LP1A3 – Exercícios

Aula 6 – POO - Polimorfismo, Atributos e Métodos Finais e Estáticos, Classes Abstratas e Interfaces

Instruções para entrega das listas de exercícios:

Meio de Entrega: As resoluções das listas de exercícios devem ser entregues exclusivamente por meio do ambiente Moodle (http://eadcampus.spo.ifsp.edu.br).

Forma de Entrega: Para exercícios com uma única classe, deve ser entregue o arquivo da classe (extensão JAVA) referente ao exercício. Por exemplo: Exercicio3.java. Para exercícios com mais de uma classe, cada exercício deve ter uma pasta, na qual serão colocados os arquivos JAVA referentes ao exercício. Por exemplo: Para o Exercício 4, deve existir uma pasta "Exercicio4" contendo todos os arquivos JAVA deste exercício. **Entregue apenas os arquivos JAVA**. Todos os arquivos da lista devem ser compactados em um único arquivo (extensão RAR ou ZIP), cujo nome deverá conter a aula, o nome e um sobrenome do aluno. Por exemplo: Aula2_JoaoSilva.zip.

Prazo de Entrega: O prazo de entrega está definido na própria página de exercícios do Moodle, lembrando que o sistema bloqueia o envio de arquivos após a data e horário indicados.

Obs.: A resolução deste(s) exercício(s) deve ser feita de forma INDIVIDUAL. Listas de exercícios com uma ou mais respostas idênticas serão desconsideradas integralmente para efeitos de nota de participação.

1. Faça um programa em Java que leia os valores das vendas de uma loja nos últimos seis meses e acumule o valor total em um atributo estático por meio de um método dinâmico, o qual será chamado por uma instância v1 do tipo **Venda**. Ao final, apresente o valor total das vendas ao usuário por meio de outro método dinâmico, o qual será chamado por uma instância v2, também do tipo **Venda**. Transforme o atributo estático em dinâmico, execute novamente o programa e veja se o resultado foi alterado.

2. Faça um programa em Java que apresente ao usuário as opções a seguir, enquanto ele não digitar a opção 0 (zero). De acordo com o número da opção informada, o programa deverá efetuar a operação, solicitando as informações necessárias ao usuário.
   1) **Calcular raiz quadrada** – Deverá chamar o método sobrecarregado e estático ExecutarCalculo(double num) da classe **Calculo**.
   2) **Calcular potenciação** – Deverá chamar o método sobrecarregado e estático ExecutarCalculo(double base, double expoente) da classe **Calculo**.
   3) **Calcular fatorial** – Deverá chamar o método sobrecarregado e estático ExecutarCalculo(int num) da classe **Calculo**.

3. Faça um programa em Java que apresente ao usuário as opções a seguir, enquanto ele não digitar a opção 0 (zero). De acordo com o número da opção informada, o programa deverá efetuar a operação, solicitando uma cadeia de caracteres ao usuário.
   1) **Informar texto** – Deverá chamar o método setTexto(String texto) da classe **Pesquisa**.
   2) **Buscar string** – Deverá chamar o método BuscarString(String cadeiaCaracteres) da classe **Pesquisa**, que retornará se a cadeia de caracteres procurada existe ou não no texto.
   3) **Buscar string no início** – Deverá chamar o método sobrescrito BuscarString(String cadeiaCaracteres) da classe **PesquisaInicio**, filha da classe Pesquisa, que retornará se a cadeia de caracteres procurada existe ou não no início do texto. Configure este método para que não possa ser sobrescrito novamente em uma eventual classe filha da classe PesquisaInicio.

4) **Buscar string no fim** – Deverá chamar o método sobrescrito BuscarString(String cadeiaCaracteres) da classe **PesquisaFim**, filha da classe Pesquisa, que retornará se a cadeia de caracteres procurada existe ou não no final do texto. Configure este método para que não possa ser sobrescrito novamente em uma eventual classe filha da classe PesquisaFim.

4. **(Atividade assíncrona referente ao dia 29/08/2020)** Faça um programa em Java que apresente ao usuário as opções a seguir, enquanto ele não digitar a opção 0 (zero). De acordo com o número da opção informado, o programa deverá calcular a quantidade de tempo, em dias (opção 1) ou em horas (opção 2), entre uma data/hora inicial e uma data/hora final. Após realizar o cálculo, o programa deverá apresentar o resultado ao usuário.
   1) **Calcular dias** – Solicite a data/hora inicial e a data/hora final no formato "dd/mm/yy HH:mm". Os dados recebidos devem ser passados como argumento pelo construtor da classe **Dia**. A partir deste construtor, deve ser chamado o método **calcularTempo()** implementado nesta classe. O programa deve retornar inclusive a parte decimal dos dias, por exemplo: 4,8 dias.
   2) **Calcular horas** – Solicite a data/hora inicial e a data/hora final no formato "dd/mm/yy HH:mm". Os dados recebidos devem ser passados como argumento pelo construtor da classe **Hora**. A partir deste construtor, deve ser chamado o método **calcularTempo()** implementado nesta classe. O programa deve retornar inclusive a parte decimal das horas, por exemplo: 77,2 horas.

   Obs.: Deverá haver uma classe abstrata **Tempo** que possuirá o método abstrato **calcularTempo()**, o qual é implementado nas classes **Dia** e **Hora**. Esta classe também deverá ter os atributos **dataHoraInicial** e **dataHoraFinal**, além de seus respectivos métodos **get** e **set**. **Dica:** Para fazer os cálculos, use o método **parse** da classe **SimpleDateFormat** e os métodos **getInstance**, **setTime** e **getTimeInMillis** da classe **Calendar**. Os dias podem ser obtidos subtraindo a data inicial (em milisegundos) da data final (em milisegundos), e dividindo o resultado por (24 * 60 * 60 * 1000). Siga a mesma lógica para as horas, porém dividindo o resultado por (60 * 60 * 1000).

5. **(Atividade assíncrona referente ao dia 29/08/2020)** Faça uma cópia do programa do exercício anterior. Transforme a classe abstrata **Tempo** em uma interface e faça as alterações necessárias para que o programa funcione. **Obs.:** Lembre-se que uma interface não permite atributos dinâmicos, nem a implementação de métodos concretos e dinâmicos (com implementação e manipulação de atributos dinâmicos).