**UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO**

SISTEMAS DE INFORMAÇÃO

**GRUPO:** CAIO FERNANDES

JOÃO VÍTOR FEIJÓ

LORENA VASCONCELLOS

**RELATÓRIO JAVA**

Primeiro trabalho da disciplina de LP3

RIO DE JANEIRO

2022

CAIO FERNANDES

JOÃO VÍTOR FEIJÓ

LORENA VASCONCELLOS

**RELATÓRIO JAVA**

Primeiro trabalho da disciplina de LP3

Trabalho da disciplina de Linguagem de Programação III lecionada pelo prof. Claver Pari Soto do curso de Sistemas de Informação da Universidade Federal Rural do Rio de Janeiro.

RIO DE JANEIRO

2022

1. INTRODUÇÃO

Este relatório tem por objetivo apresentar os códigos dos exercícios do primeiro trabalho de avaliação da disciplina de LP3, detalhando seu processo de solução e o que foi aprendido durante a realização do trabalho.

1. PROBLEMA 1

O problema começou com a criação de 3 classes principais: Navio, Cruzeiro e Cargueiro. Sendo “Navio”, a classe mãe e as demais uma extensão da classe “Navio”, herdando suas características.

Para cada uma das classes, foram adicionados os atributos solicitados no enunciado e os métodos construtores, acessadores (getters) e mutadores (setters), respectivamente. Além disso, à cada uma das classes foi adicionado um método “toString”, que retorna os valores settados para os atributos definidos através dos métodos getters, onde nas Classes filhas “Cruzeiro” e “Cargueiro”, esse método sobrescreve o “toString” da classe mãe, exibindo apenas o que foi solicitado em cada uma delas.

No final da resolução do problema, foi criada uma Classe de teste contendo uma ArrayList de elementos do tipo “Navio”, onde foram adicionados objetos do tipo “Navio”, “Cruzeiro” e “Cargueiro”. Nessa Classe, foi construído um for que percorre todos os elementos da lista, imprimindo com o “System.out.println” os métodos “toString” de cada elemento.

* Código da Classe Navio:

package trabalho1;

public class Navio {

public String nomeNavio;

private String anoNavio; //Ano de construção do navio

public Navio () {

}

public void setNomeNavio (String nome) {

this.nomeNavio = nome;

}

public String getNomeNavio () {

return this.nomeNavio;

}

public void setAnoNavio (String ano) {

this.anoNavio = ano;

}

public String getAnoNavio () {

return this.anoNavio;

}

public String toString () {

return ("Nome do navio: " + this.getNomeNavio() + " Ano de construção: " + this.getAnoNavio());

}

}

* Código da Classe Cruzeiro:

package trabalho1;

public class Cruzeiro extends Navio {

private int maxPassageiros;

public Cruzeiro () {

}

public void setMaxPassageiros (int max) {

this.maxPassageiros = max;

}

public int getMaxPassageiros () {

return this.maxPassageiros;

}

@Override

public String toString () {

return "Nome do navio: " + getNomeNavio() + " Número máximo de passageiros: " + getMaxPassageiros();

}

}

* Código da Classe Cargueiro:

package trabalho1;

public class Cargueiro extends Navio {

private int capacidadeCarga;

public Cargueiro () {

}

public void setCapacidadeCarga (int toneladas) {

this.capacidadeCarga = toneladas;

}

public int getCapacidadeCarga () {

return this.capacidadeCarga;

}

@Override

public String toString () {

return "Nome do navio: " + getNomeNavio() + " Capacidade máxima de carga: " + getCapacidadeCarga();

}

}

* Código da Classe TesteNavio:

package trabalho1;

import java.util.ArrayList;

import java.util.List;

public class NavioTeste {

public static void main(String[] args) {

List<Navio> dadosNavio;

dadosNavio = new ArrayList<Navio>();

Navio navio = new Navio();

Cruzeiro cruzeiro = new Cruzeiro();

Cargueiro cargueiro = new Cargueiro();

navio.setNomeNavio("SS America");

navio.setAnoNavio("1938");

dadosNavio.add(navio);

cruzeiro.setNomeNavio("Titanic");

cruzeiro.setAnoNavio("1909");

cruzeiro.setMaxPassageiros(2400);

dadosNavio.add(cruzeiro);

cargueiro.setNomeNavio("Dimitrios");

cargueiro.setAnoNavio("1950");

cargueiro.setCapacidadeCarga(5000);

dadosNavio.add(cargueiro);

for (int i = 0; i < dadosNavio.size(); i++) {

System.out.println(dadosNavio.get(i).toString());

}

}

}

1. PROBLEMA 2
2. PROBLEMA 3

Após identificar no código de teste que foi apresentado as variáveis que estavam sendo manipuladas, foi criada uma Classe “Livro”, contendo os atributos “Titulo” e “Autor” protegidos por encapsulamento e os seus respectivos métodos acessadores (getters) e mutadores (setters).

* Código da Classe Livro:

public class Livro {

private String titulo;

private String autor;

public String getTitulo() {

return titulo;

}

public void setTitulo(String titulo) {

this.titulo = titulo;

}

public String getAutor() {

return autor;

}

public void setAutor(String autor) {

this.autor = autor;

}

}

1. PROBLEMA 4
2. Na solução deste problema, foi criada uma Classe “Retangulo” com as variáveis de instância “comprimento” e “largura” protegidas por encapsulamento. Em seguida, foram criados os métodos assessores (getters) e mutadores (setters) de cada uma delas e mais dois métodos foram criados: O método “getPerimetro”, que calcula o perímetro do retângulo através do comprimento e da largura e o método “getArea”, que calcula a área do retângulo. Para isso, todos os valores do objeto foram definidos como tipo “double”, para permitir o cálculo preciso utilizando números decimais.

* Código da Classe Retangulo:

public class Retangulo {

protected double comprimento;

protected double largura;

public double getComprimento() {

return comprimento;

}

public void setComprimento(double comprimento) {

this.comprimento = comprimento;

}

public double getLargura() {

return largura;

}

public void setLargura(double largura) {

this.largura = largura;

}

public double getPerimetro() {

double perimetro = (2 \* largura) + (2 \* comprimento);

return perimetro;

}

public double getArea() {

double area = largura \* comprimento;

return area;

}

}

1. Na segunda parte do problema, foi criada uma Classe de teste “TesteRetangulo” que instancia dois objetos: ret1 e ret2 (representando 2 retângulos). Após settar o comprimento e a largura dos dois retângulos, é utilizado o “System.out.printf” para imprimir os resultados do cálculo do perímetro e da área feitos pelos métodos “getPerimetro” e “getArea” de cada um dos objetos.

* Código da Classe TesteRetangulo:

public class TesteReatangulo {

public static void main(String[] args) {

Retangulo ret1 = new Retangulo();

Retangulo ret2 = new Retangulo();

ret1.setComprimento(4);

ret1.setLargura(2);

ret2.setComprimento(7.3);

ret2.setLargura(3.5);

System.out.printf("\n1°Perim.: %.2f | 1°Area: %.2f",

ret1.getPerimetro(), ret1.getArea());

System.out.printf("\n2°Perim.: %.2f | 2°Area: %.2f",

ret2.getPerimetro(), ret2.getArea());

}

}

1. CONCLUSÃO

Ao final desse trabalho, pudemos aprender mais sobre os conceitos de Herança e Polimorfismo, criação de listas dinâmicas (ArrayList) e Encapsulamento através dos métodos getters e setters. Tanto na parte teórica como na parte prática que foi demonstrada através da construção e análise dos códigos exigidos.