

Décima Primeira Lista de Exercícios

Interfaces

Norton Trevisan Roman

16 de junho de 2011

1. Seu chefe está fazendo um programa para manipulação de figuras geométricas. Como o programa é muito complexo, ele pediu a sua ajuda encomendando a você a implementação de três classes (Quadrado, Retângulo e Círculo). Você é livre para definir como será a implementação mas seu chefe definiu que as suas classes devem implementar a seguinte interface:

```
interface Figura2D {  
    double calculaÁrea() ;  
    double calculaPerímetro();  
    void mudaCor(String cor);  
    String pegaCor();  
}
```

2. Escreva duas implementações para a interface VeículoDeCorrida, a seguir:

```
interface VeículoDeCorrida {  
    String marca();  
    String modelo();  
    String cor();  
    int potênciaEmCavalos();  
}
```

Agora, escreva um método veículoPreferido() que recebe um array de veículos de corrida como parâmetro e, dentre os veículos vermelhos, imprime a marca e o modelo do que possuir a maior potência.

3. Você foi contratado para trabalhar em um berçário (!!!) e no seu primeiro dia de trabalho, deve escrever uma classe para informar aos pais dados sobre os seus bebês. A dificuldade é que muitos estrangeiros freqüentam esse berçário e o seu programa deve ser capaz de dar informações em português e em inglês (e, futuramente, em outros idiomas também). Para permitir isso, você deverá prover duas implementações da interface a seguir, que indica as mensagens de texto que deverão ser mostradas aos pais:

```
interface MensagensSobreNeoNatal {  
    String nomeDoBebe(String nome);  
    String dataDeNascimento(Bebe b);  
    String peso(double pesoEmQuilos);  
    String temperatura(double temperaturaEmCelsius);  
}
```

Ao se chamar o método `nomeDoBebe`, por exemplo, sua implementação deve devolver uma mensagem como “O nome do bebê é Godofredo Manoelino de Moraes”. Ao se chamar o método `peso`, deve-se devolver algo como “O peso do bebê é 3140 gramas”. A primeira implementação deve se chamar `MensagensBrasileiras` e a segunda `MensagensEstadosunidenses`. As mensagens para os americanos devem apresentar o peso em libras (pounds) e a temperatura em graus fahrenheit. A interface `Bebê`, que vocês não precisam implementar, pois podem supor que o berçário já possui as implementações, é a seguinte:

```
interface Bebe {
    String nome();
    int diaNascimento();
    int mesNascimento();
    int anoNascimento();
    double peso(); // Kg
    double temperatura(); // celsius
}
```

A classe que o berçário vai usar para imprimir as informações para os pais terá um método similar ao seguinte:

```
class Berçário {
    Bebê[] listaDeBebes;

    // aqui vão outros métodos para inserção e remoção de bebês da lista

    void imprimeDadosSobreBebe(Bebe b, MensagensSobreNeoNatal m) {
        System.out.println(m.nomeDoBebe(b.nome()));
        System.out.println(m.dataDeNascimento(b));
        System.out.println(m.peso(b.peso()));
        System.out.println(m.temperatura(b.temperatura()));
    }

    void imprime(int IDdoBebe, String idioma) {
        MensagemSobreNeoNatal m;
        if (idioma.equals("Português")) m = new MensagensBrasileiras();
        else m = new mensagensEstadosunidenses();
        imprimeDadosSobreBebe(listaDeBebes[IDdoBebe], m);
    }
}
```

4. Dada a classe

```
class Pessoa {
    String nome;
    int rg;
    int cpf;
}
```

escreva duas implementações da interface

```
interface LocalizadorDePessoas {
    Pessoa localizaPorRg(int rg, Pessoa[] vp);
}
```

utilizando busca seqüencial e busca binária.

5. Implemente uma Interface denominada SalarioFuncionário, com os seguintes métodos:
 - defineNome(String nome): define nome do funcionário
 - valorHoraBase(double hora): define o valor base da hora trabalhada
 - calculaHE(int quantidade): calcula o valor que o funcionário receberá referente a horas-extras, considerando uma determinada quantidade de horas extras
 - calculaSalarioFinal():
6. Construa as classes FuncionárioComum e Gerente, que implementam a interface do exercício anterior, observando:
 - o valor base da hora (tanto do funcionário quanto do gerente devem ser informados quando os objetos forem instanciados);
 - o valor base da hora do funcionário não pode ser maior que 200. Se for fornecido um valor maior que este o sistema deve emitir uma mensagem de erro;
 - o valor de hora-extra do gerente é calculado considerando 20% sobre o da hora base;
 - o valor de hora-extra do funcionário é calculado considerando 100% sobre o valor da hora base, mas ele pode fazer no máximo 40 horas-extras.
 - o salário final do funcionário é calculado sobre 200 horas mensais.
 - o salário final do gerente é calculado sobre 180 horas mensais mais uma comissão fixa no valor de R\$2000,00.
 - a classe FuncionarioComum deve ter métodos para definir o cargo do funcionário (String) e a data de admissão.
7. Construa uma classe para instanciar 2 funcionários e 3 gerentes usando as especificações dos exercícios anteriores. Imprima o valor da hora base, o nome do funcionário, a quantidade de horas-extras, o valor total de horas-extras e o salário final. Para os funcionários comuns, imprima também o cargo e a data de admissão.
8. Construa uma classe que forneça métodos para:
 - Criar e inicializar um array de funcionários com 10 posições.
 - Imprimir listas de funcionários ordenadas por valor da hora-base, nome e data de admissão (use interfaces para evitar repetição de código).
9. Considere as classes Livro e Biblioteca, dadas a seguir. Deseja-se imprimir duas listagens distintas, sendo a primeira em ordem alfabética crescente de título e a segunda em ordem crescente de ano. Utilizando a interface ComparadorDeLivro dada a seguir, faça:
 - Duas classes que implementem a interface fornecida e que sejam usadas para imprimir as listagens solicitadas;
 - Um método para ordenar um vetor de Livros. Este método deve utilizar obrigatoriamente objetos das classes criadas no item anterior, com conceito de interface, para ordenar o vetor de acordo com o atributo solicitado (título ou ano);

- Complete o método main da classe Biblioteca nos locais indicados para imprimir as listagens solicitadas.

```

class Livro {
private String titulo;
private int ano;
private int tipo; // valor de 1 a 20 que classifica a categoria de cada livro

void setTitulo (String t) {titulo= t;}
void setAno (int a) {ano = a;}
void setTipo (int ti) {tipo = ti;}
String getTitulo () {return titulo;}
int getAno () { return ano;}
int getTipo () { return tipo;}
}

interface ComparadorDeLivro {
    /*
        retorna: 1 se atributo específico de livro1 é maior que livro2
        -1 se atributo específico de livro1 é menor que livro2
        0 se atributo específico de livro1 é igual a livro2
    */
    int livroMenor(Livro livro1, Livro livro2);
}

class Biblioteca {
    // método para criar vetor de Livros
    // aqui vai o código para criar o vetor de Livros (você não precisa completar)

    // método imprime
    // aqui vai o código para imprimir o vetor de Livros (você não precisa completar)

    // método para ordenar o vetor
    //... aqui vai código para ordenar o vetor (você deve apresentá-lo completo-)

    // método principal
    public static void main(String[] args) {
        Biblioteca bibl= new Biblioteca();

        System.out.println("Livros desordenados");
        bibl.imprime();

        System.out.println("Livros em ordem de título")
        //(você deve completar aqui)

        System.out.println("Livros em ordem de ano")
        //(você deve completar aqui)
    }
}

```