Pola Desain Perangkat Lunak

[Week] 3 – Creational Pattern, Singleton
Prepared by: Tifanny Nabarian

Design Patterns Category

In general, a pattern has four essential elements:

- The **pattern name** is a handle we can use to describe a design problem, its solutions, and consequences in a word or two.
- The **problem** describes when to apply the pattern.
- The **solution relationships**, describes the elements responsibilities, and that make up collaborations.
- The **consequences** are the results and trade-offs of applying the pattern.

Design Patterns Category

Creational Patterns	Creational patterns prescribe the way that objects are created.
Structural Patterns	• Structural patterns are concerned with how classes and objects are composed to form larger structures
Behavioral Patterns	• Behavioral patterns are concerned with algorithms and the assignment of responsibilities between objects.
Concurrency Patterns	• Concurrency patterns prescribe the way access to shared resources is coordinated or sequenced

Design Patterns Scope

		Purpose		
		Creational	Structural	Behavioral
Scope	Class	• Factory method	• Adapter	• Interpreter • Template method
	Object	• Abstract factory • Builder • Prototype • Singleton	• Adapter • Bridge • Composite • Decorator • Fasad • Flyweight • Proxy	• Chain of responsibility • Command • Iterator • Mediator • Memento • Observer • State • Strategy • Visitor

Creational Pattern

Singleton

"

وَمَا خَلَقْتُ الْجِنَّ وَالْإِنْسَ إِلَّا لِيَعْبُدُونِ

Dan aku tidak menciptakan jin dan manusia melainkan supaya mereka mengabdi kepada-Ku.

QS. Az-Dzariyat Ayat 56

Creational Pattern

- Fokus pada proses instansiasi objek
- Membantu sebuah sistem independen mulai dari diciptakan, disusun dan direpresentasikan
- Menyembunyikan informasi mengenai:
  - Kelas Konkret yang digunakan oleh sistem
  - Bagaimana instansiasi dari setiap kelas dibentuk dan diciptakan bersama
- Memberikan fleksibilitas terhadap objek yang diciptakan, siapa yang menciptakan serta bagaimana dan kapan objek diciptakan

Singleton Concept

Definisi GoF

- Memastikan setiap kelas hanya memiliki satu *instance,* dan menyediakan poin akses global terhadap *instance* tersebut.

Konsep

- Sebuah kelas tidak dapat memiliki *multiple instances.* Sekali objek diciptakan, untuk selanjutnya maka hanya menggunakan instansiasi satu objek tersebut. Cara ini membantu untuk menghalangi penciptaan objek yang tidak dibutuhkan pada sistem yang tesentralisasi sehingga memudahkan untuk proses *maintenance*.

Singleton – Real World Example

Singleton – Computer World Example

In some specific software systems, you may prefer to use only **one file system** for the **centralized management of resources**. Also, this pattern can implement a **caching** mechanism.

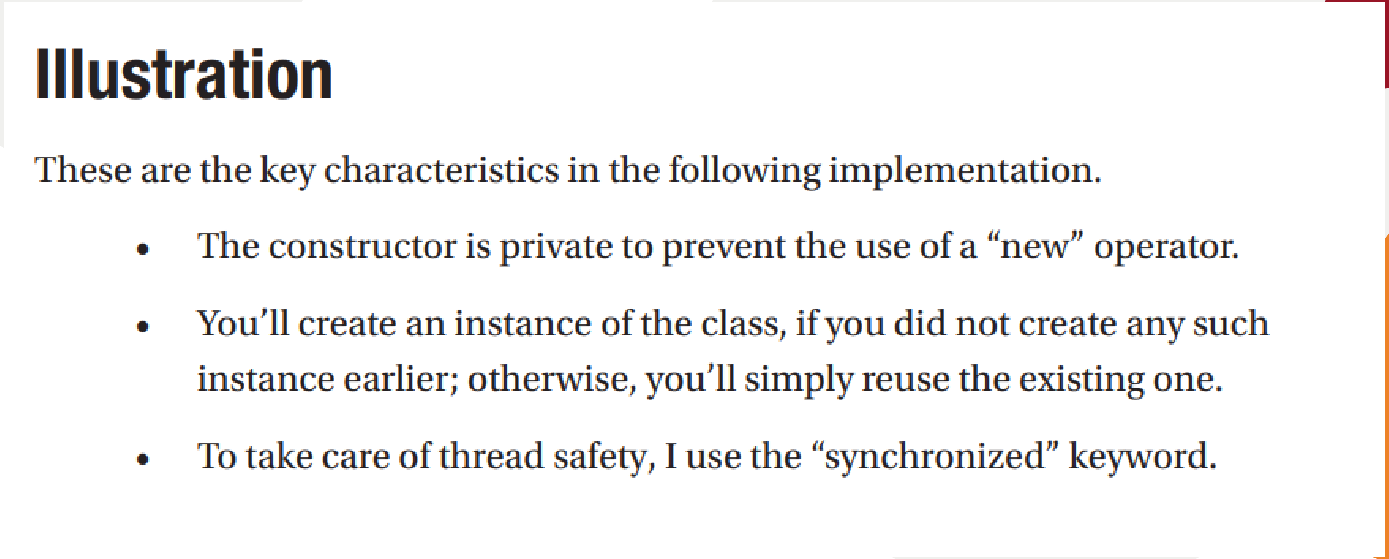
CHARACTER BUILDING CAMPUS

STT - NF

NURUL FIKRI

Any questions?

Let’s Play with the code!



Illustration

These are the key characteristics in the following implementation.

- The constructor is private to prevent the use of a “new” operator.
- You’ll create an instance of the class, if you did not create any such instance earlier; otherwise, you’ll simply reuse the existing one.
- To take care of thread safety, I use the “synchronized” keyword.

Eager Inializations VS Lazy Inializations

Eager Inializations

Pros

- It is straightforward and cleaner.
- It is the opposite of lazy initialization but still thread safe.
- It has a small lag time when the application is in execution mode because everything is already loaded in memory.

Cons

The application takes longer to start (compared to lazy initialization) because everything needs to be loaded first

Any discussions?

Thank You!

CHARACTER BUILDING CAMPUS
STT - NF
NURUL FIKRI