Faculdade de Tecnologia de São Paulo – FATEC – SP ILP 007 - POO

Análise e Desenvolvimento de Sistemas – Noturno Prof.ª Grace

**Avaliação 02 – Em grupo de 2 ou 3 alunos - Entrega até 31/05**

Aluno 1: Fabiano de Oliveira Preto Junior | RA: 20219495

Aluno 2: Wander David Vicente | RA: 19100495

Aluno 3: Leonardo Rocha da Silva | RA: 17111515

1. (1.0) Responda as seguintes questões quanto a modelagem de sistemas:

a. O que é Modelagem de Sistemas? Por que é importante?

RESPOSTA:

A Modelagem de Sistemas é o processo de representar em modelos abstratos um determinado sistema. Cada modelo apresenta uma visão do sistema sob análise, de modo a facilitar a compreensão por diferentes equipes trabalhando sobre o sistema. Atualmente, o modo mais utilizado para criar as representações dos sistemas é a UML (Unified Modeling Language), que utiliza símbolos gráficos para apresentar as interações e relações entre as diferentes partes do sistema.

b. O que é UML? Cite pelo menos um exemplo de cada um dos seguintes tipos de diagrama: Estrutura, Comportamento e Interação da UML.

RESPOSTA:

Segundo o consórcio Object Management Group (OMG), a UML “é uma linguagem para especificação, construção, visualização e documentação de artefatos de um sistema de software intensivo”. É uma linguagem gráfica para análise, especificação e construção de sistemas para representar projetos orientados a objetos utilizando uma notação comum.

A UML garante a liberdade para o desenvolvedor adotar qualquer processo, metodologia ou linguagem de programação sem deixar de expressar-se claramente para os seus usuários e outros desenvolvedores, pois utiliza uma notação padrão, comum a todos os ambientes e empresas.

* São Diagramas de Estrutura: Diagrama de Classes.
* Diagramas de Comportamento: Diagrama de Casos de Uso.
* Diagramas de Interação: Diagrama de Sequência.

2. (1.0) Declare (e instancie) vetores em Java preparados para armazenar:

a. Rendimento mensal de uma aplicação financeira ao longo de um ano.

RESPOSTA:

float[] rendimentoMensal = new float[12];

b. As notas da prova escrita e prova prática dos 50 candidatos de um concurso.

RESPOSTA:

Notas[] arrNotas = new Notas[50];

3. (2.0) Defina os seguintes conceitos:

a. Sobrecarga

RESPOSTA:

A sobrecarga é um conceito do polimorfismo que consiste em criar variações de um mesmo método (ou seja, a criação de dois ou mais métodos com nomes totalmente iguais). A sobrecarga permite que seja utilizado o mesmo nome em mais de um método contanto que suas listas de argumentos sejam diferentes para que seja feita a distinção deles.

b. Polimorfismo

RESPOSTA:

De forma genérica, polimorfismo significa “várias formas”. No caso da Orientação a Objetos, polimorfismo denota uma situação na qual um objeto pode se comportar de maneiras diferentes dependendo do seu tipo de criação.

O polimorfismo permite que novas classes sejam adicionadas sem a necessidade de modificação de código já existente, pois cada classe apenas define os seus métodos e atributos.

c. Herança

RESPOSTA:

A herança é um tipo de relacionamento que define que uma classe “é uma” de outra classe. Isso possibilita que as implementações realizadas na classe origem sejam “herdadas” pela nova implementação e, com isso, as características, métodos e constantes também sejam repassadas, como na genética de pai para filho.

d. Classe abstrata

RESPOSTA:

Pode-se dizer que as classes abstratas servem como “modelo” para outras classes que dela herdem, não podendo ser instanciada diretamente. Para ter um objeto de uma classe abstrata é necessário criar uma classe mais especializada herdando dela e então instanciar essa nova classe.

4. (2.0) Faça um programa que leia e armazene o salário de uma amostra de 15 trabalhadores. Calcule a média salarial e indique a estatística de: quantas pessoas estão abaixo dessa média, quantas estão acima e quantas estão na média.

RESPOSTA:

- Classe de Teste

import java.util.Scanner;

public class App {

public static void main(String[] args) throws Exception {

final int tamArray = 15;

double[] arrSalarios = new double[tamArray];

double mediaSalarios;

Scanner inputDouble = new Scanner(System.in);

System.out.println("<===== Calculadora de salários =====>\n\n");

System.out.println("Por favor digite: \n");

for(int i = 0; i < arrSalarios.length; i++)

{

System.out.print("Valor do " + (i + 1) + "º salário: ");

arrSalarios[i] = inputDouble.nextDouble();

}

mediaSalarios = Salario.calcularMediaSalarios(tamArray, arrSalarios);

System.out.printf("\n\nMédia salárial dos trabalhadores: R$ %.2f", mediaSalarios);

System.out.printf("\nNº trabalhadores com salário acima da média: %d", Salario.calcularAcimaDaMedia(mediaSalarios, arrSalarios));

System.out.printf("\nNº trabalhadores com salário abaixo da média: %d", Salario.calcularAbaixoDaMedia(mediaSalarios, arrSalarios));

System.out.printf("\nNº trabalhadores com salário na média: %d\n\n", Salario.calcularNaMedia(mediaSalarios, arrSalarios));

inputDouble.close();

}

}

- Classe auxiliar static de Salario

public class Salario {

public static double calcularMediaSalarios(int numTrabalhadores, double[] arrSalarios)

{

double total = 0;

for (double salario : arrSalarios)

{

total = total + salario;

}

return total / numTrabalhadores;

}

public static int calcularAcimaDaMedia(double media, double[] arrSalarios)

{

int cont = 0;

for (double salario : arrSalarios) {

if(salario > media)

cont++;

}

return cont;

}

public static int calcularAbaixoDaMedia(double media, double[] arrSalarios)

{

int cont = 0;

for (double salario : arrSalarios) {

if(salario < media)

cont++;

}

return cont;

}

public static int calcularNaMedia(double media, double[] arrSalarios)

{

int cont = 0;

for (double salario : arrSalarios) {

if(salario == media)

cont++;

}

return cont;

}

}

5. (4.0) Implemente o seguinte conjunto de classes usando herança. Codifique um programa que leia os dados de 5 funcionários, armazene em vetor e exiba seus dados usando polimorfismo. Observe que a classe funcionário é abstrata e o pagamento do funcionário é calculado nas subclasses concretas: Programador (possui salário mensal) e Instrutor (recebe de acordo com as horas trabalhadas).

*Funcionario*

#nome: String

#mail: String

<<construtor>> Funcionario (nome, mail)

+ setNome( nome: String)

+ getNome( ): String

+ setMail( mail: String)

+ getMail( ): String

*+ pagamento( ): float*

+ exibeDados( )

Instrutor

- disciplina: String

- horasMensais: integer

- valorHoraAula: float

<<construtor>> Instrutor (nome, mail, disciplina)

+ setDisciplina(disc: String)

+ getDisciplina( ): String

+ setHorasMensais( horas: integer)

+ setValorHoraAula( vlr: float)

+ pagamento( ): float

+ exibeDados( )

Programador

- linguagem: String

- salarioMensal: float

<<construtor>> Programador (nome, mail, linguagem)

+ setlinguagem(ling: String)

+ getLinguagem( ): String

+ setSalario( salario: float)

+ pagamento( ): float

+ exibeDados( )

RESPOSTA:

- Classe de Teste

import java.util.Scanner;

public class App {

public static void main(String args[]) {

final int FUNCIONARIOS = 5;

Scanner floatInput = new Scanner(System.in);

Scanner stringInput = new Scanner(System.in);

Scanner intInput = new Scanner(System.in);

Funcionario array[] = new Funcionario[FUNCIONARIOS];

String nome, mail, linguagem, disciplina;

float salario, valorHoraAula;

int tipo, horasMensais;

for (int i = 0; i < FUNCIONARIOS; i++)

{

System.out.printf("\n\nDigite o nome do funcionario: ");

nome = stringInput.nextLine();

System.out.printf("Digite o mail do funcionario: ");

mail = stringInput.nextLine();

System.out.printf("Digite o tipo do funcionario - (1) programador; (2) instrutor: ");

tipo = intInput.nextInt();

if (tipo == 1)

{

System.out.printf("Digite a linguagem do programador: ");

linguagem = stringInput.nextLine();

System.out.printf("Digite o salario do programador: ");

salario = floatInput.nextFloat();

Programador funcionario = new Programador(nome, mail, linguagem);

funcionario.setSalario(salario);

array[i] = funcionario;

}

else if (tipo == 2)

{

System.out.printf("Digite a disciplina do instrutor: ");

disciplina = stringInput.nextLine();

System.out.printf("Digite o valor da hora aula do instrutor: ");

valorHoraAula = floatInput.nextFloat();

System.out.printf("Digite a quantidade de horas mensais do instrutor: ");

horasMensais = intInput.nextInt();

Instrutor funcionario = new Instrutor(nome, mail, disciplina);

funcionario.setValorHoraAula(valorHoraAula);

funcionario.setHorasMensais(horasMensais);

array[i] = funcionario;

}

}

for (int i = 0; i < FUNCIONARIOS; i++)

{

System.out.printf("\n\nFuncionario %d:\n", i + 1); array[i].exibeDados();

}

floatInput.close();

stringInput.close();

intInput.close();

}

}

- Classe Funcionario

public abstract class Funcionario

{

private String nome, mail;

public Funcionario(String nome, String mail)

{

setNome(nome);

setMail(mail);

}

public void setNome(String nome)

{

this.nome = nome;

}

public String getNome()

{

return nome;

}

public void setMail(String mail)

{

this.mail = mail;

}

public String getMail()

{

return mail;

}

public void exibeDados()

{

System.out.printf("Nome: " + nome + "\n"); System.out.printf("Mail: " + mail + "\n");

}

public abstract float pagamento();

}

- Classe Programador

public class Programador extends Funcionario

{

private String linguagem;

private float salarioMensal;

public Programador(String nome, String mail, String linguagem)

{

super(nome, mail);

setLinguagem(linguagem);

}

public void setLinguagem(String linguagem)

{

this.linguagem = linguagem;

}

public String getLinguagem()

{

return linguagem;

}

public void setSalario(float salario)

{

salarioMensal = salario;

}

public float pagamento()

{

return salarioMensal;

}

public void exibeDados()

{

super.exibeDados();

System.out.printf("Linguagem: " + linguagem + "\n"); System.out.printf("Pagamento: R$ %.2f\n", pagamento());

}

}

- Classe Instrutor

public class Instrutor extends Funcionario

{

private String disciplina;

private int horasMensais; private float valorHoraAula;

public Instrutor(String nome, String mail, String disciplina)

{

super(nome, mail);

setDisciplina(disciplina);

}

public void setDisciplina(String disciplina)

{

this.disciplina = disciplina;

}

public String getDisciplina()

{

return disciplina;

}

public void setHorasMensais(int horas)

{

horasMensais = horas;

}

public void setValorHoraAula(float vlr)

{

valorHoraAula = vlr;

}

public float pagamento()

{

return valorHoraAula \* horasMensais;

}

public void exibeDados()

{

super.exibeDados();

System.out.printf("Disciplina: " + disciplina + "\n"); System.out.printf("Pagamento: R$ %.2f\n", pagamento());

}

}

*Boa prova!*