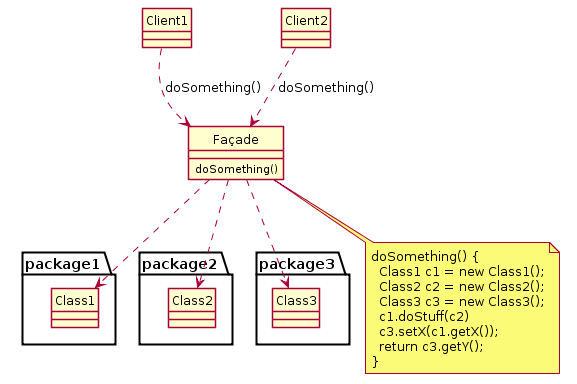
Façade Design Pattern

GOF : **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.**

# **Class Diagram**



# **Java Code**

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();

}

}

# **Another Example**

public class Façade {

    public void PerformAction() {

        Class1A c1a = new Class1A();

        Class1B c1b = new Class1B();

        Class2A c2a = new Class2A();

        Class2B c2b = new Class2B();

        int result1a = c1a.Method1A();

        int result1b = c1b.Method1B(result1a);

        int result2a = c2a.Method2A(result1a);

        c2b.Method2B(result1b, result2a);

    }

}

public class Class1A {

    public int Method1A() {

        // Sample code

    }

}

public class Class1B {

    public int Method1B(int param) {

        // Sample code

    }

}

public class Class2A {

    public int Method2A(int param) {

        // Sample code

    }

}

public class Class2B {

    public void Method2B(int param1, int param2) {

        // Sample code

    }

}