Disciplina: Programação Orientada a Objetos

Profa Dra: Marcela Alves de Souza

# Tarefa 1

***Informações Importantes:***

* Anexar o arquivo .zip no SIGAA até a data estabelecida no sigaa
* Não será possível reabrir o SIGAA para reenvio de atividades, então, fiquem atentos aos prazos.

Questão 1) Analise os códigos abaixo e compare-os:

**Exemplo1 de Código Procedural em C: Função para cadastrar um usuário e imprimir seu nome e email**

#include <stdio.h>

void cadastrarUsuario(char nome[], char email[]) { printf("Cadastrando usuário: %s\n", nome); printf("Email: %s\n", email);

}

int main() { char nome[] = "João"; char email[] = "joao@example.com";

cadastrarUsuario(nome, email); return 0;

}

**Exemplo2 de Código Procedural em Java: Cadastro de Usuário e imprimir seu nome e email**

public class CadastroUsuario { public static void main(String[] args) {

String nome = "João";

String email = "joao@example.com"; cadastrarUsuario(nome, email);

}

public static void cadastrarUsuario(String nome, String email) {

System.out.println("Cadastrando usuário: " + nome);

Disciplina: Programação Orientada a Objetos

Profa Dra: Marcela Alves de Souza

System.out.println("Email: " + email);

}

}

**Exemplo3 de Código Orientado a Objetos em Java: Cadastro de Usuário e imprimir seu nome e email**

public class Usuario { private String nome; private String email;

public Usuario(String nome, String email) { this.nome = nome; this.email = email;

}

public void cadastrar() {

System.out.println("Cadastrando usuário: " + nome);

System.out.println("Email: " + email);

}

}

public class CadastroUsuario { public static void main(String[] args) {

Usuario usuario = new Usuario("João", "joao@example.com"); usuario.cadastrar();

}

}

Responda às seguintes perguntas:

1. Quais são as principais diferenças entre os 3 códigos (procedural e POO)?
2. Como a POO melhora a organização e a escalabilidade do sistema?
3. Quais são os benefícios de usar objetos no lugar de funções simples?

Questão 2) Modifique o código POO Exemplo3 para incluir um segundo usuário, criando um método para alterar o e-mail de um usuário.

Questão 3)Explique os tipos de dados primitivos em Java. Dê exemplos de como utilizar cada um desses tipos no código, e descreva as diferenças entre tipos de dados inteiros, Disciplina: Programação Orientada a Objetos

Profa Dra: Marcela Alves de Souza

ponto flutuante, booleanos e caracteres. Além disso, explique como a conversão entre tipos de dados (casting) pode ser realizada em Java, dando um exemplo prático.

Questão 4) Explique o funcionamento das estruturas condicionais if, else if e else em Java. Em seguida, crie um exemplo de código onde um programa decide, com base na idade do usuário, se ele pode votar ou não. A idade mínima para votar é 16 anos, mas o voto é obrigatório apenas para maiores de 18 anos.

Questão 5) Discuta os diferentes tipos de laços de repetição em Java: for, while e do-while. Em seguida, escreva um programa que, utilizando um laço, exiba os números de 1 a 100 que são divisíveis por 3. Explique qual seria a diferença caso você usasse um laço for em vez de um while para resolver este problema.

Questão 6) Explique como o tipo String funciona em Java. Discuta suas características, como imutabilidade, e as principais operações que podem ser realizadas com strings, como concatenar e comparar. Escreva um código que leia uma palavra do usuário e verifique se ela contém a letra "a" (em qualquer posição). Se a letra "a" for encontrada, o programa deve exibir "A letra 'a' foi encontrada!" e, caso contrário, "A letra 'a' não foi encontrada!".

Respostas

Questão 1) a) No primeiro exemplo feito em C é utilizada uma função para cadastrar, já no segundo e terceiro exemplo, feito em java, nota-se que foram feitas utilizações de métodos para cadastrar, com foco no terceiro que utilizou um objeto também.

b) Com POO os métodos ficam delimitados de uma forma melhor, facilitando a identificação e cada projeto.

c) Melhor organização, o projeto fica delimitado de uma maneira mais clara.

Questão 3) Os dados primitivos em java em sua maioria são numéricos como, int, float, long, com exceção do boolean que só admite “true” ou “false”, e a variável char que admite 1 caractere. Dados inteiros só armazenam valores inteiros; ponto flutuante são para armazenar valores decimais; booleanos só recebem verdadeiro ou falso; e o de caracteres como o nome já diz, só recebe caractere.

A conversão de dados é basicamente você pegar um dado por exemplo 10,5 ,que é do tipo float, e você converte ele para 10 , que é do tipo int. As vezes, um valor pode conter muitas casas decimais e você não quer que apareçam todas, você pode simplesmente converter esse dado para o tipo int.

Questão 4) Normalmente essas estruturas são postas de forma sequencial dentro do código, começando com o if, condição primária, se a primeira condição não tiver acontecido, parte para o else if, que não possue um limite de quantos você pode colocar, se após todas as condições que você colocou não aconterem, você utiliza o else. Podem ser feitas uma infinidade de ações dentro dessas estruturas quando suas condições são atendidas.

Questão 5) Basicamente, no laço for, você precisará de uma variável auxiliar para servir como um contador e também como um comparativo, será posto uma condição para esse contador no início do laço e até que essa condição seja atingida, o contador será incrementado; por outro lado, o while e o do-while só necessitam de uma variável para condição, a diferença é que, no while a condição é no começo do laço, e no do-while é no final, ou seja, o do while roda primeiro, depois ele faz a verificação.

Na questão eu optei por utilizar o for, já que eu iria necessitar de uma variável para ficar encrementando e o for já ajuda com isso, se fosse com while eu teria que encrementar a variável dentro do laço.

Questão 6) Podemos dizer que o tipo String é uma forma aprimorada do tipo char, enquanto o char só recebe 1 caractere, na String não há esse limite, você pode digitar uma palavra ou até uma frase, ela se torna muito flexível por esse fator além de que, voce ainda pode realizar operações com ela, como por exemplo: concatenar uma palavra com outra, contar quantos cacteres foram inseridos na string, e até procurar um caractere específico.