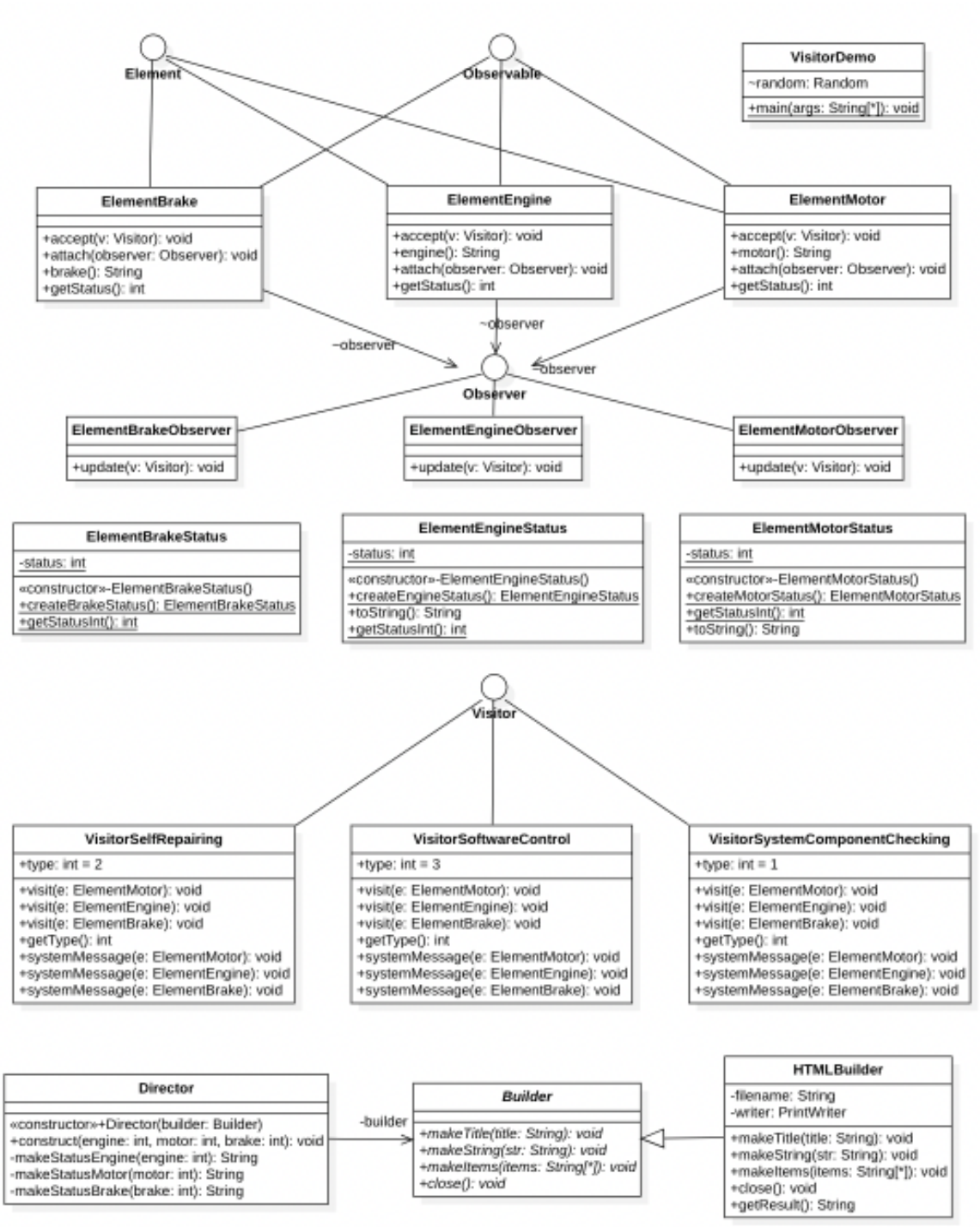
⚡

OODP PHEV Simulation

21700663 정예찬

2022, OODP The 3rd Draft

Class Diagram



Design Pattern 적용

Visitor Pattern

```
//ElementEngine.java
public class ElementEngine implements Element{
```

```
    @Override
    public void accept(Visitor v) {
        ElementEngineStatus.createEngineStatus();
        v.visit(this);
        this.observer.update(v);
    }

  ...

}
```

visitor를 사용하기 위한 Element와 accept함수입니다.

```
//Visitor.java

interface Visitor {
  public void visit(ElementEngine e);
  public void visit(ElementMotor e);
  public void visit(ElementBrake e);
}
```

Visitor Pattern을 적용하기 위한 Interface입니다.

visit 메서드를 오버로딩하여 다른 클래스들을 같은 문법으로 사용 가능합니다.

```
//VisitorSystemComonentChecking.java 일부

  @Override
  public void visit(ElementMotor e) {
    System.out.println("=M System/Component Checking on " + e.motor());
    systemMessage(e);
  }

  @Override
  public void visit(ElementEngine e) {
    System.out.println("=E System/Component Checking on " + e.engine());
    systemMessage(e);
  }

  @Override
  public void visit(ElementBrake e) {
    System.out.println("=B System/Component Checking on " + e.brake());
    systemMessage(e);
  }
  public void systemMessage(ElementMotor e){
    switch(e.getStatus())//.. Element의 Status에 따른 문장 출력
  }
  public void systemMessage(ElementEngine e){
    ...
  }
  public void systemMessage(ElementBrake e){
```

```
    ...
  }
```

Observer Pattern

```
//Observer.java
public interface Observer {
    public void update(Visitor v);
}

//Observable.java
public interface Observable {
    public void attach(Observer observer);
}
```

Observer Pattern을 사용하기 위한 Observer와 Observable 인터페이스입니다.

```
//ElementEngine.java
public class ElementEngine implements Observable{
  Observer observer;

    @Override
    public void accept(Visitor v) {
        ...
        this.observer.update(v);
    }

    @Override
    public void attach(Observer observer) {
        this.observer = observer;
    }

    ...

}
```

Element를 Observable로 implement하고, visitor가 accept될때마다 Observer를 업데이트 하였습니다.(NotifyObservers 역할)

Singleton Pattern

```
//ElementEngineStatus.java

public class ElementEngineStatus {
    private static ElementEngineStatus theObject;
    private static int status;
```

```
    //private construct를 사용하였음
    private ElementEngineStatus(){
        ...
        ElementEngineStatus.status = num;
    }

    //public create method로 singleton pattern 적용
    public static ElementEngineStatus createEngineStatus(){
        if (theObject == null)
            theObject = new ElementEngineStatus();
        return theObject;
    }

    ...

}
```

각각의 Element에 대하여 Element~~Status 파일을 생성하여 Singleton 역할을 하게 했음.

Builder Pattern

```
//Builder.java

public abstract class Builder {
    public abstract void makeTitle(String title);
    public abstract void makeString(String str);
    public abstract void makeItems(String[] items);
    public abstract void close();
}
```

```
//Director.java 일부
public void construct(int engine, int motor, int brake) {
        builder.makeTitle("PHEV_Checkingus_Log");
        builder.makeString("Gasoline Engine Being Checked");
        String[] engineChecker = {"Checking Engine",makeStatusEngine(engine)};
        builder.makeItems(engineChecker);
        builder.makeString("Checked Electric Motor");
        String[] motorChecker = {"Motor Checked",makeStatusMotor(motor)};
        builder.makeItems(motorChecker);
        builder.makeString("Checking Regenerative Brake System");
        String[] brakeChecker = {"Regenerative Brake Checked",makeStatusBrake(brake)};
        builder.makeItems(brakeChecker);
        builder.close();
}

private String makeStatusEngine(int engine) {
        switch (engine) {
            ...//엔진의 상태에 따라 다른 String 반환!
        }
}
```

위의 방식으로 Builder Pattern을 사용하여 HTML 파일 생성합니다.

Output

output 1

```
jeong-yechan@jeong-yechan-ui-MacBookAir PHEV % javac *.java
jeong-yechan@jeong-yechan-ui-MacBookAir PHEV % java VisitorDemo

============ PHEV SIMULATOR START ============

=E System/Component Checking on Gasoleine Engine
Engine Oil To Be Checked on Gasoline Engine
Gasoline Engine Log/Observer: Suggesting Oil Change

=M System/Component Checking on Electric Motor
Irregular Motor Power with Electric Motor
Electric Motor Log/Observer: Suggesting Motor Brush Contact

=B System/Component Checking on Brake System
No Electricity Generated with Regenerative Brake System
Regnerative Brake System Log/Observer: Suggesting Brake Generator Contact Checking

PHEV_Checkingus_Log.htmlis made.

=E Self Repairing on Gasoleine Engine
Exchange oil for Gasoline Engine
Gasoline Engine Log/Observer: Repairing Engine Oil Warning

=M Self Repairing on Electric Motor
Apply oil to motor brush contact of Electric Motor
Electric Motor Log/Observer: Repairing Irregular Motor Power Warning

=B Self Repairing on Brake System
Apply Oil to Brake Generator Contact Point of Regenerative Brake System
Regnerative Brake System Log/Observer: Repairing No Electricity Generated for Regnerative Brake System

=E Software Control and Recovery for Gasoleine Engine
Increase the control software level of oil supply forGasoline Engine
Gasoline Engine Log/Observer: Resetting Oil Level for engine oil warning

=M SSoftware Control and Recovery for Electric Motor
Generating a Motor Power Monitoring Module from the Abstract Factory
Electric Motor Log/Observer: Software Resetting with Motor Power Monitoring Module for Irregular Motor Power Warning

=B Software Control and Recovery for Brake System
Initiate the monitoring SW for No Electricity Problem from Resen. Brake System
Regnerative Brake System Log/Observer: Resetting Electricity Monitoring SW for Regenerative Brake System

 Engine Status: Oil
 Motor Status: Irregular Motor Power
 Regernative Brake Status: No Electric
Engine Running with Slow-Down
jeong-yechan@jeong-yechan-ui-MacBookAir PHEV %
```

output 2

```
jeong-yechan@jeong-yechan-ui-MacBookAir PHEV % java VisitorDemo

============ PHEV SIMULATOR START ============

=E System/Component Checking on Gasoleine Engine
Irregular Engine Power with Gasoline Engine
Gasoline Engine Log/Observer: Suggesting Spark Plug Checking

=M System/Component Checking on Electric Motor
Normal Motor Power
Electric Motor Log/Observer: Motor Running OK

=B System/Component Checking on Brake System
Normal Regen. Brake Working with Regenerative Brake System
Regnerative Brake System Log/Observer: Regen. Brake Working OK

PHEV_Checkingus_Log.htmlis made.

=E Self Repairing on Gasoleine Engine
Self clean the spark plug of Gasoline Engine
Gasoline Engine Log/Observer: Repairing Irregular Engine Power Warning

=M Self Repairing on Electric Motor
Electric Motor Log/Observer: Motor Status OK

=B Self Repairing on Brake System
Regnerative Brake System Log/Observer: Regnerative Brake System OK

=E Software Control and Recovery for Gasoleine Engine
Software Resetting for Gasoline Engine
Gasoline Engine Log/Observer: Software Resetting with Power Monitoring Module for Irregular Engine Power Warning

=M SSoftware Control and Recovery for Electric Motor
Electric Motor Log/Observer: Keep Curren SW Monitoring Module for Motor

=B Software Control and Recovery for Brake System
Regnerative Brake System Log/Observer: Keep Curren SW Monitoring Module for Regenerative Brake System

 Engine Status: Irregular Power
 Motor Status: OK
 Regernative Brake Status: OK
Run with Electric Mode
jeong-yechan@jeong-yechan-ui-MacBookAir PHEV %
```

output 3

```
jeong-yechan@jeong-yechan-ui-MacBookAir PHEV % java VisitorDemo

============ PHEV SIMULATOR START ============

=E System/Component Checking on Gasoleine Engine
Gasoline Engine Log/Observer: Engine Running OK

=M System/Component Checking on Electric Motor
Car Vibration with Electric Motor
Electric Motor Log/Observer: Suggesting Motor Mounting Bolt Checking

=B System/Component Checking on Brake System
Weak Generative Electricity with Regenerative Brake System
Regnerative Brake System Log/Observer: Suggesting Generator Coil Damage Checking

PHEV_Checkingus_Log.htmlis made.

=E Self Repairing on Gasoleine Engine
Gasoline Engine Log/Observer: Engine Status Good

=M Self Repairing on Electric Motor
Turn Tightly Motor Mounting Bolt of Electric Motor
Electric Motor Log/Observer: Repairing Car Vibration with Motor

=B Self Repairing on Brake System
Use an extra generator system for Regenerative Brake System
Regnerative Brake System Log/Observer: Repairing Generator Coil Damage for Regnerative Brake System

=E Software Control and Recovery for Gasoleine Engine
Gasoline Engine Log/Observer: Keep Curren SW Monitoring Module for Engine

=M SSoftware Control and Recovery for Electric Motor
Initiate Monitoring SW for Motor Vibration of Electric Motor
Electric Motor Log/Observer: Resetting the Monitoring SW for Car Vibration with Motor Action

=B Software Control and Recovery for Brake System
Initiate the monitoring SW for Weak Electricity Problem of Regenerative Brake System
Regnerative Brake System Log/Observer: Resetting Coil Damage Monitoring SW for Regenerative Brake System

 Engine Status: OK
 Motor Status: Vibrating Motor
 Regernative Brake Status: Weak Electric
Run with Engine Mode
jeong-yechan@jeong-yechan-ui-MacBookAir PHEV %
```

HTML output

PHEV_Checkingus_Log

Gasoline Engine Being Checked

- Checking Engine
- Engine OK!

Checked Electric Motor

- Motor Checked
- Suggesting Motor Mounting Bolt Checking

Checking Regenerative Brake System

- Regenerative Brake Checked
- Suggesting Generator Coil Damage Checking

Developer Note

- Element들의 상태와 Visitor의 구분자들을 관리의 용이를 위해 String이 아닌 Integer로 구분하였다.