Segunda Lista de Exercícios

1. **Altere o programa de impressão de calendário que fizemos em sala de aula para que** **comece a semana no domingo e com as iniciais dos dias em inglês. Note que não basta** **apenas alterar o cabeçalho. (FEITO)**
2. **Considere o método nextInt da classe Scanner. Ele é um método de acesso ou** **modificador? Por quê? E o método nextInt da classe Random?**

R: O método nextint() da classe Scanner é um método modificador e de acesso, pois ele recebe um token de entrada (a que for escolhida pelo programador, seja uma string, arquivo ou entrada padrão teclado) e verifica se ela é ou não válida, ou seja, é ou não um dado do tipo Integer, se sim ela retorna o dado lido pela entrada, se não, ele retorna uma exceção que informa o motivo da leitura não ser válida. Já o método nextInt() da classe Random é um método de acesso, pelo fato de seu funcionamento ser semelhante ao de um getter.

1. **É possível ter um método modificador que retorne algo diferente de void? É possível ter um método de acesso que retorne void? Dê exemplos quando possível.**

R: É sim possível um método modificador que retorne algo diferente de zero, pelo fato de que se pode criar um método que modifique o estado de uma variável e então retorne essa mesma variável em seu atual estado após sua modificação ou até mesmo o resultado de uma operação aritmética que a utilizou após sua alteração de estado.

Ex:

**public** class Exemplo {

**private** int x;

**public** int exemplo(int x) {

this.x = x;

**return** x; //ou (return x + 5;)

}

}

O inverso é valido para um método de acesso, ou seja, um método de acesso deve sempre retornar algo diferente de **void**, caso contrário o seu acesso a uma variável que é do modificador de acesso **private** se torna desnecessário e irrelevante.

1. **É possível implementar um método Java que troque o conteúdo de duas variáveis int? E no caso de dois objetos Integer? Escreva um método para testar cada caso e explique os resultados.**

R: Em nenhum dos casos é possível, pois nesses casos em especifico os métodos vão receber somente os valores, mas se fossem métodos que modificam variáveis existentes na classe em que foi declarada, a troca seria possível.

**public** class Exemplo {

**public** void troca(int x, int y) {

**int** temp = x;

x = y;

y = temp;

}

}

1. **Implemente uma classe imutável chamada Point que descreva um ponto no plano. Forneça um construtor para criar um ponto com coordenadas específicas, um construtor sem argumentos para definir um ponto na origem e métodos getX() e getY para obter as coordenadas do ponto. Escreva também os métodos translate()(soma ao x, y) e scale()(multiplica ao x, y). O método translate deve mover o ponto por uma determinada distância na direção x e y. O método de scale() redimensiona ambas as coordenadas por um determinado fator. Implemente esses métodos para que eles retornem novos pontos com os resultados. Por exemplo, Point p = new Point(3, 4).translate(1, 3).scale(0.5), deve definir p como um ponto com coordenadas (2, 3.5). (FEITO)**
2. **Repita o exercício anterior, mas agora implemente os métodos translate() e scale() como modificadores. (FEITO)**
3. **Implemente a classe Carro que modela um carro viajando em linha reta, consumindo gasolina à medida que se move. Forneça um método que modele a movimentação do carro por um determinado número de quilómetros, um método para adicionar um determinado número de litros ao tanque de gasolina, um método para obter a distância corrente da origem e um método para obter o nível de combustível. Especifique a eficiência de combustível (em quilómetros/litro) no construtor. A classe Carro deve ser uma classe imutável? Por que ou por que não? No programa principal, crie um carro, abasteça, e o faça andar por 50 km. Imprima as informações relacionadas ao combustível a cada 10 km. (FEITO)**

R: Não é possível e nem viável a classe carro ser uma classe do imutável, devido á muitas de suas variáveis necessitarem de alteração constante, como por exemplo, a quantidade de combustível e a quilometragem andada.

1. **Implemente a classe Calendário com método estático para imprimir o calendário de um dado ano e mês. Use importes estáticos para as classes System e LocalDate. Lembre-se imports estáticos permitem usar métodos estáticos sem mencionar as classes. (FEITO)**
2. **Faça um programa para imprimir “hello world!” no arquivo HelloWorld.java. Declare a classe HelloWorld como sendo do pacote cap01.sec01. Coloque o arquivo em algum diretório, mas não em cap01/sec01. A partir desse diretório, execute javac HelloWorld.java. Verifique se um arquivo .class foi obtido e, em caso positivo, onde? Em seguida, execute o programa com o comando java HelloWorld. O que acontece? Por quê? (Dica: execute javap HelloWorld e estude a mensagem de warning). Finalmente, tente javac -d . HelloWorld.java. O que acontece? Explique.**

R: Ao executar “**javac HelloWord.java**”, foi criado um arquivo **.class** no mesmo diretório de HelloWorld.java. Quando é pedido para executar “**java HelloWorld**”, é informado que arquivo HelloWorld não foi encontrado, isso é devido ao fato de que ao adicionar “pakage cap01.sec01” no cabeçalho de seu algoritmo, é definido para a mesmo aquela localização de diretório, e a não ser que seja incluído no path do java, o programa não será executado, pois a JVM não irá o encontrar.

1. **Complemente a implementação da classe RedeSocial apresentada durante as aulas. Adicione um método à inner class Membro para possibilitar a inserção de um membro na lista de amigos baseado no nome. Para simplificar, considere que na rede social não existam duas pessoas com o mesmo nome. (FEITO)**
2. **Faça a implementação completa da classe Fatura vista em sala (uma classe aninhada Estática). Forneça um método que imprima a fatura e um programa de demonstração que constrói e imprime uma fatura de amostra. (FEITO)**
3. **Implemente a classe Fila, uma fila ilimitada de strings. Forneça os métodos adicionar(), que adiciona ao fim da fila, ​​e remover(), que remove do início da fila. Armazene os elementos como uma lista encadeada de objetos da classe No.**
4. **Refaça o exercício 12 implementando a classe No como uma classe aninhada. Nesse caso, a classe No deve ser estática ou não? Implemente um iterador, isto é, um objeto que produz referências para os elementos da fila, um de cada vez. Implemente a classe Iterador como uma classe aninhada com métodos next() (retorna uma referência para o próximo No) e hasNext() (retorna true se houver um próximo No e false caso contrário). Forneça um método iterador() da Classe Fila que gera um objeto Fila.Iterador. O Iterator deve ser estático ou não?**