

	FACULDADE 7 DE SETEMBRO		
	Curso: Sistemas de Informação		
	Disciplina: Programação Orientada a Objetos	Professor: Robério Gomes Patricio	Data: 24/04/2014
Aluno:		Avaliação:	
VP2			

Observações:

- Prova SEM CONSULTA e INDIVIDUAL;
- A interpretação das questões faz parte da avaliação.

1. Escreva uma classe que implementa a interface CharSequence encontrada no pacote java.lang. A implementação deve retornar a string em ordem reversa. Escolha uma das frases de sua preferência para usar como os dados. Escreva uma classe com um pequeno método principal para testar sua classe; certifique-se de chamar todos os quatro métodos da classe CharSequence.
2. Escreva uma definição de classe para uma classe abstrata, Vehicle, que contém:
 - a. uma variável double, maxspeed
 - b. uma variável de instância double protegida, currentspeed
 - c. um construtor aceitando um double usado para inicializar a variável de instância maxspeed
 - d. um método abstrato, acelerar, que não aceita parâmetros e retorna nada.
 - e. um método getCurrentSpeed que retorna o valor de currentspeed
 - f. um método getMaxSpeed que retorna o valor de maxspeed
 - g. um método, pedalToTheMetal, que chama repetidamente até acelerar a velocidade do veículo é igual a maxspeed. pedalToTheMetal deve retornar nada.

O programa principal escrito abaixo irá chamar os métodos da classe. Você precisa apenas escrever a classe como mencionado acima.

```

1  import java.util.Scanner;
2
3  public class Testdemo extends Vehicle{
4
5      public testdemo(double no){
6          super(no);
7      }
8
9      void accelerate(){
10         System.out.print(currentSpeed+" ");
11         currentSpeed++;
12     }
13
14     public static void main(String[] arg){
15         Scanner scanner =new Scanner(System.in);
16         System.out.println("Enter the number:");
17         int n=scanner.nextInt();
18         testdemo td=new testdemo(n);
19         System.out.println("Current Speed:"+td.getCurrentSpeed());
20         System.out.println("Max Speed:"+td.getMaxSpeed());
21         td.pedalToTheMetal();
22     }
23 }

```

3. Dada a existência de uma classe de Phone. Defina uma subclasse, CameraPhone, que contém duas variáveis de instância: um inteiro chamado, imageSize, que representa o tamanho em megapixels (para simplificar assumir um pixel ocupa um byte – assim um megapixel é igual a um megabyte) de cada imagem (ou seja, 2 significa que cada imagem é composta por dois megapixels), e um número inteiro chamado memorySize, representando o número de gigabytes de memória da câmara (isto é, 4 significa 4 Gigabytes de memória).

Há um construtor que aceita dois parâmetros inteiros correspondentes às duas variáveis de instância acima e que são usados para inicializar as respectivas variáveis de instância.

Há também um método chamado numPictures que retorna (como um inteiro), o número de imagens de memória da câmera pode conter.

Não se esqueça que um gigabyte é de 1.000 megabytes.

O programa principal escrito abaixo irá chamar os métodos da classe. Você precisa apenas escrever a classe como mencionado acima.

```
1 import java.util.Scanner;
2
3 public class Inheritance{
4     public static void main(String[] arg){
5         Scanner scanner =new Scanner(System.in);
6         System.out.println("Enter the imagesize:");
7         int imagesize=scanner.nextInt();
8         System.out.println("Enter the memorysize:");
9         int memorysize=scanner.nextInt();
10
11         CameraPhone cph=new CameraPhone(imagesize,memorysize);
12         System.out.println("imagesize:"+cph.getImageSize());
13         System.out.println("memorysize:"+cph.getMemorySize());
14         System.out.println("numPictures:"+cph.numPictures());
15     }
16 }
```