SENG 301 - Software Analysis and Design

Lab 10 Façade Design Pattern

May Mahmoud

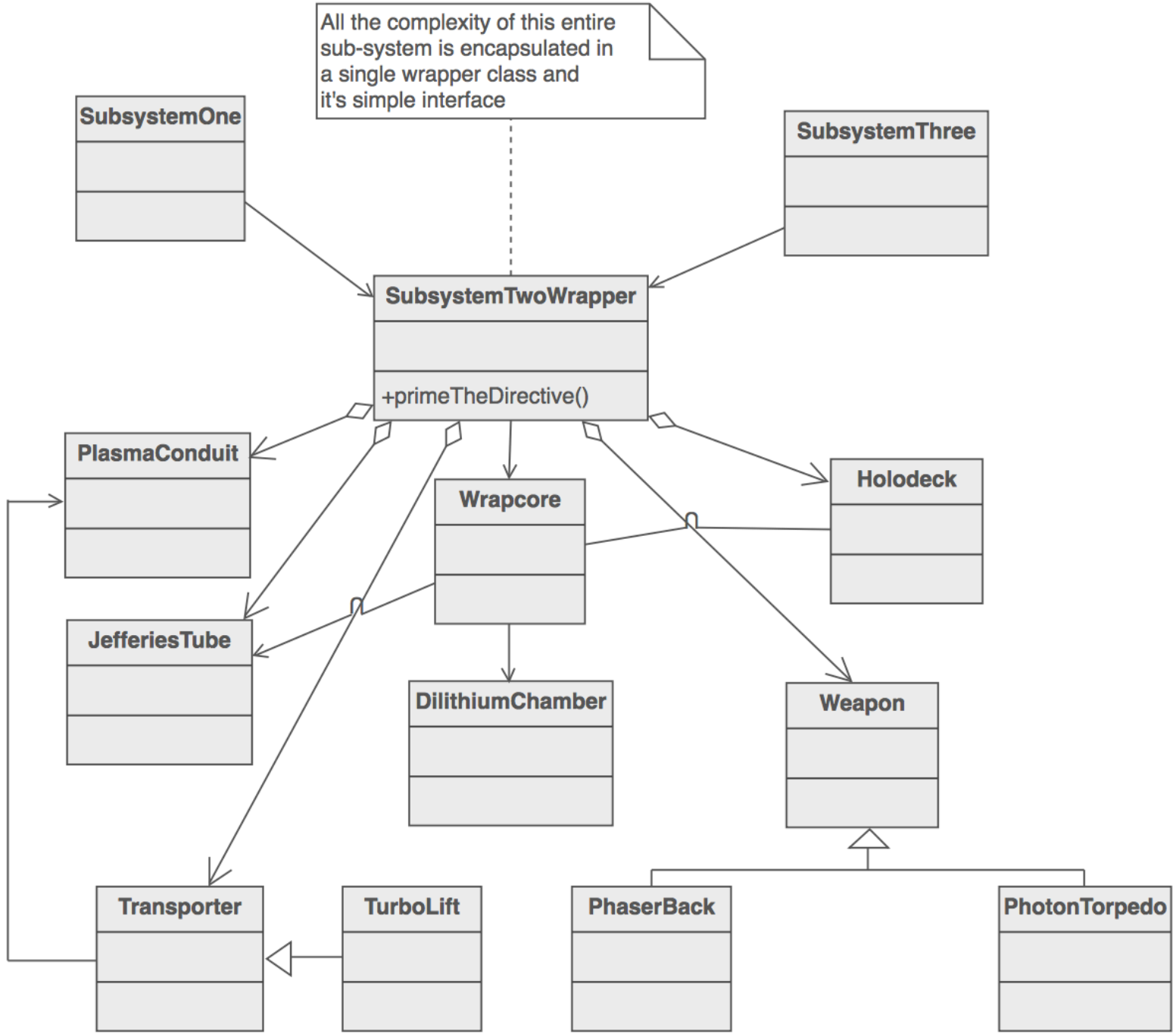
Design Patterns

- A software design pattern is a general reusable solution to a commonly occurring problem within a given context in software design

- It is a description or template for how to solve a problem that can be used in many different situations

Façade Design Pattern

- Provide a unified interface to a set of interfaces in a subsystem. Facade defines a higher-level interface that makes the subsystem easier to use.
- Wrap a complicated subsystem with a simpler interface.


All the complexity of this entire sub-system is encapsulated in a single wrapper class and it's simple interface
SubsystemOne
SubsystemThree
SubsystemTwoWrapper
+primeTheDirective()
PlasmaConduit
Wrapcore
Holodeck
JefferiesTube
DilithiumChamber
Weapon
Transporter
TurboLift
PhaserBack
PhotonTorpedo

```
/* Complex parts */

class CPU {
    public void freeze() { ... }
    public void jump(long position) { ... }
    public void execute() { ... }
}

class Memory {
    public void load(long position, byte[] data) { ... }
}

class HardDrive {
    public byte[] read(long lba, int size) { ... }
}

/* Facade */

class ComputerFacade {
    private CPU processor;
    private Memory ram;
    private HardDrive hd;

    public ComputerFacade() {
        this.processor = new CPU();
        this.ram = new Memory();
        this.hd = new HardDrive();
    }

    public void start() {
        processor.freeze();
        ram.load(BOOT_ADDRESS, hd.read(BOOT_SECTOR, SECTOR_SIZE));
        processor.jump(BOOT_ADDRESS);
        processor.execute();
    }
}
```

```
/* Client */

class You {
    public static void main(String[] args) {
        ComputerFacade computer = new ComputerFacade();
        computer.start();
    }
}
```

START LAB 10