



Tecnologia em Análise e Desenvolvimento de Sistemas

Programação Orientada a Objetos Caderno de Exercícios Avaliativos



1º/2018





Exercício 1

- Pesquise sobre Coleta de Lixo em Java e o método finalize().
- Elabore um texto explicando o funcionamento, frequência, vantagens, desvantagens, fazendo um comparativo com outras linguagens de programação (Ex. C++). O texto deve explicar a utilização do método finalize() e um exemplo simples de sua utilização.
- Gerar arquivo em formato PDF.



Exercício 2

- Implemente um sistema que simule um jogo de dados, o qual o jogador lança 2 dados de seis lados.
- Deve-se construir uma classe Dado que representa um dado.
- O programa principal deve lançar os dados 3 vezes e apresentar a soma dos números dos dados a cada um dos lances.
- **Material de pesquisa:**
 - Números aleatórios em java:
<https://www.devmedia.com.br/numeros-aleatorios-em-java-a-classe-java-util-random/26355>



Exercício 3

- Crie uma classe **USMoney** com dois atributos inteiros: **dollars** e **cents**.
- Adicione um construtor com dois parâmetros para a inicialização do objeto USMoney. O construtor deve verificar se o valor de cents está entre 0 e 99 e, se não estiver, transferir alguns dos cents para o atributo dollars para que ela passe a ter entre 0 e 99.
- Implemente um método **plus** que recebe um objeto USMoney como argumento. Esse método deve criar e retornar um novo objeto USMoney representando a soma dos objetos cujo método plus() está sendo chamado mais o argumento, sem modificar os valores dos dois objetos já existentes.
- Deve-se assegurar que o valor do atributo cents do novo objeto esteja entre 0 e 99. Por exemplo, se x for um objeto USMoney com 5 dollars e 80 cents e se y for um objeto USMoney de 1 dollar e 90 cents, **x.plus(y)** retornará um novo objeto USMoney com 7 dollars e 70 cents.
- Implemente um programa principal que teste vários casos de teste para o método plus da classe USMoney.



Exercício 4

- O sistema de controle de fila de um banco opera da seguinte forma. O cliente que chega a agência retira uma senha que pode ter prioridade preferencial (para gestantes, portadores de necessidades especiais e maiores de 65 anos) ou não preferencial. A agência dispõe de um caixa preferencial e vários caixas não preferenciais. Ao decidir qual o próximo cliente a ser atendido, são consideradas duas políticas:
 - O caixa preferencial seleciona o primeiro cliente preferencial da fila. Não havendo clientes preferenciais, é selecionado o primeiro cliente da fila;
 - O caixa não preferencial respeita a ordem de chegada e seleciona o primeiro cliente que está aguardando na fila, preferencial ou não.
- Para armazenar as informações da fila, utiliza-se um array de clientes, em que cada posição possui um objeto com os atributos e métodos a seguir:
 - Senha – número sequencial gerado somando 1 da senha gerada anteriormente;
 - Prioritario – booleano que indica se o cliente é prioritário ou não.
- Nessa classe, o campo senha armazena o número da senha atribuída a um cliente (o número da senha é sequencial), o campo prioridade é um booleano que pode ter os valores true ou false, indicando que o cliente é preferencial(true) ou não preferencial (false).
- Implemente um sistema orientado a objetos que gerenciará a fila de atendimento do banco.



Exercício 5

- Laboratório 1



Exercício 6

- Laboratório 2



Exercício 7

- Foi disponibilizado o pacote JogoAdivinha.jar e sua documentação.
 - <https://www.dropbox.com/s/gmhnc9quqv51sdz/EAvaliativo7.zip.zip?dl=0>
- O jogo consiste em sortear um número inteiro e depois ler vários palpites do jogador, até que o mesmo desista ou acerte o valor sorteado.
- Implemente:
 - Um programa principal que utilize a classe JogoAdivinha.
 - Estenda a classe JogoAdivinha para que:
 - Armazene todos os palpites dados em ordem crescente;
 - Limite o número de palpites que o jogador tem direito;
 - Um programa principal que utilize a classe estendida e teste todo o sistema.



Exercício 8 (APS)

- **Implemente o jogo “Papel, Pedra e Tesoura”.**

- Deve ser implementada uma classe “Coisa” que será uma abstração (classe abstrata) de Papel, Pedra e Tesoura. A classe Jogador instanciará um objetos das classes especializadas de Coisa, que será “colocadas à prova”.
- Na classe Jogo deve implementar as seguintes possibilidades:
 - Dois jogadores automáticos: São instanciados pois jogados virtuais que escolhem qual “Coisa” irão instanciar.
 - Humano x Computador: Instanciado um jogador humana, esse seleciona qual Coisa instanciar e cria-se um jogador virtual e este, aleatoriamente, define qual Coisa é instanciada.
 - Humano x Computadores: Existe um jogador humano e N (usuário define quantos) jogadores virtuais.
 - Permitir os vários “confrontos”. Pode-se escolher confronto unitário ou melhor de 3 confrontos. O sistema apresenta o nome do jogador vencedor.

- **Referências:**

- https://pt.wikipedia.org/wiki/Pedra,_papel_e_tesoura
- <http://www.aprendendojapones.com/2008/06/07/jankenpon/>

- **Conceitos obrigatórios:**

- Orientação a objetos, encapsulamento, métodos de acesso, herança, polimorfismo, sobrecarga, documentação em Javadoc.

- **Entrega:**

- A entrega da atividade deve ser feita pelo GitHub. No moodle o aluno deverá enviar o link do projeto. O último commit deve ser executado até a data limite definida no moodle, caso haja commit posterior a atividade será desconsiderada.



Exercício 9 (APS)

- **Implemente um sistema orientado a objetos para gerenciamento de empréstimos de livros da biblioteca.**

- A biblioteca possui uma coleção de títulos de livros onde são armazenados nome dos autores, título do livro, subtítulo, editora e ISBN. Na biblioteca existem 1 ou mais exemplares de um Livro, os exemplares são únicos, possuem um identificador composto pelo ISBN acrescido em um número inteiro sequencial. Por fim, um exemplar pode ser ou não disponibilizado para empréstimo.
- Os usuários da biblioteca são professores, alunos e servidores. Sobre os usuários são armazenados: nome, prontuário e senha (só é possível o empréstimo após verificação da senha [senha deve ser armazenada em MD5]), além disso, para cada usuário é implementada uma política de empréstimo: professores 30 dias, alunos 7 dias e servidores 4 dias.
- O sistema deve permitir o cadastro de livros e usuários e o gerenciamento de empréstimos e devoluções. Atenção pois um mesmo exemplar não pode ser emprestado duas vezes ao mesmo tempo.
- Para facilitar a população dos dados, implemente em *hardcode* a criação de vários objetos, mas permitindo, de alguma forma acesso às funcionalidades do sistema.
- Gere relatório:
 - Alunos e livros emprestados.
 - Livros da biblioteca com a quantidade de exemplares disponíveis no momento.
 - Livros emprestados com a indicação do exemplar e aluno com o livro.

- **Referências**

- <http://codare.aurelio.net/2007/02/02/java-gerando-codigos-hash-md5-sha/>
- <https://desenvolvedores.migrate.com.br/2014/03/07/gerando-o-codigo-hash-no-formato-md5/>
- <https://www.youtube.com/watch?v=xKCCOovGxGw>

- **Conceitos obrigatórios:**

- Orientação a objetos, encapsulamento, métodos de acesso, herança, polimorfismo, sobrecarga, documentação em Javadoc.

- **Entrega:**

- A entrega da atividade deve ser feita pelo GitHub. No moodle o aluno deverá enviar o link do projeto. O último commit deve ser executado até a data limite definida no moodle, caso haja commit posterior a atividade será desconsiderada.