



Disciplina Programação Modular	Departamento Ciência da Computação	Turno Manhã/Noite	Período 2º
Professor Hugo de Paula (hugo@pucminas.br)			
Matrícula:	Aluno:		

## Exercício 4

1. Se métodos abstratos não são definidos, ou seja, não possuem nenhum corpo, porque eles são necessários?
2. A classe Ordenador a seguir implementa uma classe utilitária para ordenar um vetor de objetos ordenáveis utilizando o método da inserção. Um objeto ordenável deve seguir o padrão de comportamento da interface Ordenavel.

```
public interface Ordenavel {  
    public boolean menorQue(Ordenavel o);  
}  
  
public class Ordenador {  
    public static Ordenavel[] crescente(Ordenavel[] entrada){  
        Ordenavel temp;  
        Ordenavel[] saida = entrada.clone();  
        for (int i = 1; i < saida.length; i++) {  
            for(int j = i ; j > 0 ; j--){  
                if(saida[j].menorQue(saida[j-1])){  
                    temp = saida[j];  
                    saida[j] = saida[j-1];  
                    saida[j-1] = temp;  
                }  
            }  
        }  
        return saida;  
    }  
  
    public static Ordenavel[] decrescente(Ordenavel[] entrada){  
        Ordenavel temp;  
        Ordenavel[] saida = entrada.clone();  
        for (int i = 1; i < saida.length; i++) {  
            for(int j = i ; j > 0 ; j--){  
                if(!saida[j].menorQue(saida[j-1])){  
                    temp = saida[j];  
                    saida[j] = saida[j-1];  
                    saida[j-1] = temp;  
                }  
            }  
        }  
        return saida;  
    }  
}
```

Matrícula:	Aluno:
------------	--------

Modifique o código desenvolvido no Exercício 2 Questão 3 para que ele utilize classes abstratas. Veja se será necessário criar algum método abstrato ou transformar algum método já existente em abstrato. Faça com que a classe Data se torne Ordenavel. Implemente no método main da classe Aplicacao, uma lista de datas que seja inicializada aleatoriamente e depois imprima a lista original, a lista ordenada em ordem crescente e a lista ordenada em ordem decrescente.

Altere o diagrama UML para incluir Ordenador e Ordenavel, além de refletir as modificações feitas no código.

3. Baseado no código disponibilizado na questão anterior, altere a classe Pessoa do Exercício 3 Questão 3, para que ela também seja ordenável em ordem alfabética por nome. A classe pessoa deve ser abstrata? Ela deve possuir algum método abstrato? Escreva um comentário em Java no código da classe Pessoa explicando se ela deve ou não ser abstrata, justificando sua resposta. Faça as alterações necessárias na classe. Teste a implementação ordenando uma lista de clientes. Desenvolva todo o código necessário para que a aplicação funcione. Dependendo da forma como sua classe Cliente foi implementada, pode ser necessário que ela implemente a interface Ordenavel.

Atualize o diagrama UML para refletir as alterações do código.