

## Trabalho Prático II

### JAVA: Programando um Jogo de Cartas

---

#### Instruções Gerais

- O trabalho é de especificação, projeto e implementação de um Jogo de Cartas em Java e tem como finalidade a aplicação dos conceitos de Programação Orientada a Objetos (POO) estudados durante o semestre letivo.
- O trabalho é **individual** e cada aluno deverá apresentar sua solução, contudo são incentivadas discussões de idéias, desafios e soluções em grupo.
- Parte do trabalho poderá ser implementado no Laboratório 30 do DECOM às segundas e quartas das 8h20 às 10h, horários em que serão inclusive contabilizadas presenças.
- Não serão aceitos trabalhos que caracterizem cópia (mesma estrutura e algumas pequenas modificações) de outro aluno ou de códigos da *Internet* (será usada uma ferramenta para detecção automática de plágio de códigos);

#### Sobre o Jogo de Cartas

Neste segundo trabalho, é esperado que cada aluno escolha, especifique e implemente em JAVA um “Jogo de Cartas”, que deverá conter funcionalidades e desafios para dois níveis diferentes de dificuldade: iniciante e avançado, ou, opcionalmente, para duas outras formas de interação distintas (e.g., dois participantes ou mais). O jogo deverá permitir a interação via *mouse* e também via teclado e ter uma interface gráfica (implementada usando um “*widget toolkit GUI*”, como o *Swing* ou outro). É permitida a utilização de quaisquer bibliotecas, inclusive de terceiros, que julgarem necessárias. Exemplos de jogos de cartas podem ser encontrados em vários locais na *Internet*, incluindo em: <http://www.1001jogos.pt/cartas>.

#### Sobre a Entrega

- Deve ser especificado um relatório, em formato “*.pdf*”, contendo as seguintes informações:
  - Identificação do aluno
  - Descrição do Jogo de Cartas escolhido e dos níveis de dificuldade implementados OU diferentes formas de interação.
  - Projeto do Sistema Proposto: especificar o diagrama de classes usando *UML*<sup>1</sup>.
  - Decisões de Projeto: descrever e justificar suas principais decisões de projeto.
  - Considerações finais: Discutir, em linhas gerais, impressões sobre a atividade, incluindo pontos positivos e desafios.
- O código-fonte e também o Makefile devem ser entregues via Run.Codes juntamente com o

---

1 Unified Modelling Language.

relatório em um arquivo “.zip”.

- Data de Entrega: Até 30 de Março às 12h00 via Run.Codes (**limite firme que NÃO SERÁ ALTERADO**).

## Sobre a Apresentação

- Cada aluno deverá apresentar seu trabalho, prevalecendo a ordem de chegada dos alunos. A seguir, os horários disponíveis para apresentação:
  - Quinta-feira (30/03) @Lab30 do DECOM
    - 13h-18h
  - Sexta-feira (31/03) @Lab30 do DECOM
    - 13h – 16h.

## Sobre a Avaliação das Soluções

Serão critérios de avaliação:

- Funcionamento adequado do programa
  - Compilação (códigos que não compilem serão zerados, e *warnings* diminuirão a nota);
  - Corretude (independente se gerado por IDE ou manualmente).
- Clareza e corretude do relatório e do diagrama de classes;
- Código devidamente indentado e comentado;
- Definição e Tratamento adequado de exceções.
- Uso correto e adequado de conceitos de POO vistos em aula, inclusive os mais avançados.
- Decisões de projeto que promovam reúso, alta coesão e baixo acoplamento dos módulos de *software*.
- Usabilidade e adequação dos componentes da interface gráfica.

## Sobre a Distribuição de Pontos Extras

Somente serão atribuídos pontos extras ao trabalho se a nota original for maior ou igual a 60%.

- Melhor interface (somente um trabalho): 10%;
- Efeitos Sonoros: 5%.