Compartilhar o seu link com: luciorocha @ professores.utfpr.edu.br

Michael Pariz Pereira: Cópia de Aula 13 - POCO4A - 22/09/2022 - Turma A - Exercícios propos…
Leonardo G. Fagote Cópia de Aula 13 - POCO4A - 22/09/2022 - Turma A - Exercícios propostos
Matheus H. A. Pereira: Cópia de Aula 13 - POCO4A - 22/09/2022 - Turma A - Exercícios propo…
Vitor Luis de Queiroz Batista: Cópia de Aula 13 - POCO4A - 22/09/2022 - Turma A - Exercícios…
Rodrigo Leandro Benedito: Cópia de Aula 13 - POCO4A - 22/09/2022 - Turma A - Exercícios pr…
Raphael Uematsu: Cópia de Aula 13 - POCO4A - 22/09/2022 - Turma A - Exercícios propostos
Lucas Prado: Cópia de Aula 13 - POCO4A - 22/09/2022 - Turma A - Exercícios propostos
Ruan Perondi Urbanjos: Copy of Aula 13 - POCO4A - 22/09/2022 - Turma A - Exercícios propo…
Arnaldo: Aula 13 - 22/09
Gabriel Candelária: Cópia de Aula 13 - POCO4A - 22/09/2022 - Turma A - Exercícios propostos
Mariana Pedroso Naves: Aula 13 - POO

Leitura recomendada: Apostila de Interfaces e Classes Abstratas: Apostila_cap11_poo.tif

Leitura recomendada: Apostila de Classes Internas e Classes Anônimas: Apostila_cap13_poo.tif

Exemplo de Classe Interna

```
public class Externa {
        private int var=111;

        public class Interna {
              public void imprimir(){
                  System.out.println(“INTERNA: ” + var);
              }
        }

        public void imprimir(){
              System.out.println(“EXTERNA”);
       }

       public static void main( String [ ] args ){

            Externa externa = new Externa();
            externa.imprimir();
            Externa.Interna  interna = new Externa().new Interna();
            interna.imprimir();
```

```
    }
}
```

```
public class Externa {
        private int var=111;

        //public class Interna { //Estah disponivel para outras classes
        //private class Interna { //Classes externas nao tem acesso
        private static class Interna {   //Classe isolada
                private int var=222;
                public void imprimir(){
                    System.out.println("INTERNA: " + var);
                }
        }

        public void imprimir(){
                System.out.println("EXTERNA");
       }

       public static void main( String [ ] args ){

             Externa externa = new Externa();
             externa.imprimir();
             //Externa.Interna  interna = new Externa().new Interna();
             //interna.imprimir();

            Interna interna = new Interna();
            interna.imprimir();
       }
}
```

Exercícios

1) Observe o diagrama a seguir:

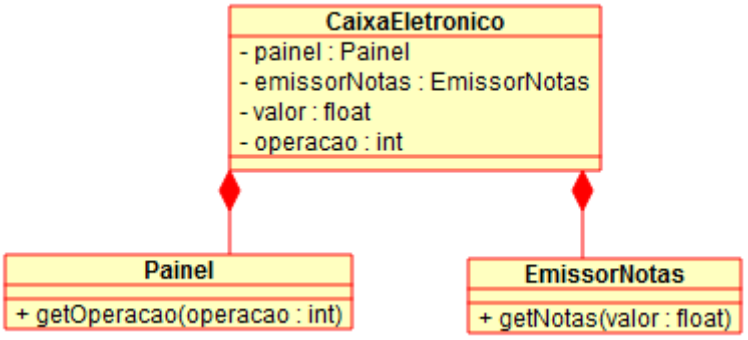


Figura 1 - Diagrama UML de Classes.

a) Implemente um programa orientado a objetos que utilize classes internas para implementar o diagrama da Figura 1.

```
package caixaeletronico;

/**
 *
 * @author a2150336
 */
public class CaixaEletronico {

   private float valor;
   private int operacao;
   private EmissorNotas emissorNotas;
   private Painel painel;

   public CaixaEletronico(){
      this.painel = new Painel();
      this.emissorNotas = new EmissorNotas();
      this.valor = 1;
      this.operacao = 1;
   }
   public class Painel {

      public int getOperacao() {
         return operacao;
      }

   }

   public class EmissorNotas {

      public float getNotas() {
         return valor;
      }
   }

   public void main(String[] args) {
      CaixaEletronico caixaEletronico = new CaixaEletronico();
   }

}
```

b) Utilize classes internas com o modificador de acesso ‘static’.

```
package caixaeletronico;

/**
 *
 * @author a2150336
 */
public class CaixaEletronico {

   private EmissorNotas emissorNotas;
   private Painel painel;

   public CaixaEletronico() {
      this.painel = new Painel();
      this.emissorNotas = new EmissorNotas();
   }

   private static class Painel {

      private int operacao;

      public Painel() {
         this.operacao = 1;
      }

      public int getOperacao() {
         return operacao;
      }

   }

   private static class EmissorNotas {

      private float valor;

      public EmissorNotas() {
         this.valor = 1;
      }

      public float getNotas() {
         return valor;
      }
   }

   public void main(String[] args) {
      CaixaEletronico caixaEletronico = new CaixaEletronico();
   }

}
```

Exemplo de Classe Interna Anônima com Classe Concreta

```
public class Externa {

    public abstract class Interna {
        protected int var = 111;
        public abstract void imprimir();
    }

    public Externa(){

        //Classe interna anonima: eh uma subclasse da classe

        Interna interna = new Interna(){  //Classe interna anonima
                public void imprimir(){
                        System.out.println("222" + var);
                }
        };
        interna.imprimir();

    }

    public static void main( String [ ] args ){
          Externa externa = new Externa();
    }

}
```

Exemplo de Classe Interna Anônima com Classe Abstrata

```
public class Externa {

    public abstract class Interna {
        protected int var = 111;
        public void imprimir();
    }

    public Externa(){

        //Classe interna anonima: eh uma subclasse da classe

        Interna interna = new Interna(){  //Classe interna anonima
                public void imprimir(){
```

```
                    System.out.println("222" + var);
                }
        };
        interna.imprimir();

    }

    public static void main( String [ ] args ){
         Externa externa = new Externa();
    }

}
```

Exemplo de Classe Interna Anônima com Interface

```
public class Externa {

    public interface  Interna {
        final int var = 111;
        public abstract void imprimir();
    }

    public Externa(){

        //Classe interna anonima: eh uma classe que implementa a interface

        Interna interna = new Interna(){  //Classe que implementa a interface
                public void imprimir(){
                    System.out.println("222" + var);
                }
        };
        interna.imprimir();


    }

    public static void main( String [ ] args ){
         Externa externa = new Externa();
    }

}
```

Exercícios

1) Observe o diagrama a seguir:

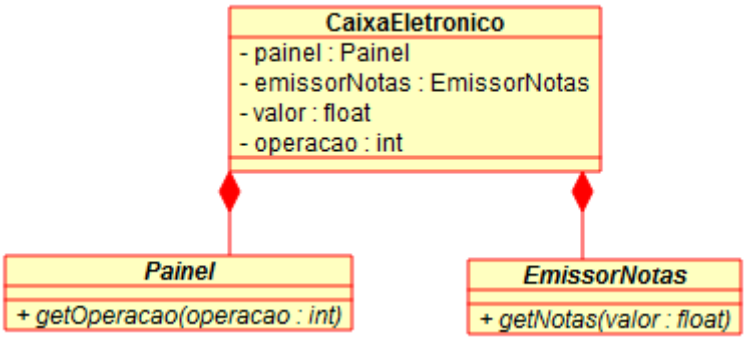


Figura 2 - Diagrama UML de Classes.

a) No diagrama da Figura 2, as classes Painel e EmissorNotas são classes abstratas com todos os métodos abstratos. Implemente um programa orientado a objetos que utilize classes internas anônimas para implementar o diagrama da Figura 2.

b) As classes internas anônimas, da mesma forma que as outras classes, invocam o construtor da superclasse. Ilustre um exemplo.

c) A Classe interna anônima pode acessar os membros da sua Classe de primeiro nível. Ilustre um exemplo.

d) Modifique o exemplo anterior para que Painel e EmissorNotas sejam interfaces. Implemente um programa orientado a objetos que utilize classes internas anônimas para implementar o diagrama da Figura 2.

```
package caixaeletronico;

/**
 *
 * @author a2150336
 */
public class CaixaEletronico {

   private float valor;
   private EmissorNotas emissorNotas;
   private Painel painel;
   private int operacao;

   public CaixaEletronico() {
```

```
      this.painel = new Painel() {
         public int getOperacao() {
            return operacao;
         }
      };
      this.emissorNotas = new EmissorNotas() {
         public float getNotas() {
            return valor;
         }
      };
   }

   private abstract class Painel {

      public Painel() {

      }

   }

   private abstract class EmissorNotas {

      public EmissorNotas() {

      }

   }

   public void main(String[] args) {
      CaixaEletronico caixaEletronico = new CaixaEletronico();
   }

}
```

```
public class CaixaEletronico {

   private Painel painel;
   private EmissorNotas emissorNotas;




   public CaixaEletronico(){
      painel = new Painel(){ //classe interna anonima
          public int getOperacao(){
              return operacao;
```

```
        }
    };
    System.out.println( painel.getOperacao() );
    painel.imprimir();

    emissorNotas = new EmissorNotas(){ //classe interna anonima
      public float getNotas(){
          return valor;
      }
    };
    System.out.println( emissorNotas.getNotas() );


    IPainel  painel2 = new IPanel(){
        public int getOperacao(){
            return operacao;
        }
        public void imprimir(){
            System.out.println("CLASSE INTERNA");
        }
    };
    System.out.println(painel2.getOperacao() );
    painel2.imprimir();
  }

public interface IPainel {
    final int operacao=20;
     public abstract int getOperacao();
     public abstract void imprimir();
}


  public abstract class Painel{

    protected int operacao;

    public Painel(){
      operacao = 20;
    }
    public abstract int getOperacao();
    public void imprimir(){
        System.out.println("CLASSE ABSTRATA");
    }
  }

  public abstract class EmissorNotas{

    protected float valor;
```

```
    public EmissorNotas(){
       valor = 0;
    }
     public abstract float getNotas();
  }

  public static void main(String[] args) {
     CaixaEletronico caixaEletronico = new CaixaEletronico();
  }
}
```

2) Faça a implementação Orientada a Objetos do problema anunciado a seguir:

a) Crie 3 (três) classes não relacionadas por herança: Construcao, Carro e Bicicleta.

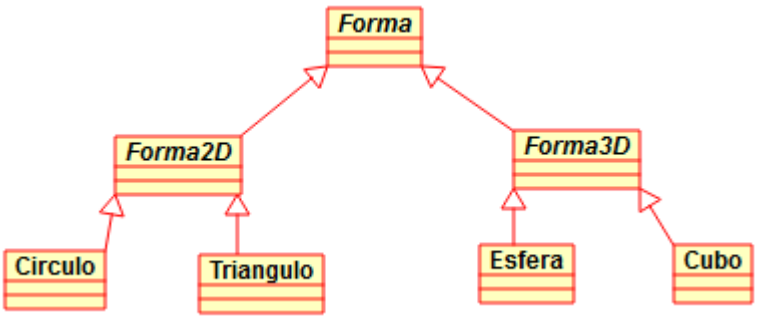
b) Dê a cada Classe atributos e comportamentos únicos que não estão presentes em outras classes.

c) Crie a Interface EmissaoCarbono com um método getEmissaoCarbono.

d) Cada Classe deve implementar a Interface EmissaoCarbono.

e) Invoque o método getEmissaoCarbono de cada objeto.

3) Observe a Figura 2 a seguir:



a) Implemente a hierarquia de Classes mostrada na Figura. Apenas as Classes folha são Classes concretas, as demais são classes abstratas.

b) A Classe Forma2D deve conter o método getArea.

c) A Classe Forma3D dever conter os métodos getArea e getVolume.

d) Crie uma Classe Principal que tenha um vetor de Formas com objetos de cada Classe concreta.

e) O programa deve imprimir o tipo de cada objeto instanciado.