru untuk elemen UI lain (contoh : SquareButton), kelas manakah di bawah ini yang juga harus kita modifikasi agar kita dapat menambahkan sebuah objek SquareButton ke dalam objek CustomPanel kita?

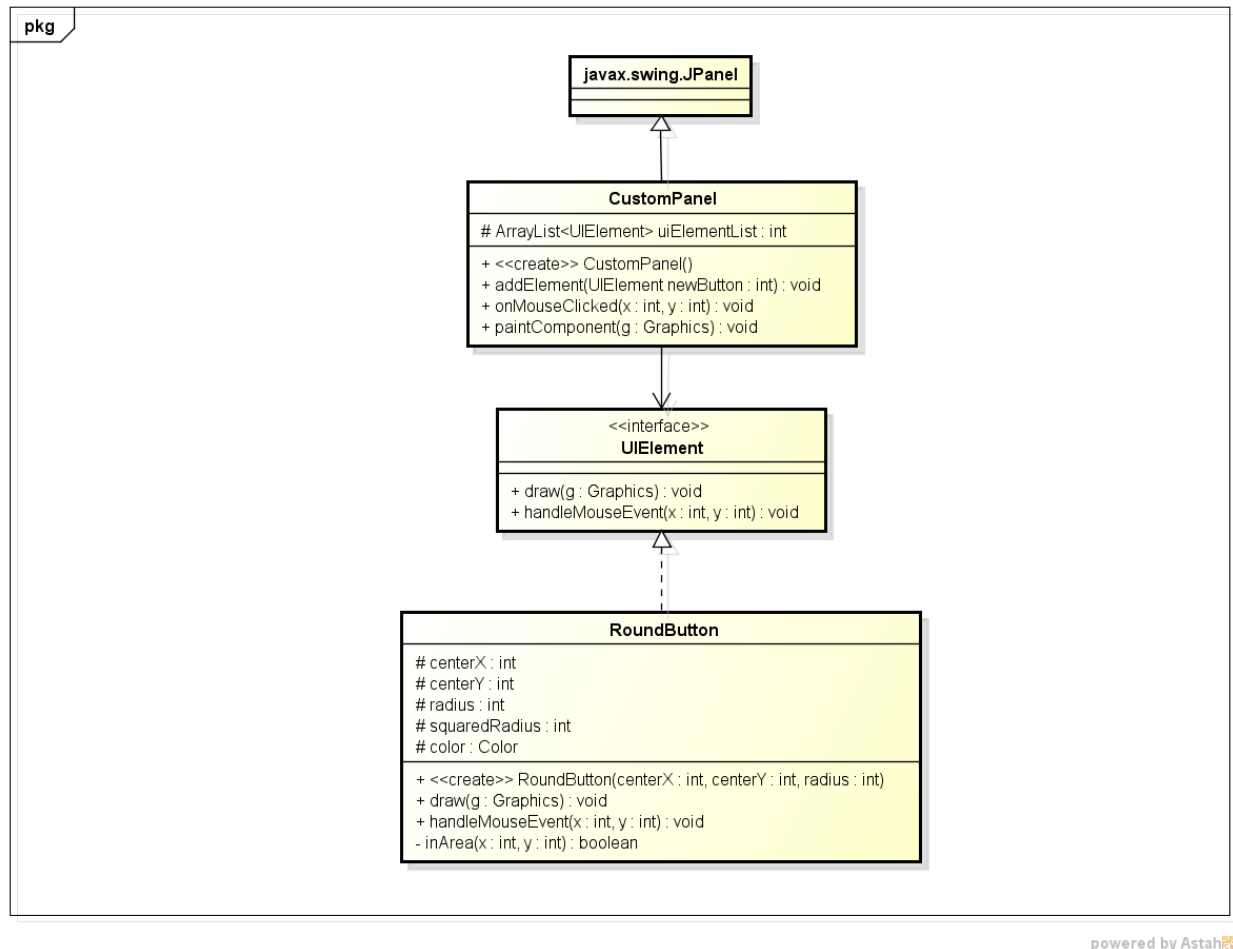


powered by Astah

Gambar 1 Rancangan kelas yang tidak memenuhi prinsip Open-Closed

Pada rancangan kelas di atas, pada saat kita menambahkan elemen UI baru (contoh : SquareButton), maka kita harus melakukan modifikasi pada kelas CustomPanel. Ini berarti bahwa kelas CustomPanel tidak bersifat **closed for modification**.

Sebagai alternatifnya, di bawah ini adalah rancangan kelas yang lebih memenuhi prinsip Open-Closed.



**Gambar 2 Desain Kelas 2**

Apakah desain kelas yang kedua ini memungkinkan kita menambahkan tipe elemen UI baru (apakah desain kelas ini open for extension) ? Apakah desain kelas ini mengharuskan kita memodifikasi kelas CustomPanel maupun kelas-kelas lainnya (selain Tester) pada saat kita menambahkan tipe elemen UI baru (apakah desain kelas ini closed for modification) ?

Seperti dapat dilihat di atas, prinsip Open-Closed dapat kita penuhi dengan melakukan prinsip-prinsip dasar OO dengan baik dan dengan penggunaan pattern-pattern. Dapatkah anda menemukan dimana letak pattern Observer pada desain kelas di atas?

Untuk memudahkan anda mengerjakan tugas kali ini, anda telah diberikan kode lengkap dari kelas CustomPanel, RoundButton, interface UIElement, dan sebuah kelas Tester. Tester yang diberikan akan

membuat sebuah window berukuran 600 x 600 pixel yang akan berisi elemen-elemen UI yang anda tambahkan ke dalam objek CustomPanel-nya. **Pahamilah cara kerja seluruh kelas-kelas yang diberikan agar anda dapat mengerjakan kedua tugas di bawah ini!**

## R1201xxyyy.jar

Tambahkan 2 kelas untuk merepresentasikan elemen-elemen UI berikut ini dan ujliah dalam Tester yang diberikan:

### 1. SquareButton



Gambar 3 Contoh Tampilan SquareButton

Elemen UI ini akan menampilkan sebuah persegi panjang berwarna biru dengan posisi dan ukuran sesuai dengan parameter yang diberikan pada konstruktornya. Warna dari elemen UI ini akan berganti-ganti antara biru dan merah tiap kali dia di-*click*.

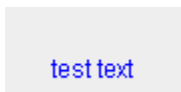
Konstruktur dari kelas ini akan memiliki header :

```
public SquareButton(int topLeftX, int topLeftY,int width,int height)
```

Parameter topLeftX dan topLeftY adalah nilai koordinat X dan Y dari pojok kiri atas elemen UI ini.

Anda dapat menggunakan method fillRect pada kelas Graphics untuk menggambar tampilan dari elemen ini pada CustomPanel. Anda boleh menambahkan atribut/method lain yang anda butuhkan pada kelas ini.

### 2. Label



Gambar 4 Contoh Tampilan Label

Elemen UI ini akan menampilkan sebuah teks pada posisi tertentu. Elemen UI ini tidak akan bereaksi bila di-*click*. Konstruktur dari kelas ini akan memiliki header :

```
public Label (int bottomLeftX, int bottomLeftY,int width,String text)
```

Parameter bottomLeftX dan bottomLeftY adalah nilai koordinat X dan Y dari pojok kiri bawah elemen UI ini.

Carilah method yang dapat anda gunakan untuk menuliskan teks pada kelas Graphics! Anda boleh menambahkan atribut/method lain yang anda butuhkan pada kelas ini.

## R1202xxyyy.jar

### RightTriangleButton



Gambar 5 Contoh Tampilan RightTriangleButton

Tambahkan sebuah kelas dengan nama RightTriangleButton untuk menampilkan elemen UI berupa sebuah tombol berbentuk segitiga siku-siku seperti pada gambar yang diberikan. Warna dari elemen UI ini akan berganti-ganti antara biru dan merah tiap kali dia di-*click*. Konstruktors dari kelas ini akan memiliki header :

```
public RightTriangleButton (int bottomLeftX, int bottomLeftY,int width,int height)
```

Parameter bottomLeftX dan bottomLeftY adalah nilai koordinat X dan Y dari pojok kiri bawah elemen UI ini. Anda boleh menambahkan atribut/method lain yang anda butuhkan pada kelas ini.

Anda dapat menggambar sebuah segitiga dengan menggunakan method fillPolygon pada kelas Graphics.