

Pontifícia Universidade Católica de Minas Gerais - Instituto de Ciências Exatas e Informática
Bacharelado em Ciência da Computação
Engenharia de Software II
Trabalho IV

Componentes: Daniel Gunna Santana da Silva Souza¹ e Felipe Coelho Silva²

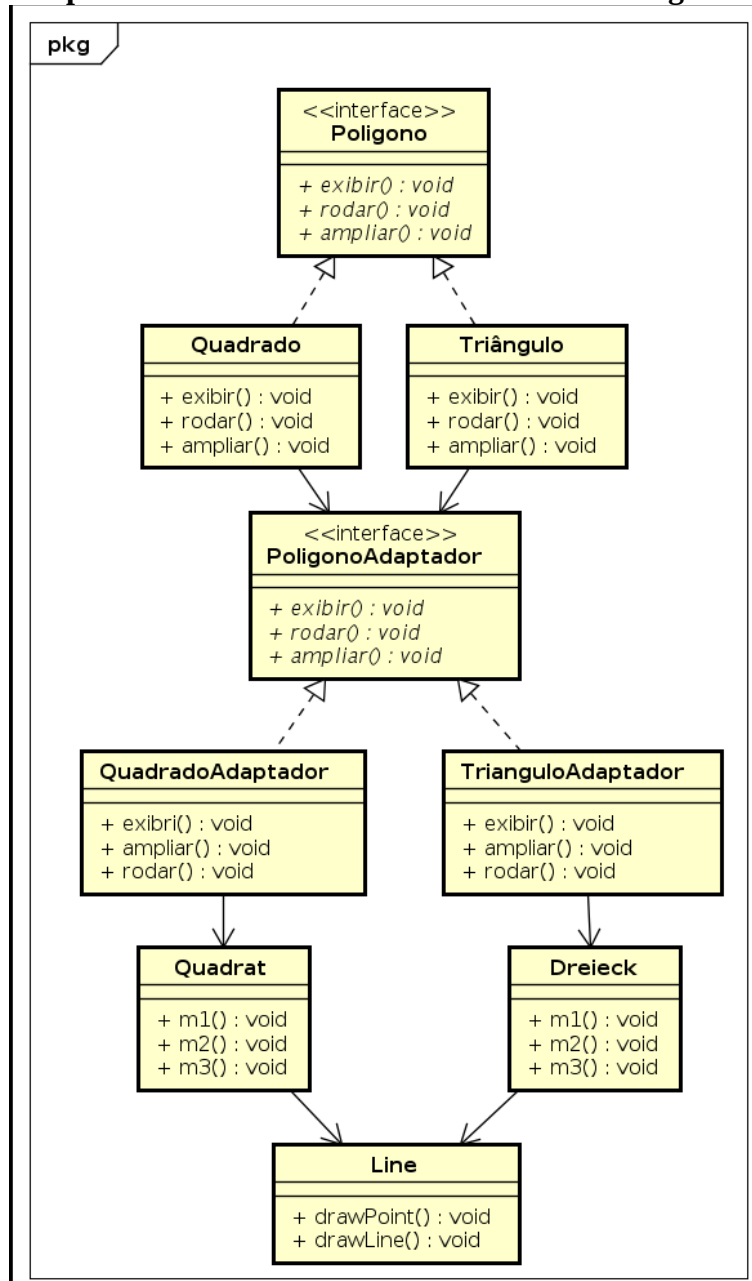
¹Aluno do Bacharelado em Ciência da Computação(daniel.gunna@sga.pucminas.br)

²Aluno do Bacharelado em Ciência da Computação(fcs4200@gmail.com)

Questão (01) -

Criaria uma interface para adaptar as chamadas aos métodos das classe dreiecke e quadrat e chamaria as funções de desenho por esta interface através de duas classes adaptadoras que chamam os reais métodos das classes em alemão.

Para solução do problema de utilizar as classes em alemão segue o diagrama abaixo:



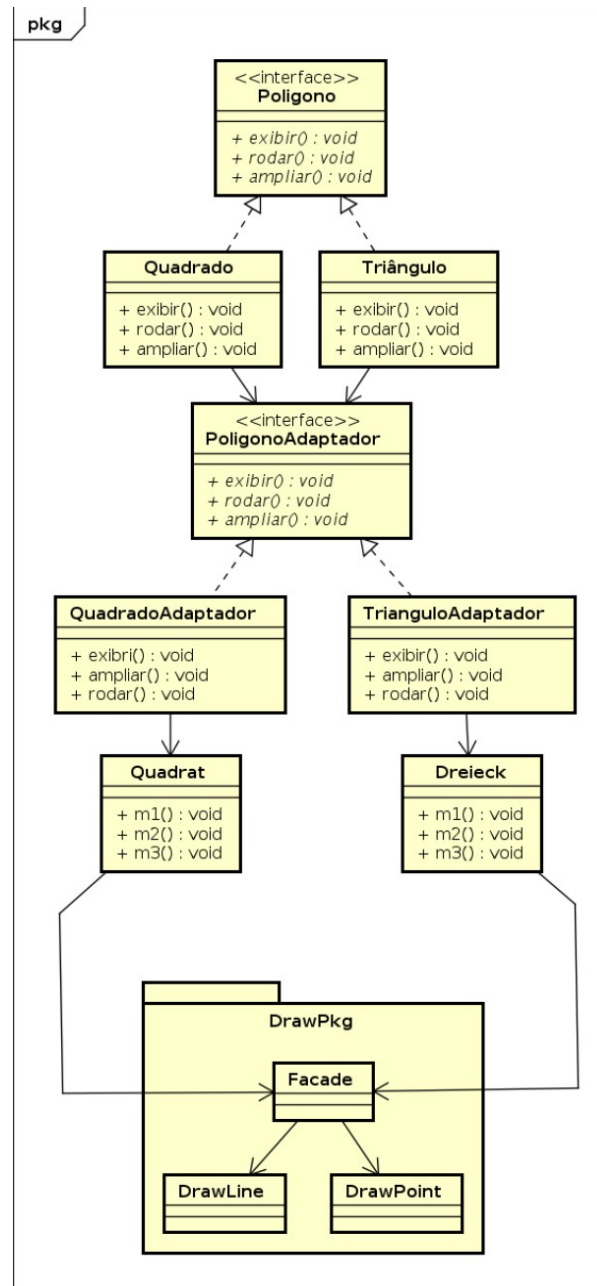
Pontifícia Universidade Católica de Minas Gerais - Instituto de Ciências Exatas e Informática
Bacharelado em Ciência da Computação
Engenharia de Software II
Trabalho IV

Componentes: Daniel Gunna Santana da Silva Souza¹ e Felipe Coelho Silva²

¹Aluno do Bacharelado em Ciência da Computação(daniel.gunna@sga.pucminas.br)

²Aluno do Bacharelado em Ciência da Computação(fcs4200@gmail.com)

Questão (02)



Pontifícia Universidade Católica de Minas Gerais - Instituto de Ciências Exatas e Informática
Bacharelado em Ciência da Computação
Engenharia de Software II
Trabalho IV

Componentes: Daniel Gunna Santana da Silva Souza¹ e Felipe Coelho Silva²

¹Aluno do Bacharelado em Ciência da Computação(daniel.gunna@sga.pucminas.br)

²Aluno do Bacharelado em Ciência da Computação(fcs4200@gmail.com)

Esqueleto do código:

```
public interface Poligono{

    public void exibir();
    public void ampliar();
    public void rodar();
}

public class Quadrado implements Poligono{

    @Override
    public void exibir(){

    }

    @Override
    public void ampliar(){

    }

    @Override
    public void mostrar(){

    }

}

public class Triangulo implements Poligono{

    @Override
    public void exibir(){

    }

    @Override
    public void ampliar(){

    }

    @Override
    public void mostrar(){

    }

}

public interface Adaptador(){
    public void exibir();
    public void ampliar();
    public void rodar();
}

public class TrianguloAdaptador implements Adaptador{

    @Override
```

Pontifícia Universidade Católica de Minas Gerais - Instituto de Ciências Exatas e Informática
Bacharelado em Ciência da Computação
Engenharia de Software II
Trabalho IV

Componentes: Daniel Gunna Santana da Silva Souza¹ e Felipe Coelho Silva²

¹Aluno do Bacharelado em Ciência da Computação(daniel.gunna@sga.pucminas.br)

²Aluno do Bacharelado em Ciência da Computação(fcs4200@gmail.com)

```
        public void exibir(){
new Dreiecke().m1();
}

        @Override
        public void ampliar(){
new Dreiecke().m2();
}

        @Override
        public void mostrar(){
new Dreiecke().m3();
        }

}

public class QuadradoAdaptador implements Adaptador{

        @Override
        public void exibir(){
new Quadrat().m1();
}

        @Override
        public void ampliar(){
new Quadrat().m2();
        }

        @Override
        public void mostrar(){
new Quadrat().m3();
        }

}

public class Quadrat{

        public void m1(){
new Facade.drawQuadrat();
        }

        public void m2(){
new Facade.zoomQuadrat();
        }

        public void m3(){
new Facade.rotateQuadrat();
        }
}
```

Pontifícia Universidade Católica de Minas Gerais - Instituto de Ciências Exatas e Informática
Bacharelado em Ciência da Computação
Engenharia de Software II
Trabalho IV

Componentes: Daniel Gunna Santana da Silva Souza¹ e Felipe Coelho Silva²

¹Aluno do Bacharelado em Ciência da Computação(daniel.gunna@sga.pucminas.br)

²Aluno do Bacharelado em Ciência da Computação(fcs4200@gmail.com)

```
}

public class Dreiecke{

    public void m1(){
new Facade.drawDreiecke();
    }

    public void m2(){
new Facade.zoomDreiecke();
    }

    public void m3(){
new Facade.rotateDreiecke();
    }

}

public class Facade {

    void drawDreiecke(){ new DrawLine.draw ... ; new DrawPoint.draw ...; }
    void zoomDreiecke(){ new DrawLine.draw ... ; new DrawPoint.draw ...;}
    void rotateDreiecke(){ new DrawLine.draw ... ; new DrawPoint.draw ...;}
    void rotateQuadrat(){new DrawLine.draw ... ; new DrawPoint.draw ...; }
    void zoomQuadrat(){ new DrawLine.draw ... ; new DrawPoint.draw ...;}
    void drawQuadrat(){ new DrawLine.draw ... ; new DrawPoint.draw ...;}

}

public class DrawLine {
    void draw(){ ... }
}

public class DrawPoint {
    void draw(){ ... }
}
```