

ADS – IFPB – Campus Monteiro
LISTA SEMANAL – PROGRAMAÇÃO II - PROGRAMAÇÃO ORIENTADA A OBJETOS
Prof. Cleyton Caetano de Souza

Observação: essa lista pode ser feita em dupla e, caso a dupla faça todas as questões, ela pode ser entregue impressa.

Semana 5

PROJETO MÁQUINA DE LAVAR

1. Vamos simular o funcionamento de uma máquina de lavar “inteligente”. Para isso, comece criando a classe Roupa, que possuirá os seguintes atributos: cor, tamanho e estado. **Todos os atributos da classe Roupa são do tipo Enum.** Use para a cor um Enum com os valores: branco e colorido. Use para o tamanho um Enum com os valores: P, M ou G. Além disso, use para o estado um Enum com os valores: Limpa ou Suja. Siga as convenções de nomenclatura e visibilidade vistos em sala de aula.

2. Crie a classe Máquina de Lavar. Adicione à Máquina De Lavar o método lavar, o qual recebe como entrada um array de roupas e retorna um valor inteiro. Esse método retorna o tempo necessário para que a lavagem das roupas do array ocorra. Dependendo do tamanho e do estado da roupa, uma roupa leva mais ou menos tempo para ser lavada (roupas pequenas acrescentam 3 minutos, roupas médias acrescentam 5 minutos e grandes acrescentam 8 minutos; roupas coloridas levam 2 minutos para serem levadas e roupas brancas 1 minuto; além disso, se o estado da roupa for Suja, a máquina leva 3 minutos adicionais para cada roupa suja). Entretanto, antes de realizar a lavagem, o método lavar checa se o array de roupas possui roupas brancas e coloridas misturadas. Se houver roupas brancas e coloridas misturadas, o método lavar retorna o valor -1, para indicar que a lavagem não foi bem sucedida, e nada mais acontece. Caso a lavagem ocorra, **o método lavar muda o estado de todas as roupas do array para “Limpa”** e retorna o tempo necessário para fazer a lavagem. **Sugestão: quando manipular os objetos do array, lembre-se de se assegurar que existe um objeto naquela posição. Uma posição vazia, sem objetos, terá o valor null “guardado” lá.**

3. Crie um programa para testar a máquina de lavar. O usuário vai começar informando a quantidade roupas que ele quer lavar. Em seguida, o programa deve fazer a leitura das informações de cada roupa. Exiba as informações de cada roupa, antes de realizar a lavagem. Depois, use o método lavar para informar ao usuário o tempo necessário para realizar a lavagem. Caso não seja possível realizar a lavagem, informe ao usuário que não é possível. Ao final, exiba as informações de cada roupa, após passar pelo método lavar.

PROJETO LIVRARIA

4. Crie a classe Livro, que possuirá como atributos: título (String), ano de publicação (inteiro), quantidade disponível (inteiro) e preço (float). Essa classe representará um Livro que estará disponível para venda numa livraria. O atributo “quantidade disponível” indicará a quantidade desse livro que está disponível para venda (um livro com quantidade disponível igual a zero estaria esgotado). Siga as convenções de nomenclatura e visibilidades estudadas em sala de aula.

5. Crie a classe Livraria. Essa classe servirá para organizar o estoque (livros disponíveis para venda) e o caixa da Livraria. Adicione à classe Livraria, como atributo, um array de Livro, com capacidade para armazenar 100 livros. Além disso, inclua também um atributo chamado “quantidade de livros cadastrados” (esse atributo guardará a quantidade de objetos que estão “guardados” no array – mesmo se a quantidade disponível de um livro for 10 exemplares, ele contará como um único objeto “guardado”). Por fim, a Livraria terá um atributo do tipo float chamado “saldo em caixa”, que guardará o total acumulado com as vendas da livraria (começa com o valor zero, assim como a quantidade de livros cadastrados). Siga as convenções de nomenclatura e visibilidade vistas em sala de aula, *entretanto, crie apenas os métodos gets para os atributos* (a única forma de mudar os valores dos atributos será através dos métodos da Livraria que você programará nas próximas questões).

6. Adicione à classe Livraria um método chamado “cadastrar livro”, que receberá como parâmetro de entrada um objeto do tipo Livro e retornará um valor booleano. O método “salva” o Livro recebido na próxima posição disponível do vetor de

Livros. **Entretanto, não deve ser permitido cadastrar Livros com títulos repetidos no vetor.** O método retorna verdadeiro, se o cadastro foi bem sucedido, ou falso, caso contrário. Após adicionar o Livro, lembre-se de atualizar o atributo "quantidade disponível".

7. Adicione à classe Livraria um método chamado "comprar livro", que receberá uma String, que representar o título do Livro, como parâmetro de entrada e retornará uma String, que representará o resultado da tentativa de compra. O método comprar livro irá procurar, no vetor, um Livro com o mesmo título que o recebido como parâmetro de entrada. Caso seja encontrado um livro e ele esteja disponível para venda (há quantidade disponível), o método atualiza a quantidade disponível daquele título, atualiza o caixa da Livraria (para incluir o valor do livro vendido) e retorna a String "SUCESSO". Caso seja encontra um livro, mas não haja quantidade disponível para venda, o método retorna a String "ESGOTADO". Caso não seja encontrado um livro, o método retorna "NÃO ENCONTRADO".

8. Escreva um programa, no qual você usará a classe Livraria para oferecer ao usuário um menu com as seguintes opções: (1) Cadastrar um novo livro no estoque; (2) Listar o estoque; (3) Vender um livro; (4) Consultar o caixa da livraria; (5) Encerrar o programa. Ao escolher a opção 1, você deve ler os dados do livro e tentar guarda-lo na livraria (informe ao usuário do resultado). Ao escolher a opção 2, você deve listar todos os livros que estão cadastrados na livraria e a quantidade disponível de cada um. Ao escolher a opção 3, você perguntará ao usuário qual o título do livro que ele deseja comprar e usar o método "comprar livro" da Livaria (informe ao usuário do resultado). Ao escolher a opção 4, informe o valor do caixa atual da livraria. O programa deve continuar a exibir o menu, enquanto o usuário não escolher a opção 5. Caso ele escolha uma opção inválida, avise que a opção não é válida.