



FACULDADE DE CIÊNCIAS E TECNOLOGIA
CURSO DE ENGENHARIA INFORMÁTICA 2º ANO – POO I – 2024
Exercícios de consolidação ficha nº7

1. Crie uma classe e demonstre técnicas de encapsulamento adequadas.
 - a. a classe se chamará printer
 - b. Simulará uma impressora de computador real
 - c. Deve conter campos para Nível de toner, número de páginas impressas e
 - d. também se é uma impressora duplex (capaz de imprimir nos dois lados do papel).
 - e. Adiciona métodos para preencher o toner (até no máximo 100%), outro método para
 - f. simula a impressão de uma página (o que deve aumentar o número de páginas impressas)
 - g. Decida o escopo, se usará construtores e qualquer outra coisa que você achar necessária.

```
Printer printer = new Printer(50, true);
System.out.println("Initial page count = " + printer.getPrintedPages());
int pagesPrinted = printer.print(4);
System.out.println("Pages printed was " + pagesPrinted + ". New total print count = "
+ printer.getPrintedPages());
pagesPrinted = printer.print(2);
System.out.println("Pages printed was " + pagesPrinted + ". New total print count = "
+ printer.getPrintedPages());
```

Métodos esperados.

2. Requisitos:

Classe de Alunos:

A classe Student deve ter variáveis de instância privadas para armazenar informações do aluno, como nome, ID, idade e série.

Use modificadores de acesso apropriados e forneça métodos getter e setter públicos para acessar e atualizar as informações dos alunos.

Aula de Gestão Estudantil:

A classe Student Management deve ter variáveis estáticas privadas para armazenar uma lista de alunos e o número total de alunos.

Use modificadores de acesso apropriados para controlar o acesso às variáveis.

Implemente métodos estáticos para adicionar novos alunos, atualizar informações dos alunos e recuperar detalhes dos alunos.

O método addStudent deve aceitar parâmetros para informações do aluno e criar um novo objeto Student. Deverá adicionar o aluno à lista de alunos e atualizar o número total de alunos.

O método updateStudent deve aceitar uma carteira de estudante e novas informações para o aluno. Ele deve localizar o aluno na lista e atualizar as informações do aluno de acordo.

O método getStudentDetails deve aceitar uma carteira de estudante e retornar os detalhes do aluno correspondente.

Interface do administrador:

Desenvolva uma interface de linha de comando interativa para os administradores interagirem com o Sistema de gerenciamento de registros de alunos.

Exiba um menu com opções para adicionar um novo aluno, atualizar informações do aluno e visualizar detalhes do aluno.

Solicite ao administrador as entradas necessárias e chame os métodos apropriados na classe StudentManagement para executar as operações solicitadas.

Implemente o tratamento de erros para lidar com casos em que a carteira de estudante não é encontrada ou entradas inválidas são fornecidas.

3. Crie uma classe "Room" que conterá a "altura", "largura" e "largura" da sala em três campos. Esta classe também possui um método "volume()" para calcular o volume desta sala. Crie outra classe "RoomDemo" que usará a classe anterior, criará instâncias de salas e exibirá o volume da sala.
4. Escreva um programa para implementar uma classe "aluno" com os seguintes membros. Nome do aluno. Notas do aluno obtidas em três disciplinas. Função para atribuir valores iniciais. Função para calcular a média total. Função para exibir o nome do aluno e o total de notas. Escreva uma função main() apropriada para demonstrar o funcionamento do acima.
5. Escreva uma classe "Caixa" com três variáveis-membro "altura", "largura" e "largura". Escreva construtores adequados para inicializá-los. Adicione funções como "getVolume" e "getArea" que retornarão volume e área de superfície respectivamente. Instancie duas caixas arbitrárias e imprima seu volume e área de superfície.
6. Implemente uma classe para pilha de inteiros usando um array. Observe que as operações definidas para uma estrutura de dados de pilha são as seguintes: "push", "pop", "print". Deveria haver um construtor para criar um array de inteiros; o tamanho da matriz é fornecido pelo usuário. Escreva uma função main() para (i) criar uma pilha para armazenar no máximo 30 inteiros; (ii) colocar os números 10, 20, 30, 15, 9 na pilha; (iii) imprimir a pilha; (iii) pop três vezes e (iv) imprima a pilha novamente.
7. Escreva uma classe "BankAccount" com as seguintes variáveis de instância: AccountNumber (um número inteiro), balance um número de ponto flutuante) e "ownerName" (uma String). Escreva o construtor adequado para esta classe. Escreva também os métodos balance, add (para depositar uma quantia) e subtract (para sacar uma quantia) e implemente-os. Agora crie outra classe "AccountManager" que contém um array de BankAccount. Métodos de gravação criar (para criar uma conta), excluir (para encerrar uma conta), depositar (para depositar uma quantia em uma conta) e retirar (para sacar uma quantia de uma conta). Escreva também uma classe "Banco", adicione a função main() que cria um AccountManager e adicione 5 contas. Agora imprima os dados de todas as contas deste Banco.
8. Escreva uma classe para representar números complexos com os construtores necessários. Escreva funções-membro para somar e multiplicar dois números complexos. Deverão existir três construtores: (i) para inicializar as partes reais e imaginárias com 0; (ii) inicializar a parte imaginária com 0, mas inicializar a parte real com o valor definido pelo usuário; (iii) inicializar a parte real e a parte imaginária com valores definidos pelo usuário. Além disso, escreva uma função main() para (i) criar dois números complexos $3+2i$ e $4-2i$; (ii) imprimir a soma e o produto desses números.
9. Implemente uma classe para "Data". Escreva funções-membro para (i) obter o dia anterior, (iv) obter o dia seguinte, (iii) imprimir um dia. Deve haver quatro construtores:

(i) dia, mês e ano são inicializados em 01, 01, 1970; (ii) o dia é inicializado com o valor especificado pelo usuário, mas o mês e o ano são inicializados com 01 de 1970; (iii) dia e mês são inicializados com o valor especificado pelo usuário, mas o ano é inicializado em 1970; (iv) dia, mês e ano são inicializados com valores definidos pelo usuário. Além disso, escreva uma função `main()` para (i) criar um objeto de data; (ii) imprimir no dia seguinte e no dia anterior.

10. Implemente uma classe para um “Livro”. O livro contém um título (uma String), uma lista de autores (matriz de autores), número de páginas (um número inteiro), preço (número de ponto flutuante), editora (uma String) etc. Escreva métodos construtores e acessadores/modificadores adequados. Implemente uma classe para “Biblioteca”. Uma biblioteca contém uma lista de livros (matriz de Livro). Escreva os métodos `add` (para adicionar um livro) e `remove` (para excluir um livro) para a biblioteca. Escreva uma função `main()` para criar uma “Biblioteca” e adicionar cinco “Livros” à biblioteca. Imprima o preço total de todos os livros.
11. Escreva uma classe Java “Employee” contendo informações de nome, id, endereço, salário, etc. Escreva o construtor necessário e os métodos de leitura/gravação. Crie uma classe “Depto” que tenha nome, localização, etc. O “Depto” contém um número de “Funcionário”. Escreva os métodos “`add`” e “`remove`” para adicionar e remover um funcionário deste departamento. Escreva uma função `main()` e crie o departamento de “Tecnologia da Informação”. Adicione cinco funcionários e imprima as despesas anuais deste departamento.
12. Implemente uma aula para um “Aluno”. As informações sobre um aluno incluem nome, número da matrícula e uma série de cinco nomes de disciplinas. A classe deve ter construtor e métodos `get/set` adequados. Implemente uma classe “TabulationSheet”. Uma folha de tabulação contém números de rolos e notas de cada aluno para uma matéria específica. Esta aula deve ter um método para somar as notas e números de um aluno. Implemente uma classe “MarkSheet”. Uma folha de notas contém notas de todas as disciplinas de um determinado aluno. Esta aula deve ter um método para adicionar o nome de um aluno e notas em cada disciplina. Escreva uma função `main()` para criar três objetos “Student”, cinco objetos “Tabulationsheet” para cinco disciplinas e três objetos “Marksheet” para três estudantes.

