

FCT/Unesp – Presidente Prudente
 Programação Orientada a Objetos
 Prof. Danilo Medeiros Eler
 Aula03 – Exercícios

Para cada um dos problemas apresentados abaixo, identifique as classes, os atributos e os métodos. Em seguida, faça a implementação dessas classes com alguma linguagem orientada a objetos, por exemplo, java. Além de implementar as classes para representar as entidades envolvidas no problema, você deve criar uma classe com um método principal (*main*) para realizar os testes. Adicionalmente, você deverá realizar os testes dos códigos implementados, para verificar se estão corretos, da mesma maneira como foi apresentado nos exemplos da Aula 02. Tais testes poderão ser desenvolvidos com a criação de algumas instâncias (objetos) das classes implementadas e a inserção de dados por meio dos métodos, bem como a utilização dos métodos para verificar se as classes atendem aos requisitos apresentados. Finalmente, após fazer os testes das classes e constatar que estão corretas, adicionar a entrada de dados pelo o usuário, ao invés de alimentar os objetos (inserção de dados em atributos) direto no código.

1) Os alunos da disciplina de Programação Orientada a Objetos da FCT/Unesp desenvolverão dois trabalhos e duas provas ao longo do semestre. Considere que, nesse caso, a média final é calculada pela média aritmética simples de todas as notas e que o aluno é aprovado somente se obtiver média maior ou igual a cinco.

O Aluno possui como informação o RA, Nome e as notas – a média é calculada a partir das notas e não precisa ser armazenada.

2) Uma academia mantém registro de seus alunos armazenando nome, idade, peso e altura. A academia faz um desconto especial para menores de idade, portanto, é necessário saber distinguir entre um aluno maior e menor. Além disso, a academia também tem interesse em acompanhar o desempenho de seus alunos, por isso, ela também necessita conhecer o índice de massa corporal (IMC) deles, sendo que $IMC = \text{peso} / \text{altura}^2$.

3) Um banco mantém contas de clientes armazenando o número da conta, o nome do cliente e o saldo atual da conta. Os clientes podem depositar o quanto quiserem nas contas, mas podem sacar somente enquanto a conta possuir saldo positivo.

4)) Implemente uma classe que represente um horário.

Os atributos são: hora, minuto e segundo.

Os métodos são:

- Um construtor sem parâmetros e um construtor com parâmetros
- Um método get e um método set para cada atributo
- Um método para exibir os valores dos atributos no seguinte formato hora:minuto:segundos – Exemplo: 19:45:43

- **Método:** void exibir()

- Um método para retornar o valor da hora em segundos. Por exemplo: 1:30:10 tem $1h * 3600s + 30min * 60s + 10s$ que resulta em 5410 segundos

- **Método:** int calcularSegundos()

5) Implemente uma classe que represente uma Conta Corrente Especial, em que os clientes possuem um limite negativo para saque.

Os atributos são: Titular, saldo e limite da conta.

Os métodos são:

- Um construtor sem parâmetros e um construtor com parâmetros
- Um método get e um método set para cada atributo
- Um método para exibir os valores dos atributos
- Um método para realizar depósito de dinheiro: void depositar(float valor)
 - Neste método deve-se somar o valor recebido por parâmetros ao saldo
- Um método para realizar saque: void saque(float valor)
 - Este método é um pouco diferente do método da conta corrente simples. Nesse caso, o saque pode ultrapassar o valor de saldo até um limite estabelecido. Por exemplo, se o saldo da conta é de 300 reais e 200 reais de limite, é permitido realizar saque de até 500 reais, pois esse tipo de conta permite ficar com 200 reais de saldo negativo, de acordo com o limite indicado. Portanto, deve-se verificar se a soma do saldo + o limite é maior ou igual ao valor a ser retirado. Se for o suficiente realize a operação; do contrário, emita uma mensagem de erro informando que o saldo é insuficiente.

6) Implemente uma classe para representar um eletrodoméstico.

Os atributos são: nome, potência (em W) e o tempo de utilização diária.

Os métodos são:

- Um construtor sem parâmetros e um construtor com parâmetros
- Um método get e um método set para cada atributo
- Um método para retornar a quantidade de Watts gastos por dia pelo eletrodoméstico: float wattsDia()
 - **Cálculo:** Potencia * tempo de utilização diária
- Um método para retornar a quantidade de Watts gastos por mês pelo eletrodoméstico: float wattsMes()
 - **Cálculo:** wattsDia() * 30
- Um método para retornar a quantidade de Watts gastos por ano pelo eletrodoméstico: float wattsAno()
 - **Cálculo:** wattsMes() * 12
- Um método para calcular o custo mensal em reais com o eletrodoméstico. Para isso deve-se passar como parâmetro para o método o custo do Kwatts. float gastoMes(float precoKwatts)
 - **Cálculo:** divida o wattsMes por 1000 e multiplique pelo preço dos Kwatts
- Um método para exibir um relatório, passando como parâmetro o custo do Kwatts.

Exemplo de relatório:

Eletrodoméstico: Telefone Sem Fio

Potência do aparelho: P=3.0 watts;

Tempo de utilização do aparelho: 24.0 horas/dia

Watts/Dia = 72.0 W

Watts/Mês = 2160.0 W

Watts/Ano = 25920.0 W

Preço do Kwatt R\$ 0.27

Gasto mensal é: R\$ 0.58320004

7) A Lista é uma estrutura de dados que pode ser representada computacionalmente de diferentes maneiras, por exemplo, um vetor, uma lista encadeada dinâmica, uma árvore, um *hash*, entre outras maneiras. A Lista, genericamente, pode ser vista como um tipo abstrato de dados em que as operações são conhecidas e comuns a qualquer tipo de implementação ou representação computacional. Faça a modelagem Orientada a Objetos do tipo abstrato de dados de uma Lista, a qual possui elementos armazenados e as seguintes operações: inicializar lista, verificar se lista está vazia, verificar se a lista está cheia, adicionar um elemento, recuperar um elemento, remover um elemento.

8) A FCT/Unesp é organizada por vários departamentos de ensino e cada um deles é composto por docentes. Um docente só pode pertencer a um único departamento. Desenvolva uma modelagem para ilustrar a organização das classes para um programa orientado a objetos que contemple a estrutura organizacional da FCT/Unesp.

Da universidade é necessário armazenar somente o nome e os departamentos.

Os dados que deverão ser armazenados para o departamento são: Código, Nome, Telefone, Funcionários.

Os dados que deverão ser armazenados para o funcionário são: Código, Nome, Função, Salário.

Como apresentado na aula de Sintaxe do Java, a entrada de dados no modo texto tem problemas quando é intercalada a leitura entre valores numéricos e strings. Para resolver esse problema, podemos fazer a leitura de uma linha após a leitura de um dado numérico, para que o fim de linha seja lido; ou, como outra alternativa, fazer a leitura dos dados de entrada como string (`nextLine`) e depois converte-lo para o tipo de dado necessário. Veja os exemplo a seguir, que ilustra a primeira opção.

```
class Principal{
    public static void main(String[] args) {
        Scanner scan = new Scanner (System.in);
        System.out.print ("Nome da primeira pessoa: ");
        String nome1 = scan.nextLine();
        System.out.print ("Peso em kg: ");
        int peso1 = scan.nextInt(); scan.nextLine();
        System.out.print ("Nome da segunda pessoa: ");
        String nome2 = scan.nextLine();
        System.out.print ("Peso em kg: ");
        int peso2 = scan.nextInt(); scan.nextLine();
    }
}
```