

1. Observe o código abaixo e responda as questões a seguir.

```

abstract class ItemBiblioteca {
    protected String titulo;
    protected String autor;
    protected boolean emprestado;

    public ItemBiblioteca(String titulo, String autor) {
        this.titulo = titulo;
        this.autor = autor;
        this.emprestado = false;
    }
    public abstract String getDescricao();
    public boolean isEmprestado() {
        return emprestado;
    }
}

interface Emprestavel {
    public void emprestar();
    public void devolver();
}

class Revista extends ItemBiblioteca {
    private String editora;

    public Revista(String titulo, String autor, String editora) {
        super(titulo, autor);
        this.editora = editora;
    }

    @Override
    public String getDescricao() {
        return "Revista: " + titulo + " por " + autor + " editora " + editora;
    }
}

class Livro extends ItemBiblioteca {
    private int edicao;

    public Livro(String titulo, String autor, int edicao) {
        super(titulo, autor);
        this.edicao = edicao;
    }

    @Override
    public String getDescricao() {
        return "Livro: " + titulo + " por " + autor + " por " + edicao;
    }
}

class ItemAdapter implements Emprestavel {
    private ItemBiblioteca item;

    public ItemAdapter(ItemBiblioteca item) {
        this.item = item;
    }

    @Override
    public void emprestar() {
        if(item.isEmprestado()) System.out.println("O item já encontra-se emprestado.");
        else {
            item.emprestado = true;
            System.out.println("Empréstimo realizado com sucesso.");
        }
    }

    @Override
    public void devolver() {
        if(!item.isEmprestado()) System.out.println("O item não encontrava-se emprestado.");
        else {
            item.emprestado = false;
            System.out.println("Devolução realizado com sucesso.");
        }
    }
}

```

- 10 a. Explique o conceito de herança na orientação a objetos. Quais são as vantagens e desvantagens de usar herança? Identifique onde está sendo utilizado o conceito de herança no exemplo dado.
- 10 b. O que é polimorfismo em orientação a objetos? Onde o polimorfismo está sendo utilizado no exemplo acima? Como o polimorfismo contribui para a flexibilidade do código?
- 10 c. Descreva como os conceitos de interface e classe abstrata estão sendo utilizados no exemplo. Quando você usaria uma interface em vez de uma classe abstrata e vice-versa?
- 10 d. Qual a finalidade de uma classe adaptadora? Explique a função da classe adaptadora no exemplo da questão.

- 10 e. Analise o código a seguir em conjunto com o exemplo anterior, aponte as linhas que contêm erros e comente qual o problema encontrado.

```
1. public class Biblioteca {  
2.     public static void main(String[] args) {  
3.         Livro livro = new Livro("O Senhor dos Anéis", "Tolkien", 3);  
4.         livro.emprestar();  
5.  
6.         Revista revista = new Revista("National Geographic", "Vários", "The Walt Disney Company");  
7.         System.out.println(revista.getDescricao());  
8.  
9.         ItemAdapter livroAdaptado = new ItemAdapter(new Livro("O Senhor dos Anéis", "Tolkien", 3));  
10.        livroAdaptado.emprestar();  
11.  
12.        ItemAdapter revistaAdaptada = new ItemAdapter(revista);  
13.        System.out.println(revistaAdaptada.getDescricao());  
14.  
15.        revistaAdaptada.devolver();  
16.    }  
17. }
```

- a) Exemplo na orientação a objetos é o relacionamento entre uma classe "mãe" e outra "filha". No exemplo a classe filha herda todos atributos, métodos, e métodos da classe mãe.
- Dentre os vantagens da herança, temos a reutilização do código, bem como a capacidade de herdar os métodos e facilitar a manutenção. C
- No exemplo, a carreta de exemplo é uma classe que herda os métodos de uma classe "Alvista" e "Lixa", que são subclasses de "itembiblioteca".
- b) Polimorfismo é a capacidade de um objeto em reagir de diferentes maneiras, e é uma classe que herda o método de código scena, onde o carro recebe um objeto "Itembiblioteca" (ou seus descendentes) e vai incorporar essas características e métodos dessa forma. C
- O polimorfismo contribui para a flexibilidade do código na medida que permite a atribuição de um objeto de uma classe-filha ou de uma classe-mãe, permitindo maior liberdade de código.
- c) No exemplo, Emprestavel é uma interface implementada por uma classe adaptadora; e Itembiblioteca é uma classe abstrata, implementada por suas filhas. Uma classe abstrata tem melhor desempenho que uma classe concreta, pois pode ser herada com parâmetros e métodos remanescentes, evitando repetição de código.
- Uma interface é mais fácil de ser implementada que uma classe abstrata (não é possível), mas pode ter diferentes implementações, dependendo da classe que os implementa.
- d) Uma classe adaptadora serve para conectar interface com diferentes implementações, isto é, diferentes tipos de interfaces/implementações não necessariamente da mesma classe. No exemplo, a classe adaptadora "Item adaptado" implementa a interface Emprestavel e conecta com a classe "Item biblioteca", tornando possível que os métodos interajam com os objetos da biblioteca adaptadamente.

v) Na linha 4, o método `Impressor::pintar()` corre direta no `Itembiblioteca`, mas tem a
correr `Itemadotar`, terendo o comando item.lns.

Nesta 10, o método `getImprimir()`; ~~pintar~~ corre revista, logo, pode ser impressa
uma lista polimórfica de objetos revista.

Na linha 11, o item não foi imprimido, ~~porque~~ a função de impressão não tem nenhuma.