PROGRAMAÇÃO ORIENTADA A OBJETOS – 2007/2
2ª LISTA DE EXERCÍCIOS – ENTREGA: DURANTE A 1ª PROVA (03/10)

1.Desenvolva uma classe de nome **CodigoPostal**, cujas instâncias sejam capazes de guardar o Código Postal de uma dada rua. Note que cada Código Postal é constituído por dois números inteiros, que designaremos respectivamente por "indicativo" e "extensão", e o nome da rua (Ex: 38408-046 Armando Lombardi). Deverão poder ser criados códigos postais dados:

· "indicativo", "extensão" e rua

· "indicativo" e rua (ficando nesse caso a extensão igual a zero)

· nenhum parâmetro (ficando nesse caso os atributos numéricos a zero e a rua com a mensagem "Indisponível".

Para além dos usuais métodos seletores (***getters***) e modificadores (***setters***) implemente também o método mostra, cuja evocação, permita visualizar a informação relativa a um determinado código postal no formato:

CEP: 38408 - 046 Armando Lombardi
(ou CEP: Indisponível se for o caso).

Escreva um programa de teste para a classe **CodigoPostal**. Crie várias instâncias da classe e teste os vários métodos implementados.

2. A empresa ***BadSoft*** desenvolveu uma classe **Xpto** capaz de armazenar 3 valores do tipo ***int*** como se indica a seguir:

```
// Arquivo: Xpto.java
public class Xpto {
        //VARIAVEIS DE INSTANCIA
          public int a;
          public int b;
          public int c;

        //CONSTRUTORES
          public Xpto() {
                a=0;
                b=0;
                c=0;
          };

         public Xpto (int a1, int b1, int c1) {
                a=a1;
                b=b1;
                c=c1;
         };

         //METODOS
         public int produto(){
                return (a*b*c);
         };

         public int soma(){
                return (a+b+c);
         };
}; // class Xpto
```

Note que os autores da classe **Xpto** não incluíram, na mesma, métodos seletores (***getters***) e métodos modificadores (***setters***) capazes de acessar ou alterar os valores das respectivas variáveis de instância. Estas foram declaradas como ***public*** pelo que o respectivo acesso pode fazer-se, do exterior da classe, de forma direta sem recurso a tais métodos. Vários programadores adquiriram a classe **Xpto** e utilizaram-na nos seus programas. Um desses programas, designado por **Exemplo1**, é indicado a seguir:

```
// Arquivo: Exemplo1.java
public class Exemplo1 {
        public static void main(String[] args) {
                Xpto x1=new Xpto();
                Xpto x2= new Xpto(5,6,7);
                x1.a=1;
                x1.b=2;
                System.out.println(x1.soma());
                System.out.println(x2.soma());
                x1.c=x2.a;
                x2.a=x2.a+1;
                System.out.println(x1.c);
                System.out.println(x2.a);
                //.....
        };
}; // class Exemplo1
```

Passado algum tempo os engenheiros da ***BadSoft*** acharam que seria muito melhor guardar os 3 valores do tipo ***int*** num vetor. Assim lançaram um ***update*** da classe **Xpto** com a seção correspondente às variáveis de instância substituída por:

```
//VARIAVEIS DE INSTANCIA
public int v[];
```

a) Tendo em conta esta alteração indique que outras modificações tiverem os engenheiros da ***BadSoft*** de efetuar a nível de construtores e métodos de forma a manter a classe **Xpto** funcional.
b) Indique também quais as alterações que o programador, cliente da ***BadSoft***, teria de efetuar no seu programa **Exemplo1**.

3. A ***GoodSoft***, concorrente da ***BadSoft***, também lançou no mercado uma classe semelhante com o nome **Ypto**. Os engenheiros da *GoodSoft* optaram por declarar as variáveis de instância com a visibilidade ***private*** (e não ***public*** como os seus colegas da ***BadSoft***) sendo o respectivo acesso feito à custa de métodos seletores (***getters***) e métodos modificadores (***setters***)como se indica a seguir:

```
// Arquivo: Ypto.java
public class Ypto {
        //VARIÁVEIS DE INSTÂNCIA
        private int a;
        private int b;
        private int c;

        //CONSTRUTORES
        public Ypto() {
                a=0;
                b=0;
                c=0;
        };

        public Ypto(int a1, int b1, int c1) {
                a=a1;
                b=b1;
                c=c1;
        };

        //MÉTODOS SELETORES
        public int getA() {
                return a;
        };

        public int getB(){
                return b;
        };

        public int getC(){
                return c;
        };

        //MÉTODOS MODIFICADORES
        public void setA(int a1){
                a=a1;
        };

        public void setB(int a1){
                b=a1;
        };

        public void setC(int a1){
                c=a1;
        };

        //OUTROS MÉTODOS
        public int produto(){
                return (a*b*c);
        };

        public int soma(){
                return (a+b+c);
        };
}; // class Ypto
```

Vários programadores adquiriram a classe **Ypto** e utilizaram-na nos seus programas. Um desses programas, designado por **Exemplo2**, é indicado a seguir:

```
// Arquivo: Exemplo2.java
public class Exemplo2 {
        public static void main(String[] args)
{
                Ypto x1=new Ypto();
                Ypto x2= new Ypto(5,6,7);
                x1.setA(1);
                x1.setB(2);
                System.out.println(x1.soma());
                System.out.println(x2.soma());
                x1.setC(x2.getA());
                x2.setA(x2.getA()+1);
                System.out.println(x1.getC());
                System.out.println(x2.getA());
                //.....
        }; // main
}; // class Exemplo2
```

Os engenheiros da ***GoodSoft*** também concluíram que seria muito melhor se os 3 valores fossem guardados num vetor e por isso também lançaram um ***update*** da classe **Ypto**.
a) Tendo em conta esta alteração indique que outras modificações tiverem os engenheiros da ***GoodSoft*** de efetuar a nível de construtores e métodos por forma a manter a classe **Ypto** funcional.
b) Indique também quais as alterações que o programador, cliente da ***GoodSoft***, teria de efectuar no seu programa **Exemplo2**.
c) Que conclusão podemos tirar destes dois exercícios ?

4. Desenvolva uma classe Java que modela um objeto **Livro**, que contém ainda um método construtor e um método (***public static void***) ***main***. Um livro possui um título e quantidade de páginas. Use ***String*** para representar o título. Use inteiros para representar a quantidade de páginas. O construtor deve receber através de argumentos, os dois dados suficientes para criar um livro. O método ***main*** deve criar um ***array*** com quatro posições, e quatro livros com dados quaisquer, cada um deles associado a uma posição do ***array***.

5. Desenvolva uma classe pública denominada **Grupo**, que permita representar um grupo de trabalhos constituído por dois alunos, onde se incluem os seguintes membros públicos:
a) um construtor, onde são passados os dois alunos do grupo e uma nota do grupo (que é atribuída pela avaliação de um trabalho de grupo).
b) um método que devolve a média das notas individuais dos alunos do grupo (atenção que esta não é a nota do grupo);
c) um método que devolve o índice de desvio das notas individuais dos alunos em relação à nota do grupo; este índice é dado pela seguinte fórmula:

$$\sqrt{(Ng-a)^2+(Ng-b)^2}$$

onde ***Ng*** é a nota do grupo e ***a***,***b*** são as notas individuais de cada um dos membros do grupo.