Universidade Federal de Campina Grande
Professores: Melina Mongiovi e Fábio
Aluno: Eduardo Afonso Nunes da Silva Matrícula: 118210610

PSOFT LAB 6

Eu utilizei o move method, pois o método de checagem da classe Gerente usava muitos dados da classe Produto, então coloquei o método para lá, assim, gerente delega o aviso (contem os clientes) e a checagem do projeto. e então, eu extraí parte do método, para um novo método com o nome de avisaAtraso, assim, diminuindo a quantidade de código em um método e não colocando muitas responsabilidades em um método só.

Bad Smells identificados: Feature Envy e Método longo.

Código anterior da classe Gerente:

```
package badcode;

import java.util.List;

public class Gerente {
        public String checar( Projeto p ) {
                // verifica prazo do projeto
                if (p.d < 90) {
                // verifica se projeto ainda está em andamento
                        if (!p.isEntregue()) {
                        // projeto ainda em andamento e com prazo curto para entrega
                                return "Projeto está apertado" ;
                        } else {
                                return "Projeto entregue";}
                } else {
                        List<Cliente> clients = p.getClientes();
                        for (Cliente c : clients) {
                                c.avisaAtraso(p.d);
                        }
                        return "Projeto atrasado";
                }
        }
}
```

Código atual da classe Gerente:

```
import java.util.List;

public class Gerente {
    public String checar( Projeto p ) {
        return p.checar();
    }
}
```

Modificações: gerente passa a delegar o projeto a sua checagem e aviso aos clientes (move method).
Bad Smell: Feature Envy.

Código anterior da classe Projeto:

```
package badcode;

import java.util.List;

public class Projeto {
        public int d;
        public boolean isEntregue() {
                return false;
        }
        public List<Cliente> getClientes() {
                return null;
        }
}
```

Código atual:

```java
import java.util.List;

public class Projeto {
    public int d;
    public boolean isEntregue() {
        return false;
    }
    public List<Cliente> getClientes() {
        return null;
    }

    public String checar(){
        if (d < 90) {
            // verifica se projeto ainda está em andamento
            if (!isEntregue()) {
                // projeto ainda em andamento e com prazo curto para entrega
                return "Projeto está apertado" ;
            } else {
                return "Projeto entregue";}
        }
        else {
            avisaAtraso();
            return "Projeto atrasado";
        }
    }

    private void avisaAtraso() {
        List<Cliente> clients = getClientes();
        for (Cliente c : clients) {
            c.avisaAtraso(d);
        }
    }
}
```

Modificações: agora o próprio projeto chega e retorna seu estado, por ser o expert dessa informação (prazo e se está entregue), e o método foi quebrado (extract method) para que diminua o código do método checar() e não tenha muitas responsabilidades.

Bad smell: método longo.

Justificativa das mudanças: se o Projeto possui os clientes, ele pode manipular a lista de clientes e informá-los do atraso, e se ele possui informações de prazo e entrega, ele também pode dizer aos objetos da classes que o utilizam se está atrasado ou não.