

Programação Orientada a Objetos**Profa. Mirela Teixeira Cazzolato****Monitor: Erik Oda Coulter****Lista de exercícios 5: Aula 07 – Polimorfismo, Classes Abstratas, Interfaces e Exceções**

1. As seguintes afirmações são verdadeiras ou falsas? Justifique as alternativas falsas.

a) Todos os métodos em uma classe abstrata devem ser declarados como métodos abstratos.

() Justificativa: _____.

b) Se uma superclasse declara um método abstrato, uma subclasse deve implementar aquele método.

() Justificativa: _____.

c) Invocar um método exclusivo da subclasse por uma variável do tipo da subclasse não é permitido.

() Justificativa: _____.

d) Um objeto de uma classe que implementa uma interface pode ser considerado como um objeto daquele tipo da interface.

() Justificativa: _____.

2. Quais tipos de exceção podem ser capturadas utilizando o bloco abaixo? Por que não é adequado utilizar este tipo de tratamento de exceções?

```
try{  
  
} catch(Exception e){}
```

3. Uma companhia paga seus empregados em uma base semanal. Os empregados são de quatro tipos:

1. Empregados assalariados são pagos um salário semanal, independentemente da quantidade de horas trabalhadas.
2. Empregados por hora são pagos pela quantidade de horas trabalhadas e recebem hora extra (1,5x o salário de uma hora tradicional) para cada hora depois de 40 horas trabalhadas.

3. Empregados de comissão são pagos por uma porcentagem de suas vendas.
4. Empregados de comissão assalariados são pagos com a porcentagem de suas vendas mais um salário base.

Escreva um método que receba qualquer um destes objetos e calcule seu pagamento semanal. Na resolução, não sobrecarregue os métodos: implemente somente um método que trate esses objetos de forma polimórfica.

4. Altere o exercício anterior e crie uma classe de exceção *CondicoesTrabalhollegais*. No método de calcular os ganhos de cada tipo de empregado, avalie se os empregados de comissão recebem menos que 10% de suas vendas e se os empregados por hora trabalharam mais de 60 horas na semana.

- Caso alguma dessas situações sejam encontradas, lance a exceção que você acabou de criar e a trate adequadamente.
 - Insira uma String na exceção, explicando a exceção que ocorreu e a exiba para o usuário no bloco catch.
 - Para avisar que os ganhos de algum empregado já foram calculados, crie um bloco finally junto ao try-catch que, ao terminar o cálculo de ganhos, seja normalmente ou depois de avisar ao usuário de que o empregado está em situação de trabalho irregular, avise que os seus ganhos já estão calculados.

Nota: Ao se criar um aplicativo Java, é importante não permitir que o programa pare de executar, exceto se o próprio usuário solicite isto (o mesmo deverá ser válido para o projeto prático). Portanto, neste exercício, caso alguma exceção seja lançada para algum empregado, avise o usuário com uma mensagem, não calcule os ganhos deste empregado, mas continue calculando dos outros em situações regulares de trabalho.