

T8

Hall of Fame / Shame

(Galeria da Fama / Vergonha)

Profa Andréa Schwertner Charão

Alberto Francisco Kummer Neto

DLSC/CT/UFSM

Feedback com coleta automatizada

```
=== Magnos André Hammes (mahcoder) ===Bandeja.java:
No bad input detected.
Main.java:
No bad input detected.
Bolacha.java:
No bad input detected.
BolachaCircular.java:
No bad input detected.
BolachaRetangular.java:
No bad input detected.
BolachaTriangular.java:
No bad input detected.
Calc.java:
No bad input detected.
Ponto.java:
No bad input detected.
=== ana luisa v. solórzano (anaveroneze) ===Bolacha.java:
No bad input detected.
BolachaApp.java:
No bad input detected.
Calculo.java:
No bad input detected.
InsereForma.java:
Field "alt" seems to be a constant but lacks of "final" modifier within class "InsereForma"
Field "larg" seems to be a constant but lacks of "final" modifier within class "InsereForma"
Field "tam" seems to be a constant but lacks of "final" modifier within class "InsereForma"
Ponto.java:
Field "distCentroMax=Math.sqrt(InsereForma.getTamanho())/50)/2" seems to be a constant within class "Ponto"
=== João Machado (jgmachado90) ===Bolacha.java:
Field "REDONDA=0" seems to be a constant within class "Bolacha"
Field "RETANGULAR=1" seems to be a constant within class "Bolacha"
Field "TRIANGULAR=2" seems to be a constant within class "Bolacha"
Circulo.java:
No bad input detected.
Forma.java:
No bad input detected.
Ponto.java:
No bad input detected.
Retangulo.java:
No bad input detected.
TrabalhoDoForno.java:
No bad input detected.
Triangulo.java:
No bad input detected.
=== Filipe Simões (fsimoess) ===Bolacha.java:
No bad input detected.
Circulo.java:
No bad input detected.
Ponto.java:
Field "x=0" seems to be a constant but lacks of "final" modifier within class "Ponto"
Field "y=0" seems to be a constant but lacks of "final" modifier within class "Ponto"
Retangulo.java:
No bad input detected.
Triangulo.java:
No bad input detected.
```

Feedback com coleta automatizada

Aluno	Arquivo:Classe	1	2	3	4
Ana Luisa	InsereForma.java:InsereForma Ponto.java:Ponto		distCentroMax		alt larg tam
Filipe Sim	Ponto.java:Ponto				x y
Pablo Ca	Assadeira.java:Assadeira Bolacha.java:Bolacha Bolacha.java:Bolacha			altura largura area ponto	se id
Daniel Ma	Bolacha.java:Bolacha			tipo	
Caroline	Bolacha.java:Bolacha bolachaRetangulo.java:bolachaRetangulo Ponto.java:Ponto Bolacha.java:Bolacha			tam ponto tipo base altura coordX coordY tam ponto	
Adonai G	Array.java:Array			Bolachas	
João Vito	Pont.java:Ponto				x y
Vinícius T	Bolacha.java:Bolacha bolachaRetangulo.java:bolachaRetangulo Ponto.java:Ponto			tam ponto base altura coordX coordY	
Francisco	Bolacha.java:Bolacha Ponto.java:Ponto				tipo x y
Leonardo	chapa.java:Chapa	gerador			
Gabriel C	Ponto.java:Ponto	posX			posX

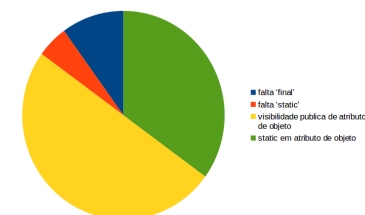
2

1

11

7

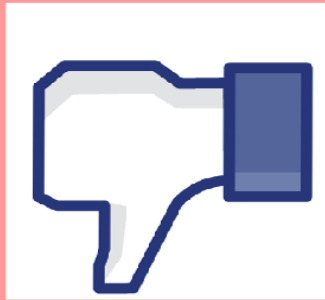
N#	Descrição
1	falta 'final'
2	falta 'static'
3	visibilidade publica de atributo de objeto
4	static em atributo de objeto



Dados gerais

- Observações baseadas em 17 trabalhos

Hall of Shame



Hall of Fame



Hall of ??: Encapsulamento

```
public class Ponto {  
    private Integer x;  
    private Integer y;  
  
    public Ponto() {...}  
  
    public Ponto(Integer x, Integer y) {...}  
  
    public void setX(Integer x) {...}  
  
    public void setY(Integer y) {...}  
  
    public Integer getX() {...}  
  
    public Integer getY() {...}  
  
}
```

Hall of Fame: Encapsulamento

- Atributos privados
- Métodos públicos

```
public class Ponto {  
  
    private Integer x;  
    private Integer y;  
  
    public Ponto() {...}  
  
    public Ponto(Integer x, Integer y) {...}  
  
    public void setX(Integer x) {...}  
  
    public void setY(Integer y) {...}  
  
    public Integer getX() {...}  
  
    public Integer getY() {...}  
  
}
```

Hall of ??: Encapsulamento

```
public class Bolacha {  
    public double tam;  
    public Ponto ponto;  
    public String tipo;  
    ...|  
}
```

Hall of Shame: Encapsulamento

- Todos os atributos públicos?!

```
public class Bolacha {  
  
    public double tam;  
    public Ponto ponto;  
    public String tipo;  
  
    ...|  
  
}
```


Hall of ??: Uso de static

```
public class Ponto{  
    public static double x = 0;  
    public static double y = 0;  
  
    public Ponto(){  
    }  
  
    public Ponto(double dx, double dy){  
        x += dx;  
        y += dy;  
    }  
}
```

Hall of Shame: Uso de static

- Posição x e y são atributos da classe?!
(= mesmo valor para todos os pontos)

```
public class Ponto{  
    public static double x = 0;  
    public static double y = 0;  
  
    public Ponto(){  
    }  
  
    public Ponto(double dx, double dy){  
        x += dx;  
        y += dy;  
    }  
}
```

Hall of ??: Herança

```
abstract class Bolacha {  
    abstract public double area();  
    abstract public String tipo();  
}
```

```
    public class BolachaRedonda extends Bolacha {  
  
        Ponto a = new Ponto();  
        double raio;  
  
        ...  
    }
```

Hall of Fame: Bom uso de herança

- Relação “is-a” (é um)
- BolachaRedonda **é uma** Bolacha

```
abstract class Bolacha {  
    abstract public double area();  
    abstract public String tipo();  
}
```

```
public class BolachaRedonda extends Bolacha {  
  
    Ponto a = new Ponto();  
    double raio;  
  
    ...  
}
```

Hall of ??: Herança

```
public abstract class Array {  
    public Ponto[] Bolachas;  
    public abstract float getArea();  
}
```

```
public class Bolacha {  
    Array b;  
    ...  
}
```

```
public class Circulo extends Array{  
    float raio;  
  
    public Circulo(Ponto p, float raio){  
        this.raio = raio;  
    }  
  
    @Override  
    public float getArea() {  
        float area;  
        area = 3.14f * raio * raio;  
        return area;  
    }  
}
```

Hall of Shame: Uso 'suspeito' de herança

■ Círculo é um Array?!

```
public abstract class Array {  
    public Ponto[] Bolachas;  
    public abstract float getArea();  
}
```

```
public class Bolacha {  
    Array b;  
  
    ...  
}
```

```
public class Circulo extends Array{  
  
    float raio;  
  
    public Circulo(Ponto p, float raio){  
        this.raio = raio;  
    }  
  
    @Override  
    public float getArea() {  
        float area;  
        area = 3.14f * raio * raio;  
        return area;  
    }  
}
```

Hall of Fame:

Herança e polimorfismo evitam repetições

- Cada tipo de Bolacha tem seu método para calcular a area

```
Bolacha v[] = new Bolacha[50];
for(int i=0; i < 50; i++){
    ...
    if(v[i].area() > maior){
        maior = v[i].area();
        index = i;
    }
    ...
}
System.out.print("Maior area: " + v[index].area() + ". " + v[index].tipo() + "\n");
```

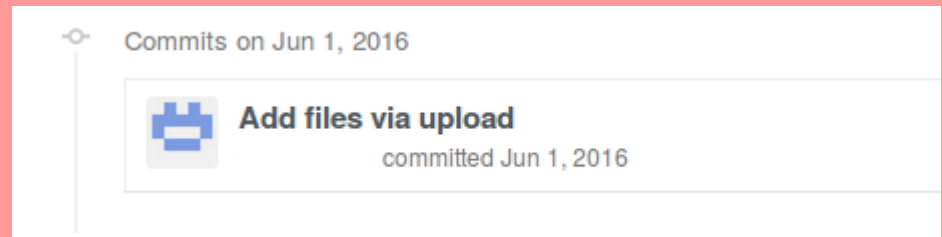
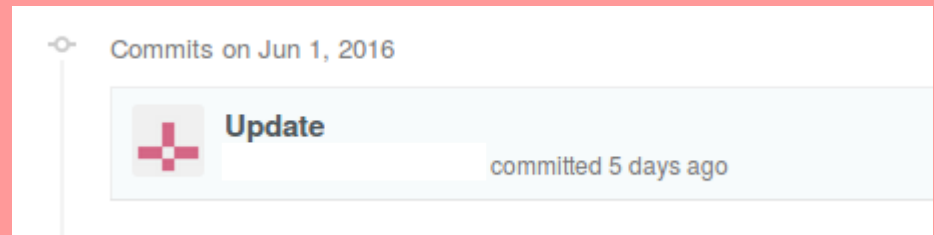
Hall of Shame:

Repetição de código desnecessária

```
for (BolachaCircular bc : bolachas_circulares)
{
    tmp = bc.getSize();
    if (tmp > size)
    {
        size = tmp;
        maior_bc = bc;
    }
}
size = 0.0;
for (BolachaRetangular br : this.bolachas_retangulares)
{
    tmp = br.getSize();
    if (tmp > size)
    {
        size = tmp;
        maior_br = br;
    }
}
size = 0.0;
for (BolachaTriangular bt : this.bolachas_triangulares)
{
    tmp = bt.getSize();
    if (tmp > size)
    {
        size = tmp;
        maior_bt = bt;
    }
}
if ((maior_bc.getSize() > maior_br.getSize()) && (maior_bc.getSize() > maior_bt.getSize()))
{
    System.out.println(maior_bc);
}
else if ((maior_br.getSize() > maior_bc.getSize()) && (maior_br.getSize() > maior_bt.getSize()))
{
    System.out.println(maior_br);
}
else if ((maior_bt.getSize() > maior_bc.getSize()) && (maior_bt.getSize() > maior_br.getSize()))
{
    System.out.println(maior_bt);
}
```


Hall of Shame: Uso do versionamento

- **Um** commit adicionando todo código de uma vez
- Commit e pushes **pouco descritivos:** como saber **quando e quais** modificações foram acrescentadas?



Hall of Fame: Uso do versionamento

- Trabalho desenvolvido incrementalmente, com commits e pushes frequentes no repositório



Commit frequentes
com título explicativo

Hall of Fame: Uso do versionamento

- Trabalho desenvolvido incrementalmente, com commits e pushes frequentes no repositório

