

Started on	Monday, 12 December 2022, 6:06 PM
State	Finished
Completed on	Monday, 12 December 2022, 6:15 PM
Time taken	9 mins 1 sec
Marks	4,00/5,00
Grade	16,00 out of 20,00 (80%)

Question 1

Correct

Mark 1,00 out of 1,00

Qual dos seguintes métodos **não se pode considerar um método de criação?**

```
public static Product f1(String name, double cost){  
    return new Product(name, cost);  
}  
  
public Product f2(){  
    return new Product(name, cost);  
}  
  
public static Product f3(boolean type, String name, double cost){  
    if(type)  
        return new Product(name, 0);  
    else  
        return new Product(name, cost);  
}  
  
public Product f4(double cost){  
    this.cost = cost;  
    return this;  
}
```

Select one:

- ☐ a. f2
- ☒ b. f4 ✓
- ☐ c. f1
- ☐ d. f3

Question 2

Correct

Mark 1,00 out of 1,00

Considere o seguinte excerto de código abaixo, que se refere à utilização de um padrão de software. Qual é o padrão que está ilustrado?

```
public class FactoryMethodMain {
    public static void main(String[] args) {
        Loja loja = new LojaAmerica();
        System.out.println(loja.listMenu());
        Pizza p1 = loja.make("type1", "pp1");
        Pizza p2 = loja.make("type2", "pp2");
        p1.applyPromotion(5);
        p2.applyPromotion(5);
        System.out.println(p1);
        System.out.println(p2);
    }
}

public interface Loja {
    Pizza make(String type, String name);
    void listMenu();
}

public class LojaAmerica implements Loja{
    public Pizza make(String type, String name) {
        switch (type){
            case "type1": return new PizzaNapoles(name);
            case "type2": return new PizzaTuna(name);
            default: throw new IllegalArgumentException();
        }
    }

    public static String listMenu(){
        return "Pizza Napoles ; PizzaTuna";
    }
}
```

Select one:

- ☒ a. Factory Method; ✓
- ☐ b. Abstract Factory
- ☐ c. Simple Factory
- ☐ d. Nenhum dos Anteriores

Question 3

Correct

Mark 1,00 out of 1,00

Considere o seguinte excerto de código que se refere ao padrão Memento, onde a classe ShoppingCart assume o papel de Concrete Originator. A classe ShoppingCart simula um carrinho de compras onde se adicionam produtos e se pode obter o seu preço total.

```
public class MainConsola {  
    public static void main(String args[]){  
        ShoppingCart cart = new ShoppingCart();  
        Caretaker caretaker= new Caretaker(cart);  
        caretaker.saveState();  
        cart.addProduct(new Product("bananas",2));  
        cart.addProduct(new Product("peras",2));  
        cart.addProduct(new Product("maças",5));  
        caretaker.restoreState();  
        System.out.println(cart.getTotal()); // imprime o valor total dos produtos no carrinho  
    }  
}
```

Qual o output do programa

Select one:

- ☒ a. 0 ✓
- ☐ b. 2
- ☐ c. Nenhum dos Anteriores
- ☐ d. 9

Question 4

Incorrect

Mark 0,00 out of 1,00

Considere o padrão Memento e o código da classe **Y** que implementa a interface *Originator*.

```
public class Y implements Originator {
    private final List<String> at1;
    private int x;

    public Y() {
        at1 = new LinkedList<>();
        x=0;
    }

    // other methods

    private class MyMemento implements Memento {
        private final List<String> at1;

        // code missing
    }
}

//Snippet A
public MyMemento(List<String> at1, X x) {
    this.at1 = new LinkedList<>(at1);
    this.x = x;
}

//Snippet B
public MyMemento(Y stateToSave) {
    this.at1 = new LinkedList<>(stateToSave.at1);
}

//Snippet C
public MyMemento(List<String> stateAt1ToSave) {
    this.at1 = stateAt1ToSave;
}
```

Relativamente à inner classe **MyMemento**, selecione a opção de código que preenche corretamente o código em falta:

Select one:

- ☒ a. Snippet C ❌
- ☐ b. Snippet A
- ☐ c. Snippet B
- ☐ d. Todas estão incorretas

Question 5

Correct

Mark 1,00 out of 1,00

No padrão MVC o padrão Observer é usado para:

Select one:

- ☒ a. Implementar o mecanismo de notificação entre o participante Model e o participante View ✔️
- ☐ b. Nenhuma das anteriores
- ☐ c. Implementar o mecanismo de notificação entre o participante Model e o participante Controller
- ☐ d. Implementar o mecanismo de notificação entre o participante Controller e o participante View