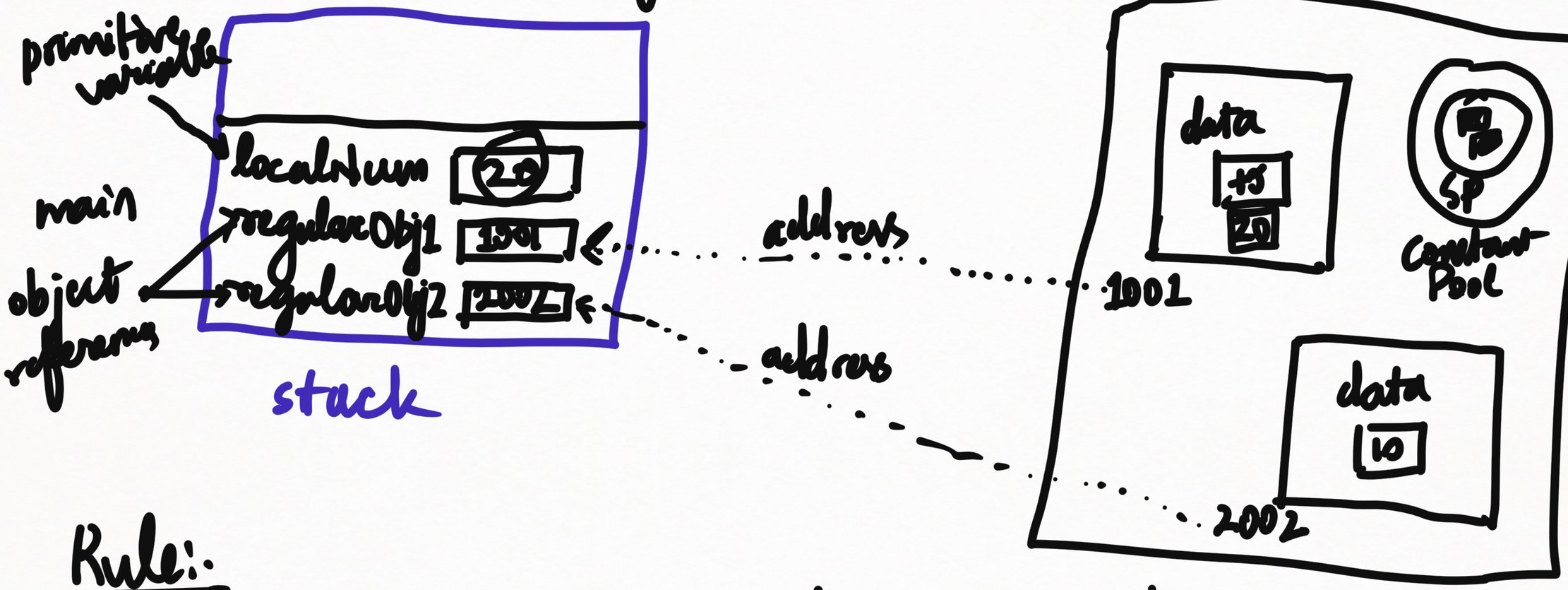
Regular class

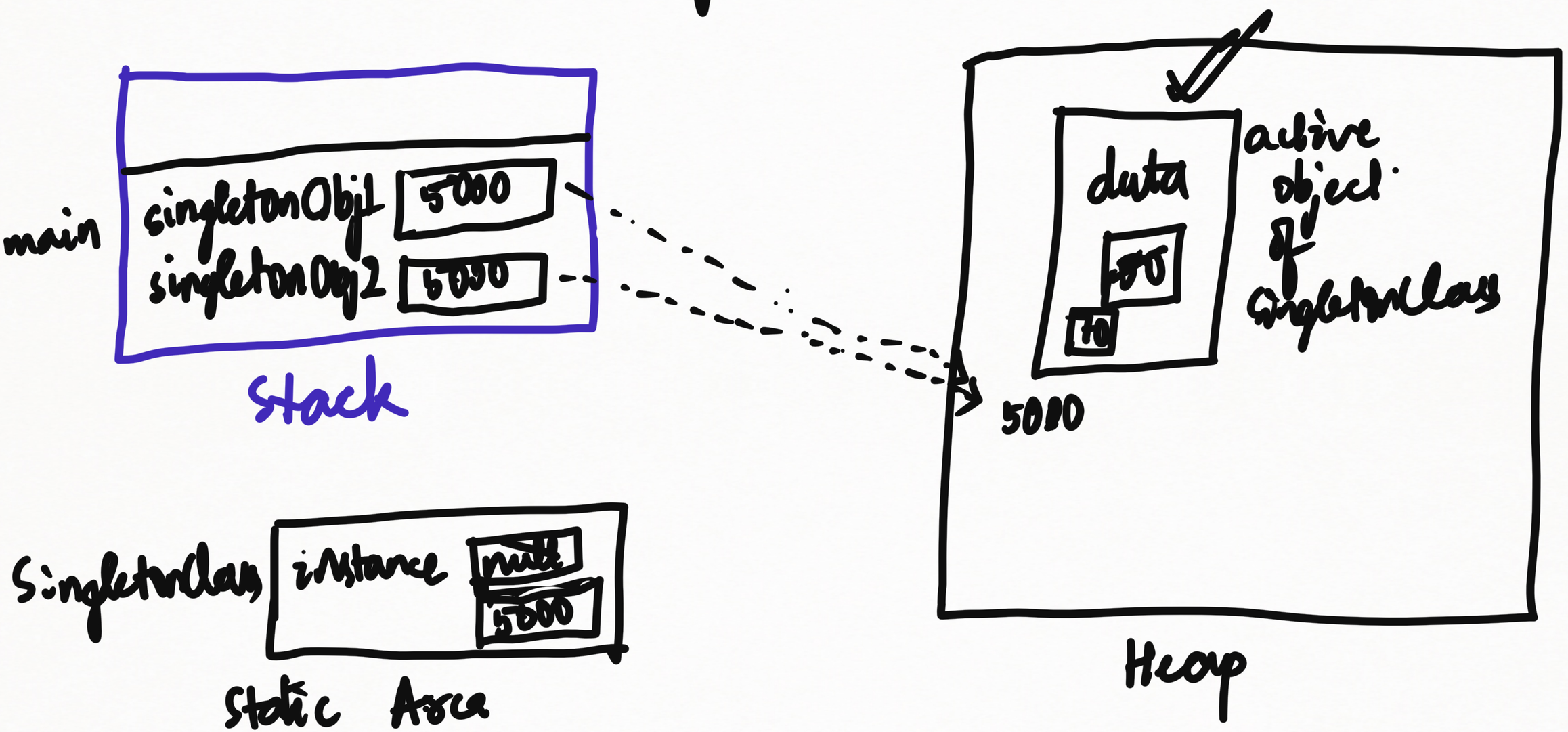


Rule:

Local variables are created in stack.

Instance variables reside in the heap memory(in objects)

Singleton Class



Use-cases of Singleton :-

i) Memory saving

ii) Data-sharing/synchronization

iii) can use an object repeatedly

Example

→ Database Connection

Limitations

i) can be unreliable in multi-threaded usage

ii) can be used unnecessarily to make things complex

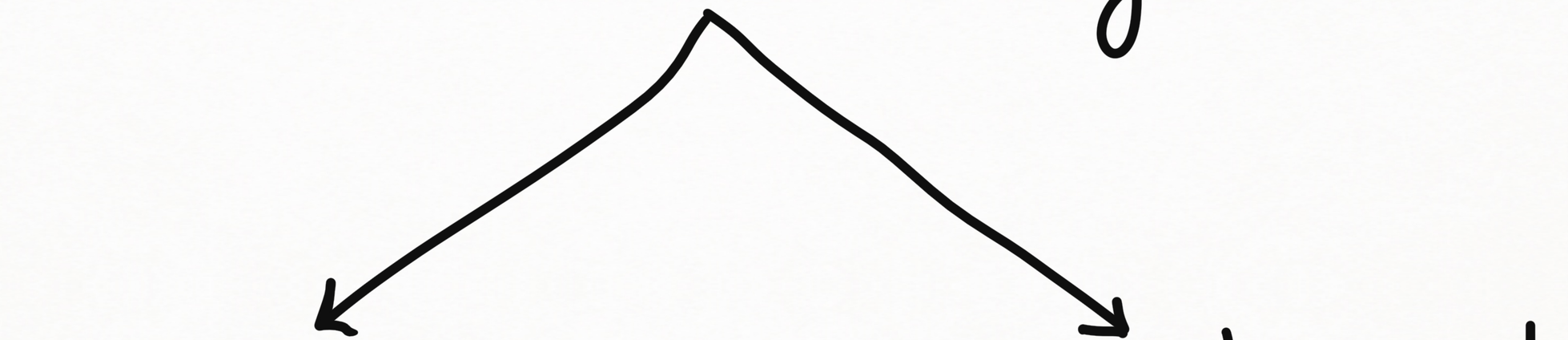
Framework

→ A set of tools to effectively and efficiently do a complex task with minimum effort.

Example :- i) Creating a set of objects using collection framework

ii) Creating a Java web application using spring framework.

Framework Learning

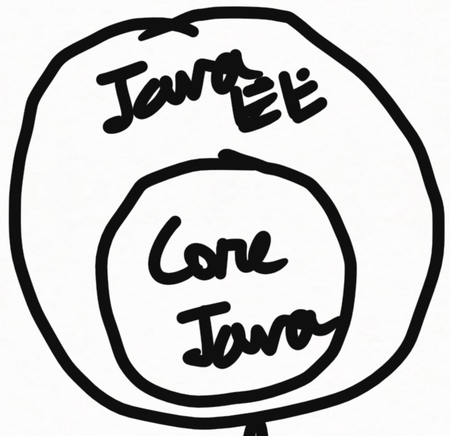


How to use the tools in the framework

How the framework works internally.

Spring Framework

→ Based Java EE (Java Enterprise Edition)



→ Modules, hence loose coupling

Inversion of Control (IoC)

→ Transfers the control of objects to the container/framework.

→ Works with object-oriented programming

→ One of the ways IoC can be achieved is dependency injection (DI).

Advantages are:-

i) Loose coupling
ii) Greater modularity
iii) easier to switch implementation
iv) easier to test

Dependency Injection (DI)

→ Implementation of IoC.

→ Allows for loose coupling

→ Moves the responsibility of managing components onto the container.

→ Fundamental aspect of Spring framework.

→ The framework will inject objects into other objects.

```
class Item{ }
```

Traditional Resolution

```
class Store{
    private Item item;
    public Store(){
        item= new Item();
    }
}
```

Dependency Injection

```
class Store{
    private Item item;
    public Store(Item item){
        this.item = item;
    }
}
```

Spring IoC container – Manages DI

Configurations Metadata

POJO = Plain Old Java Object.

↳ In Spring framework, "ApplicationContent" represents the Spring IoC container.

↳ ApplicationContent manages all the beans by using the configuration metadata.

↳ ApplicationContent is an interface.

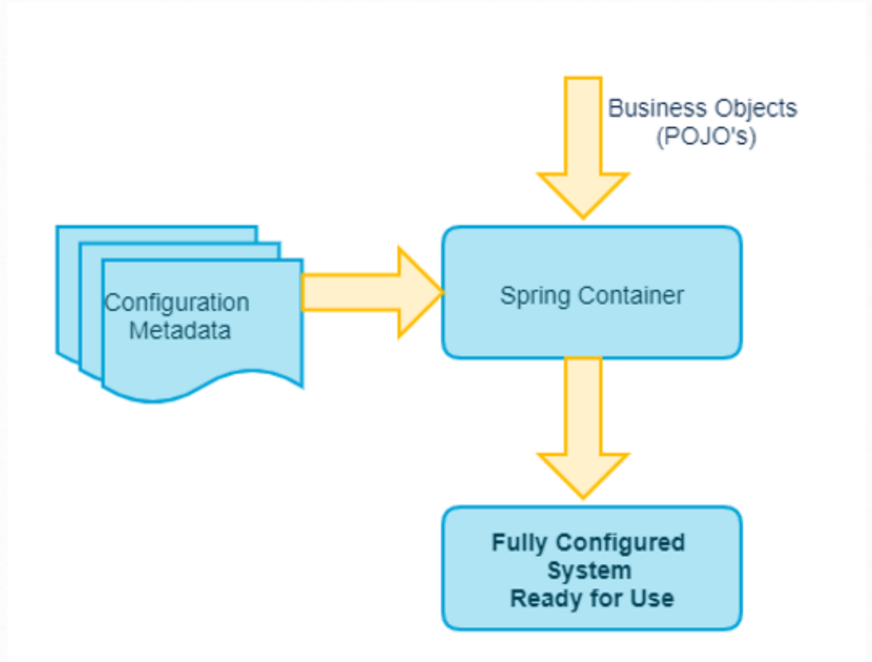
↳ Spring framework provides several implementations of the ApplicationContent interface.

↳ Some of the implementations are:-

i) ClassPathXMLApplicationContext

ii) AnnotationConfigApplicationContext

etc.



Types of DI

i) Constructor Injection ✓

ii) Property (Field) Injection ✗

iii) Setter (Method Injection) ✓ (default)

✓ = supported by Spring