

Engenharia de Software – TAREFA T8 – Conceitos OOP

Prof. Dr. Aparecido V. de Freitas

EXERCÍCIO 1

- A) Com o emprego da Linguagem **Java**, escrever uma interface chamada **RegrasEscola** que contém os seguintes elementos:
- ✓ **LimiteFaltas** - com o valor constante **10**;
 - ✓ **MediaAprovacao** - com o valor constante **6.0**;
 - ✓ **RetornaMedia()** - que retorna um valor **double**.
- B) Com o emprego da Linguagem Java, escrever uma classe chamada Pessoa com os seguintes atributos de dados (propriedades): cpf, rg, endereco e nome. Considerar encapsulamento de dados. A classe deve conter métodos para construção de objetos (construtores) e um método ImprimePessoa() que irá imprimir os dados da pessoa.
- C) Com o emprego da Linguagem Java, escrever uma classe chamada AlunoColegio com os seguintes atributos de dados (propriedades): notaP1, notaP2, notaP3 e totalFaltas. A classe deve implementar a interface RegrasEscola e é subclasse da classe Pessoa. Considerar encapsulamento de dados. A classe deve conter o método imprimeAluno() que imprime os dados do aluno, o método RetornaMedia() que retorna a média aritméticas das três notas do estudante e uma função ImprimeResultado() que imprime "Aprovado com média xxxx" caso a média seja superior ou igual a 6,0 ou "Reprovado com média xxx" caso a média seja inferior a 6,0. Caso o estudante tenha ultrapassado o limite de faltas, estará automaticamente reprovado e a função deverá imprimir "Reprovado por faltas...".
- D) Com o emprego da Linguagem Java, criar uma classe chamada AlunoFaculdade com os seguintes atributos de dados (propriedades): notaP1, notaP2, notaP3 e totalFaltas. A classe deve implementar a interface RegrasEscola e é subclasse da classe Pessoa. Considerar encapsulamento de dados. A classe deve conter o método imprimeAluno() que imprime os dados do aluno, o método RetornaMedia() que retorna a média aritmética das duas maiores notas do estudante e uma função ImprimeResultado() que imprime "Aluno aprovado com média xxx" caso a média seja superior ou igual a 6,0 ou "Reprovado com média xxx" caso a média seja inferior a 6,0. Caso o estudante tenha ultrapassado o limite de faltas, estará automaticamente reprovado e a função deverá imprimir a mensagem "Reprovado por Falta...".
- E) Com o emprego da Linguagem Java, criar uma classe chamada TesteAluno com um método main para:
- ✓ Criar um aluno de colégio com os seguintes atributos: cpf = "234.567.876-10", rg = "5.345.234-9", endereço = "Rua x, 123", nome = "Paulo de Souza", notas: 7.0, 8.0 e 9.0 e total de faltas = 10.
 - ✓ Criar um aluno de faculdade com os seguintes atributos: cpf = "123.009.459-12 ", rg = "9.098.287-9", endereço = "Rua y, 3459", nome = "Ana Carolina Silva", notas: 1.5, 5.0 e 6.0 e total de faltas = 10.
 - ✓ Instanciar os objetos e executar as funções **ImprimeAluno()** e **ImprimeResultado()** para cada objeto criado.

Observação: As classes deverão ser criadas em um package chamado **maua**.

EXERCÍCIO 2

Com o emprego da Linguagem **Java**:

- A) Construir uma classe para representar um **Empregado** com os seguintes atributos de dados: **nome**, **endereço**, **valor da hora trabalhada** e **quantidade de horas trabalhadas no mês**. Considerando encapsulamento de dados, defina os construtores (padrão com passagem de parâmetros e sem passagem de parâmetros). Defina também um construtor que seja capaz de criar um empregado a partir de um empregado existente. Defina uma função chamada **ImprimeEmpregado()** que imprime os dados do empregado. Defina também uma função chamada **CalculaSalario()** que retorna o salário do empregado (basta multiplicar o valor da hora pela quantidade de horas trabalhadas). Defina por último uma função **ImprimeSalario()** que imprime o nome e salário do empregado.
- B) Definir uma classe para representar um **Gerente** que obviamente é um empregado. Considere que o **Gerente** tenha uma secretária (considerar o nome dela) e um valor de bônus que deve ser acrescido ao salário. Considerando encapsulamento de dados, defina os construtores (padrão com passagem de parâmetros e sem passagem de parâmetros). Defina também um construtor que seja capaz de criar um gerente a partir de um gerente existente. Defina uma função chamada **ImprimeGerente()** para imprimir todos os dados do gerente (inclusive os da classe Empregado). Reescreva a função **CalculaSalario()** para imprimir o salário do **Gerente** acrescido do bônus.
- C) Definir uma classe chamada **TesteGerente** com a função **main()** e criar um empregado de nome "Antonio", endereço: "Rua Brasil, 130", valor hora = R\$50,00 , 230 horas trabalhadas. Imprimir os dados do empregado. Imprimir o salário do empregado. Criar um gerente chamado "Carlos", endereço: "Rua Silvio Torres, 36", valor hora = R\$ 50,00, quantidade de horas trabalhadas = 230, secretária "Ana" e bônus = R2.000,00. Imprimir os dados do gerente e seu salário.

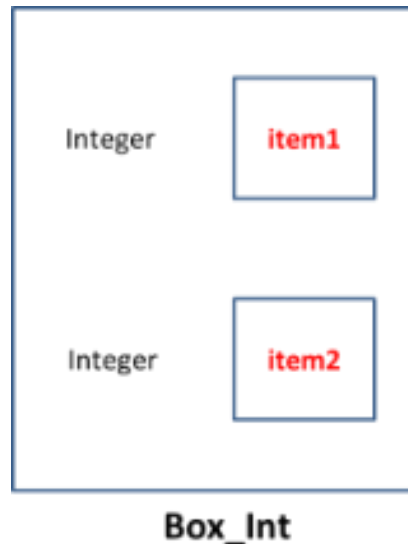
Observações:

- ✓ Considerar que todas as classes estão em um único package chamado maua.
- ✓ Considerar os conceitos de encapsulamento de dados e herança.

EXERCÍCIO 3

Com o emprego da Linguagem **Java**:

- A) Criar uma classe Java chamada **Box_Int** para modelar uma caixa (box) que irá armazenar dois números inteiros (par) com as variáveis de instância **item1** e **item2**.

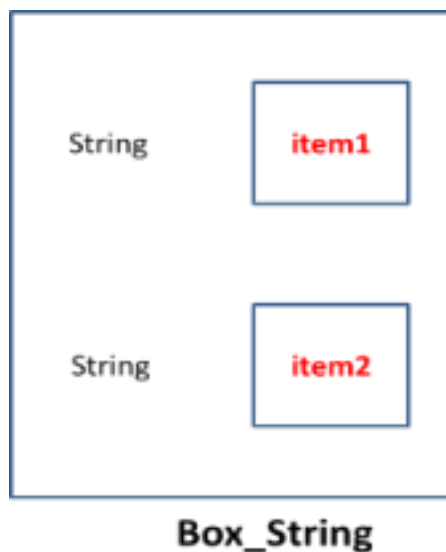


item1 – primeiro número (**Integer**)
item2 – segundo número (**Integer**)

A classe deve conter métodos para construir objetos, um método **ImprimeBoxInt()** que irá imprimir os dados **item1** e **item2** do objeto do tipo **Box_Int** e as funções **AlteraItem1()** e **AlteraItem2()** para permitir alteração dos dados.

Criar uma classe Java chamada **TesteBoxInt** para criar dois objetos (referenciados pelas variáveis **x** e **y**) com os dados (10,20) e (33,55). As classes devem ser criadas dentro de um **package** chamado **maua**.

- B) Criar uma classe Java chamada **Box_String** para modelar uma caixa (box) que irá armazenar dois strings (par) com as variáveis de instância **item1** e **item2**.

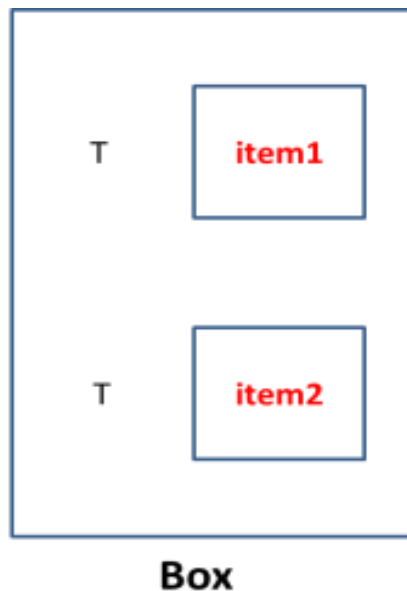


item1 – primeiro String (**String**)
item2 – segundo String (**String**)

A classe deve conter métodos para construir objetos, um método **ImprimeBoxString()** que irá imprimir os dados **item1** e **item2** do objeto do tipo **Box_String** e as funções **AlteraItem1()** e **AlteraItem2()** para permitir alteração dos dados.

Criar uma classe Java chamada **TesteBoxString** para criar dois objetos (referenciados pelas variáveis **x** e **y**) com os dados ("MAUA", "Computação") e ("Corinthians","Palmeiras"). As classes devem ser criadas dentro de um **package** chamado **maua**.

- C) Criar uma classe **GENÉRICA** chamada **Box** que irá armazenar dois itens de dados de um determinado tipo **T**.



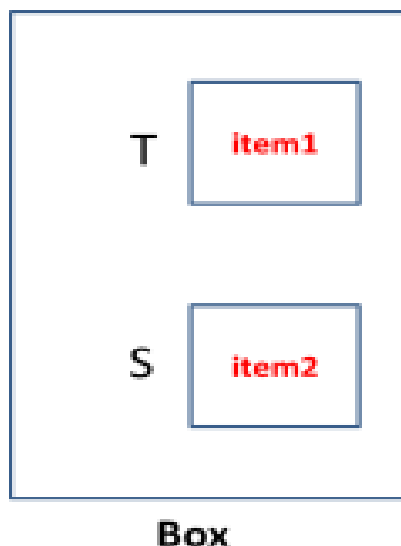
item1 – primeiro dado do tipo genérico **T**
item2 – segundo dado do tipo genérico **T**

A classe deve conter métodos para construir objetos, um método **ImprimeBox()** que irá imprimir os dados **item1** e **item2** do objeto do tipo **Box** e as funções **AlterarItem1()** e **AlterarItem2()** para permitir alteração dos dados.

Criar uma classe Java chamada **TesteBox** para criar dois objetos (referenciados pelas variáveis **x** e **y**) com os dados (10,20) e ("Computação","Maua").

As classes devem ser criadas dentro de um **package** chamado **maua**.

- D) Modificar a classe **Box** para que sejam armazenados dois itens de dados de tipos diferentes (**T** e **S**). Modificar a classe **TesteBox** para criar dois objetos (**x** e **y**) com os dados ("Maua", 99) e (33, "Palmeiras").



item1 – primeiro dado do tipo genérico **T**
item2 – segundo dado do tipo genérico **S**