



Universidade Tecnológica Federal do Paraná
Câmpus Guarapuava
Curso de Tecnologia em Sistemas para Internet
Professor Eleandro Maschio
Pensamento Computacional e Fundamentos de Programação

Atividade Avaliativa 1 (AA2): Estruturas Condicionais

Importante

- A avaliação pode ser feita em equipes de **até 3 acadêmicos da mesma turma**. Não é necessário ser a mesma equipe da Lista de Exercícios 2 (LE2).
- O **nome completo de todos os integrantes** da equipe, além da turma, devem constar como comentário no início do código.
- Todas as postagens da equipe devem ser concentradas em **um único perfil (usuário)** no Moodle.
- Mesmo que o acadêmico **opte por fazer sozinho**, o nome completo e a turma devem ser, igualmente, informados.
- **Não compartilhe soluções**. As soluções serão submetidas a um **detector de plágio** (que observa, inclusive, variações). Caso seja acusado plágio entre soluções (ou de outra fonte, como a Internet), todos os envolvidos, independentemente de quem recebeu ou forneceu, terão nota zero e registro de ocorrência disciplinar (conforme antecipado no Plano de Ensino).
- Prazo de envio: **23h55 de domingo, 4 de abril de 2021**.
- Não serão aceitas resoluções **fora do prazo e/ou da plataforma Moodle**.
- Não arrisque ao enviar a solução no **limite do horário**.
- Cada acadêmico é responsável pelo **envio correto da própria solução**. Faça o teste de baixar e abrir o arquivo depois de enviado.

Enunciado

Modele e implemente uma classe denominada **Cheque**, com um único atributo numérico inteiro. Esse atributo representa o valor daquele cheque, compreendido no intervalo aberto (0, 1.000.000). Forneça um método que retorne o valor por extenso em minúsculas, conforme a seguinte assinatura: **public String getValorPorExtenso()**.

Exemplo de retorno: cento e vinte e três mil e quatrocentos e cinquenta e seis reais

Adicione testes unitários (JUnit) por meio da classe **TesteCheque**. Obviamente, a classe **Cheque** deve passar nos referidos testes.

Dicas

- Escolha da estrutura condicional (**if-else** ou **switch-case**) mais adequada a cada situação.
- Uso eficiente do **return**.
- Divisão de responsabilidades, uma vez que a solução pode ser composta por mais de um método.
- Criação de um bom conjunto de testes. A solução será submetida a testes diversos.

Restrições

- Os testes devem estar em conformidade com 4devs.com.br/numero_por_extenso.
- Caso o valor esteja fora dos limites, retorne uma cadeia de caracteres vazia.
- A solução **não deve utilizar**:
 - método `main()`.
 - instruções de entrada e saída.
 - estruturas de repetição.
 - enumerações, matrizes, listas ou qualquer outra estrutura de dados.
 - classes externas ao pacote `java.lang`.

Postagem

- Na Moodle, em tarefa específica, devem ser postados os dois arquivos: `Cheque.java` e `TesteCheque.java`.