

Primeira prova

Aluno: Heitor Freitas Ferreira

Questão 1

A

Falsa

- **Justificativa:** O conceito de encapsulamento é alcançado por meio da definição da visibilidade **privada** aos atributos e **pública** aos métodos.

B

Verdadeira

C

Falsa

- Justificativa: O conceito de herança seria algo como uma classe macaco derivar de uma classe mamífero, o caso apresentado (motor->automóvel) é um exemplo de composição, uma associação parte/todo, pois caso destruímos o objeto carro, o motor também deverá ser destruído

D

Falsa

- Justificativa: Para uma classe ser considerada abstrata, **pelo menos um de** seus métodos devem ser abstratos. Em Java, para se definir uma classe abstrata deve-se utilizar a palavra chave "abstract" no início de sua declaração, algo do tipo `public abstract class ContaBancaria`.

Questão 2

A saída será a soma dos retornos das diferentes implementações do método `Calcula`, quando esse método é executado em um objeto `Questao` o retorno é 1, como no caso da instância `Objs[0]`, quando é executada em um objeto `Q2` o retorno é 2, como no caso da instância `Objs[1]`, quando é executada em um objeto `Q3` o retorno é 3, como no caso da instância `Objs[2]`.

Logo o somatório desses 3 retornos é $1+2+3 = 6$.

Isso será possível pela *sobreposição* de métodos, Q2 e Q3 são classes herdeiras da classe `Questao`, como todas possuem métodos iguais (com o mesmo nome, aridade, tipo de parâmetros e tipo de retorno), os métodos descritos nas classes filhos tem prioridade na execução.

Por exemplo, uma classe macaco que herde de uma classe mamífero, ambas possuem o método `andar()`, quando esse método é instanciado como macaco a implementação de `andar()` da classe macaco é executada, e não a implementação da classe mamífero.

Questão 3

A

```
public class Produto {  
    private String nomeloja;  
    private float preco;  
  
    public String getNomeLoja() {  
        return nomeloja;  
    }  
  
    public void setNomeLoja(String nomeLoja) {  
        this.nomeloja = nomeLoja;  
    }  
  
    public float getPreco() {  
        return preco;  
    }  
  
    public void setPreco(float preco) {  
        this.preco = preco;  
    }  
}
```

B

```
public class Produto {  
    private String descricao;  
    public String getDescricao() {  
        return descricao;  
    }  
  
    private String nomeloja;  
    public String getNomeLoja() {  
        return nomeloja;  
    }  
    public void setNomeLoja(String nomeLoja) {  
        this.nomeloja = nomeLoja;  
    }  
  
    private float preco;  
    public float getPreco() {  
        return preco;  
    }  
    public void setPreco(float preco) {  
        this.preco = preco;  
    }  
}
```

C

```
public class Produto {
    public Produto(String descricao, float preco) {
        this.descricao = descricao;
        setPreco(preco);
    }

    private String descricao;
    public String getDescricao() {
        return descricao;
    }

    private String nomeloja;
    public String getNomeloja() {
        return nomeloja;
    }
    public void setNomeloja(String nomeloja) {
        this.nomeloja = nomeloja;
    }

    private float preco;
    public float getPreco() {
        return preco;
    }
    public void setPreco(float preco) {
        this.preco = preco;
    }
}
```

D

```
public class Mouse extends Produto{
    private String tipo;

}
```

E

```
public class Mouse extends Produto{
    private String tipo;

    public Mouse(String descricao, float preco, String tipo) {
        super(descricao, preco);
        this.tipo = tipo;
    }
}
```

```
    }  
}
```

F

```
public class Livro extends Produto{  
    private String editora;  
}
```

G

```
public class Livro extends Produto{  
    private String editora;  
  
    public Livro(String descricao, float preco, String editora) {  
        super(descricao, preco);  
        this.editora = editora;  
    }  
}
```

H

```
public class Mouse extends Produto{  
    private String tipo;  
  
    public Mouse(String descricao, float preco, String tipo) {  
        super(descricao, preco);  
        this.tipo = tipo;  
    }  
    public String getDescricao() {  
        return super.getDescricao() + "\n" + this.tipo ;  
    }  
}  
  
public class Livro extends Produto{  
    private String editora;  
  
    public Livro(String descricao, float preco, String editora) {  
        super(descricao, preco);  
        this.editora = editora;  
    }  
  
    public String getDescricao() {  
        return super.getDescricao() + "\n" + this.editora ;  
    }  
}
```

I

```
import java.util.List;

public class Livro extends Produto{
    private String editora;

    private List autores;
    public List getAutores() {
        return autores;
    }
    public void setAutor(List autores) {
        autores = autores;
    }

    public Livro(String descricao, float preco, String editora) {
        super(descricao, preco);
        this.editora = editora;
    }

    public String getDescricao() {
        return super.getDescricao() + "\n" + this.editora ;
    }
}

public class Autor {
    private String nome;
    private String CPF;

    private List livros;
    public List getLivros() {
        return livros;
    }
    public void setLivros(List livros) {
        this.livros = livros;
    }

    public Autor(String nome, String CPF) {
        this.nome = nome;
        this.CPF = CPF;
    }

    public String getNome() {
        return nome;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }

    public String getCPF() {
        return CPF;
    }
}
```

```
    }

    public void setCPF(String CPF) {
        this.CPF = CPF;
    }
}
```

J

Sim, tanto é possível usar o recurso do polimorfismo quanto foi usado, mais especificamente ao sobrescrever o método `getDescricao()`, que foi implementado na classe mãe `Produto` e nas suas classes filhas `Livro` e `Mouse`.

Implementação completa

Classe Produto

```
public class Produto {
    public Produto(String descricao, float preco) {
        this.descricao = descricao;
        setPreco(preco);
    }

    private String descricao;
    public String getDescricao() {
        return descricao;
    }

    private String nomeloja;
    public String getNomeloja() {
        return nomeloja;
    }
    public void setNomeloja(String nomeloja) {
        this.nomeloja = nomeloja;
    }

    private float preco;
    public float getPreco() {
        return preco;
    }
    public void setPreco(float preco) {
        this.preco = preco;
    }
}
```

Classe Mouse

```
public class Mouse extends Produto{
    private String tipo;

    public Mouse(String descricao, float preco, String tipo) {
        super(descricao, preco);
        this.tipo = tipo;
    }
    public String getDescricao() {
        return super.getDescricao() + "\n" + this.tipo ;
    }
}
```

Classe Livro

```
import java.util.List;

public class Livro extends Produto{
    private String editora;

    private List autores;
    public List getAutores() {
        return autores;
    }
    public void setAutor(List autores) {
        autores = autores;
    }

    public Livro(String descricao, float preco, String editora) {
        super(descricao, preco);
        this.editora = editora;
    }

    public String getDescricao() {
        return super.getDescricao() + "\n" + this.editora ;
    }
}
```

Classe Autor

```
import java.util.List;

public class Autor {
    private String nome;
    private String CPF;

    private List livros;
    public List getLivros() {
        return livros;
    }
}
```

```
    public void setLivros(List livros) {
        this.livros = livros;
    }

    public Autor(String nome, String CPF) {
        this.nome = nome;
        this.CPF = CPF;
    }

    public String getNome() {
        return nome;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }

    public String getCPF() {
        return CPF;
    }

    public void setCPF(String CPF) {
        this.CPF = CPF;
    }
}
```