

7º Aula prática

1. Código fonte:

- Apresentador.java:

```
package controleeventos;

import java.util.ArrayList;
import java.util.List;

public class Apresentador extends Participante {

    private String titulo;

    public Apresentador(String n, Evento e, String t) {
        super(n, e);
        titulo = "\"" + t + "\"";
    }

    protected String getTitulo() {
        return titulo;
    }

    public String nomeLista(){
        return getNome()+"\n";
    }
}
```

- ApresentadorArtigo.java:

```
package controleeventos;

import java.util.ArrayList;
import java.util.List;

public class ApresentadorArtigo extends Apresentador{

    public ApresentadorArtigo(String n, Evento e, String t){
        super(n,e,t);
    }
}
```

```

    }

    public String certificado () {

        return "Declaramos que " + getNome() + " apresentou o artigo " +

            getTitulo() + " no " + getEvento().getNome();

    }

    public String nomeLista(){

        return getNome()+"\n";

    }

}

```

- ApresentadorPoster.java:

```
package controleeventos;
```

```

public class ApresentadorPoster extends Apresentador {

    public ApresentadorPoster(String n, Evento e, String t) {

        super(n,e,t);

    }

    public String certificado () {

        return "Declaramos que " + getNome() + " apresentou o poster " +

            getTitulo() + " no " + getEvento().getNome();

    }

    public String nomeLista(){

        return getNome()+"\n";

    }

}

```

- ControleEventos.java:

```
package controleeventos;

import javax.swing.*.*;

/**
 *
 * @author marcmaia
 */

public class ControleEventos {

    public static void main (String args[]) {

        String n = JOptionPane.showInputDialog("Entre com o nome do Evento");

        int max = Integer.parseInt(JOptionPane.showInputDialog("Entre com o maximo de participantes"));

        Evento e = new Evento(n,max);

        int opcao;

        do {

            opcao = Integer.parseInt(JOptionPane.showInputDialog("Entre com opcao: 0 - Participantes simples; 1 - Palestrante; 2 - Apres. Poster; 3 - Apres. Artigo; 4 - Sair"));

            String t;

            if (opcao != 4) {

                n = JOptionPane.showInputDialog("Entre com o nome do participante");

                if (opcao == 0)

                    e.addParticipante(new Participante(n,e));

                else {

                    t = JOptionPane.showInputDialog("Entre com o titulo");

                    if (opcao == 1)

                        e.addParticipante(new Palestrante(n,e,t));

                    else if (opcao == 2)
```

```

        e.addParticipante(new ApresentadorPoster(n,e,t));

    else if (opcao == 3)

        e.addParticipante(new ApresentadorArtigo(n,e,t));

    }

}

} while (opcao != 4);

JOptionPane.showMessageDialog(null, e.certificados());

JOptionPane.showMessageDialog(null, e.listaDePresenca());

}

}

```

- Evento.java:

```

package controleeventos;

public class Evento {

    private static final String SEPARADOR = "\n===== \n";

    private String nome;

    private Participante participantes[];

    private int nroParticipantes;

    public Evento(String n, int maxParticipantes) {

        nome = n;

        participantes = new Participante[maxParticipantes];

    }

    // Metodo que adiciona um participante no arranjo

    // Convencao: se o arranjo nao estiver cheio, adiciona e

    // retorna o proprio participante. Se o arranjo estiver

    // cheio retorna null.

```

```
public Participante addParticipante(Participante p) {  
    if (nroParticipantes == participantes.length) {  
        return null;  
    }  
    return participantes[nroParticipantes++] = p; // insere e depois incrementa nroParticipantes  
}
```

```
public String getNome() {  
    return nome;  
}
```

```
public String certificados() {  
    String resultado = "";  
    for (int i = 0; i < participantes.length && participantes[i] != null; i++) {  
        resultado += SEPARADOR + participantes[i].certificado() + SEPARADOR;  
    }  
    return resultado;  
}
```

```
public String listaDePresenca() {  
    String resultado = "";  
    for (int i = 0; i < participantes.length && participantes[i] != null; i++) {  
        resultado += participantes[i].nomeLista();  
    }  
    return resultado;  
}  
}
```

- Palestrante.java:

```
package controleeventos;
```

```
public class Palestrante extends Apresentador {
```

```
    public Palestrante(String n, Evento e, String t) {
```

```
        super(n, e, t);
```

```
    }
```

```
    public String nomeLista(){
```

```
        return "";
```

```
    }
```

```
    public String certificado() {
```

```
        return "Declaramos que " + getNome() + " apresentou a palestra "
```

```
            + getTitulo() + " no " + getEvento().getNome();
```

```
    }
```

```
}
```

- Participante.java:

```
/*
```

```
 * To change this license header, choose License Headers in Project Properties.
```

```
 * To change this template file, choose Tools | Templates
```

```
 * and open the template in the editor.
```

```
*/
```

```
package controleeventos;
```

```
/**
```

```
 *
```

```
 * @author marcmaia
```

```
 */
```

```
public class Participante {
```

```
private String nome;

private Evento evento;

public Participante(String n, Evento e) {

    nome = n;

    evento = e;

}

public String certificado() {

    return "Declaramos que " + nome + " participou do "

        + evento.getNome();

}

protected String getNome() {

    return nome;

}

protected Evento getEvento() {

    return evento;

}

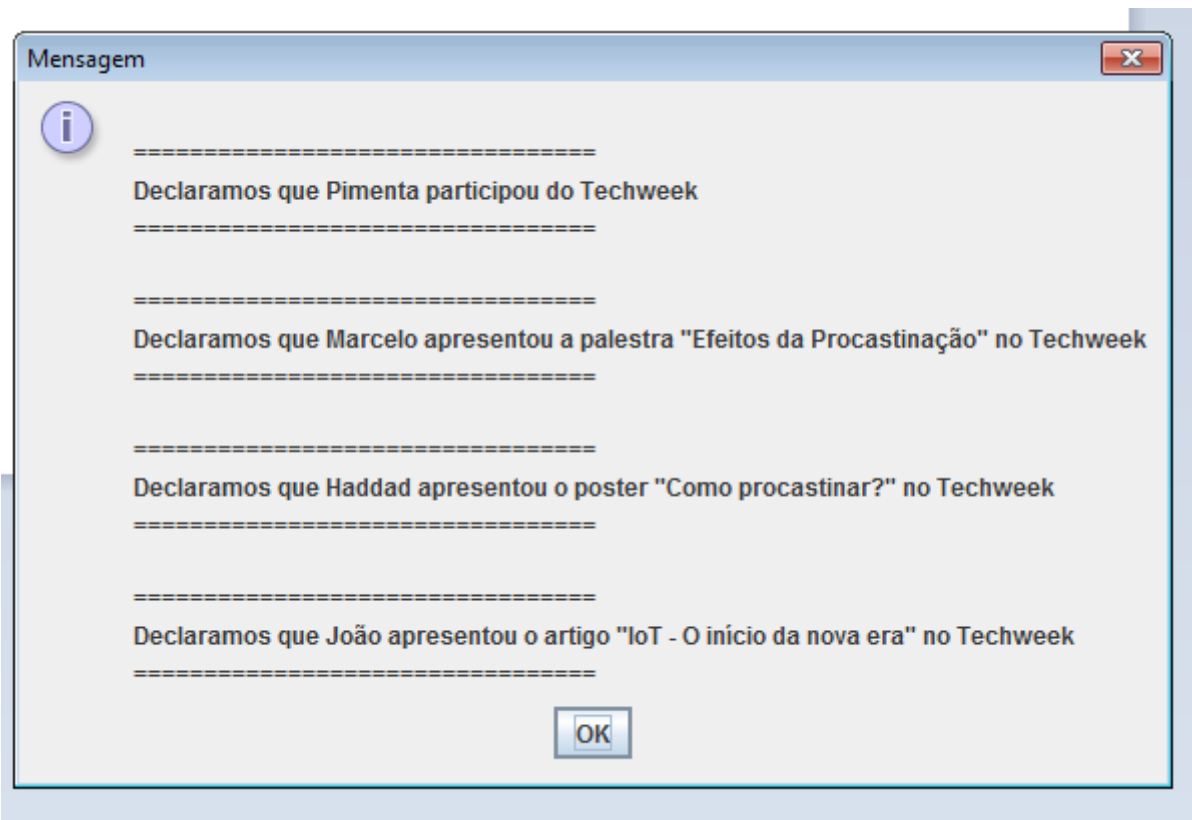
public String nomeLista(){

    return getNome()+"\n";

}

}
```

2. Print do Funcionamento:
- Certificados:



- Lista de presença:

