LABPRO

LABPRO

LABPRO

LABPRO

LABPRO

LABPRO

#

Responsi 6

Praktikum Pemrograman Berbasis Objek

Asisten IF2210 2022/2023

Outline

Exception

Exception

- Mirip dengan C++, Java juga memiliki Exception yang dapat di throw dan catch.
- Namun, object yang di-throw harus merupakan anak dari kelas Throwable. Anak kelas Throwable yang paling mudah digunakan adalah Exception

Contoh Kelas Exception

```
class IndexOutOfBoundException extends Exception {
  public IndexOutOfBoundException(String msg) {
    super(msg);
  }
}
```

Contoh Penggunaan Exception

```
class Vector {
  ...
  public void push(int val) { ... }

  public int get(int idx) throws
IndexOutOfBoundException {
    if (idx < 0 || idx >= this.size) {
      String msg = String.format("ERROR: The
index %d you're trying to access is
inaccessible", idx);
      throw new IndexOutOfBoundException(msg);
    }
    return this.data[idx];
  }
}
```

```
class Main {
  public static void main(String[] args) {
    Vector v = new Vector(10);
    v.push(19);
    v.push(29);

    try {
      System.out.println(v.get(1));
      System.out.println(v.get(2));
    } catch (Exception e) {
      e.printStackTrace();
    }
  }
}
```

Exception

- Pada bagian try … catch, perhatikan bahwa kelas yang di-catch adalah kelas Exception.
- Anda tidak harus membuat catch untuk setiap exception yang ada, melainkan bisa langsung melakukan catch terhadap kelas Exception untuk menangkap semua jenis exception yang mungkin muncul. **(ingat kembali materi polymorphism)**

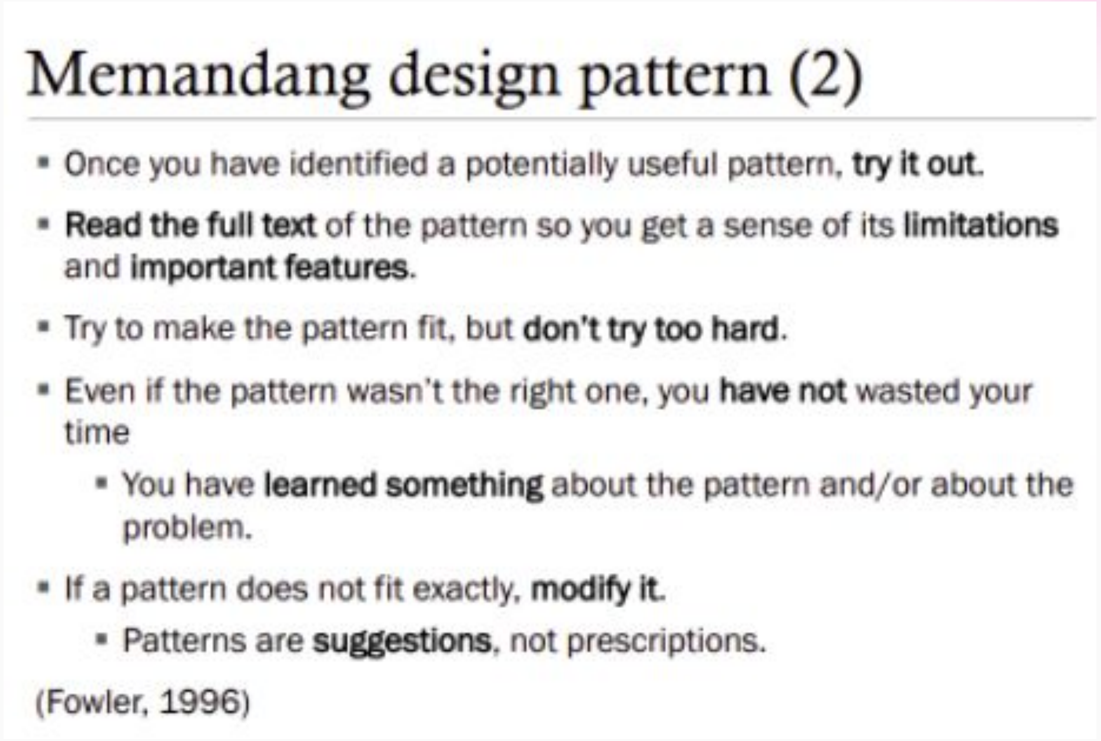
Design Pattern

Design patterns are typical solutions to common problems in software design.

Each pattern is like a blueprint that you can customize to solve a particular design problem in your code.

- refactoring.guru

Design Pattern



Memandang design pattern (2)

- Once you have identified a potentially useful pattern, **try it out**.
- **Read the full text** of the pattern so you get a sense of its **limitations** and **important features**.
- Try to make the pattern fit, but **don't try too hard**.
- Even if the pattern wasn't the right one, you **have not** wasted your time
  - You have **learned something** about the pattern and/or about the problem.
- If a pattern does not fit exactly, **modify it.**
  - Patterns are **suggestions**, not prescriptions.

(Fowler, 1996)

Mempelajari design pattern juga berguna untuk membantu kalian dalam mengerjakan tubes 2 :)

Design Pattern

Berikut adalah beberapa contoh design pattern untuk dipelajari:

- Adapter
- Publisher Subscriber
- Decorator
- Observer
- ...and more

Referensi Design Pattern

Beberapa link ini bisa jadi tambahan bacaan sebelum tidur :)

- https://sourcemaking.com/
- https://github.com/kamranahmedse/design-patterns-for-humans

Sekian.

Ditunggu praktikum dan tutorial berikutnya.