

4 – 5 - LISTA DE EXERCÍCIO

Princípios da Programação Orientada a Objetos

Exercício 1

Crie uma classe **Mensagem** que apresente um atributo texto, implemente um construtor que permita a inicialização do atributo texto. Ao final, crie um método chamado escrever que apresente o valor contido no atributo texto acrescido da data e hora corrente.

Exercício 2

Analise a imagem abaixo:

Aluno
-nome -notaFinal -disciplina
+ Aluno(String n1, float n2, String n3) + void mostrarAluno() + void mostrarSituacao()

Crie uma classe Aluno, conforme apresentado na figura acima, e implemente o método **main** para representar o contexto de utilização deste exemplo.

Exercício 3

Criar uma classe **Ordem** cujos atributos são variáveis **double**: x, y e z. A classe **Ordem** deverá possuir os métodos indicados na tabela abaixo.

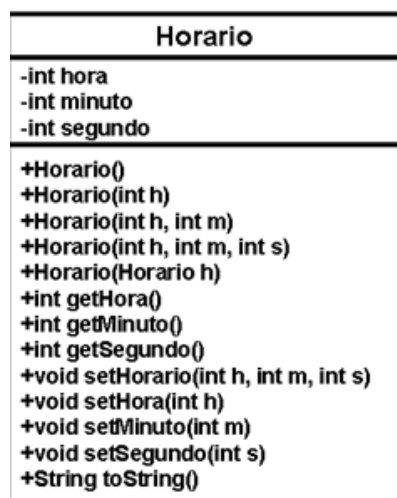
Método	Descrição
public Ordem(double a, double b, double c)	Construtor com parâmetros.
public void mostrarMaior()	Indica qual campo possui o maior valor e qual é esse valor.
public void mostrarMenor()	Indica qual campo possui o menor valor e qual é esse valor.
public void mostrarCrescente()	Mostra em ordem crescente os valores contidos em x, y e z.

public void mostrarDecrescente()	Mostra em ordem decrescente os valores contidos em x, y e z.
---	--

Criar uma classe com o método **main** que cria um objeto da classe ordem e testa todos os métodos da tabela acima.

Exercício 4

Construir uma Classe **Horario** cujo diagrama UML é dado na figura abaixo.



Dado diagrama UML da figura, pede-se:

Item (A): Implementar os métodos tal como dado no detalhamento dos métodos dado na tabela abaixo.

Método	Descrição
Horario()	Construtor sem parâmetros que inicializa os campos hora, minuto e segundo com o valor zero. Chama o método setHorario().
Horario(int h)	Construtor que inicializa o campo hora com o valor h e os demais com valor zero. Chama o método setHorario().
Horario(int h, int m)	Construtor que inicializa o campo hora com o valor h e o campo minuto com o valor m. Chama o método setHorario().
Horario(int h, int m, int s)	Construtor que inicializa o campo hora com o valor h, o campo minuto com m e o campo segundo com s. Chama o método setHorario().
Horario(Horario H)	Construtor que inicializa o campo hora com H.hora, o campo minuto com o valor contido em H.minuto e o campo segundo com o valor contido em

	H.segundo. Chama o método setHorario().
int getHora()	Método que retorna o valor contido no campo hora.
int getMinuto()	Método que retorna o valor contido no campo minuto.
int getSegundo()	Método que retorna o valor contido no campo segundo.
void setHorario(int h, int m, int s)	Método que é chamado por todos os construtores para modificar os campos hora, minuto e segundo com os valores h, m e s, respectivamente. Estas modificações são realizadas através de chamadas aos métodos setHora(), setMinuto() e setSegundo(), respectivamente.
void setHora(int h)	Método que modifica o valor do campo hora para o valor h.
void setMinuto(int m)	Método que modifica o valor do campo minuto para o valor m.
Void setSegundo(int s)	Método que modifica o valor do campo segundo para o valor s.

Item (B): Implementar mais métodos tal como dado no detalhamento dos métodos dado na tabela abaixo.

Método	Descrição
void somarHoras(Horario h2)	Método que realiza a soma de dois objetos da classe hora h1 e h2 e armazena o resultado em h1. A chamada ao método pode ser realizada através do comando: h1.somarHoras(h2); Cuidado, o valor máximo de um horário é: 23:59:59. Se a soma for maior, mostrar uma mensagem apropriada e retornar 23:59:59.
void subtrairHoras (Horario h2)	Método que realiza a soma de dois objetos da classe hora h1 e h2 e armazena o resultado em h1. A chamada ao método pode ser realizada através do comando: h1.subtrairHoras(h2); Cuidado, o valor mínimo de um horário é: 00:00:00. Se a subtração for menor, mostrar uma mensagem apropriada e retornar 00:00:00.
int numSegundos()	Método que retorna o número de segundos contidos em um dado objeto da classe Horario .
void mostrarAMPM()	Método que mostra o conteúdo dos campo segundo, minuto e hora em notação militar, ou seja, se a hora for 23:40:01 será mostrada, com este método, da seguinte forma: 11:40:01 PM. Se a hora for 11:40:01 (antes do meio-dia), então, será mostrada da seguinte forma: 11:40:01.

Exercício 5

Construir uma classe **Data** cujos campos e métodos são dados na figura abaixo.

Data
-int dia -int mes -int ano
+Data(int d, int m, int a) +void avançar() +void voltar() +boolean anoBissexto() +int diasDoMes() +void irParaFrente(int n) +void irParaTras(int n) +String toString()

Para implementar os métodos da figura acima segue o detalhamento dos métodos dado na tabela abaixo.

Método	Descrição
Data(int d, int m, int a)	Construtor com três parâmetros inteiros d, m e a que inicializam os campos dia, mês e ano, respectivamente, através do método setData().
setData(int d, int m, int a)	Método responsável por modificar os campos dia, mes e ano com os valores d, m e a, respectivamente. Se os valores d, m e a não forem válidos, então, usar os valores 1, 1 e 1900, respectivamente.
void avançar()	Método que deve avançar em 1 dia a data atual, mantendo os campos da classe em estado coerente.
void voltar()	Método que reduz em 1 dia a data atual, mantendo os campos da classe em estado coerente.
Boolean anoBissexto()	Método que retorna true se O campo ano contiver um ano tal que: (1) Seja divisível por 4, mas não por 100. (2) Seja divisível por 100 e 400.
int diaDoMes()	Método que retorna o número de dias do mês, considerando se o ano é bissexto ou não.
Void irParaFrente (int n)	Método que incrementa em n dias a data contida nos campos dia, mes e ano, mantendo estado coerente.
Void irParaTras (int n)	Método que decrementa em n dias a data contida nos campos dia, mes e ano, mantendo estado coerente.