

## Trabalho Prático 1 – Cartas na mesa

**Valor: 12 (+2) pontos**

**Data de entrega: 16/04/2018**

O objetivo deste trabalho prático é familiarizar o aluno com os principais conceitos de Orientação a Objetos e com a linguagem Java. O aluno deverá executar as etapas de análise, modelagem e desenvolvimento de um sistema referente a um jogo de cartas. O programa deverá ser feito baseado no JAVA SDK 8.

### 1. Introdução

O aluno deverá implementar as entidades necessárias para a implementação de um jogo de cartas, assim como as principais funcionalidades relacionadas a um jogo específico. Abaixo é apresentada uma lista de sugestões de jogos que podem ser implementados. A lista não é definitiva e os alunos podem sugerir outros jogos de seu interesse. A aplicação deverá ser 100% funcional, com a implementação das regras necessárias. **Cada jogo poderá ser escolhido por apenas um grupo!**

Sugestões de jogos de cartas:

1. Pôquer
2. Blackjack
3. Truco
4. Paciência
5. UNO
6. Magic
7. Super Trunfo

Todo o desenvolvimento e a entrega deverão ser feitos utilizando o sistema de controle de versão GitHub<sup>1</sup>. Sugere-se que commits/pushs sejam feitos de maneira frequente, pelo menos uma vez ao dia.

O calendário de atividades do trabalho é mostrado abaixo:

Atividade	Data
Definição do jogo (Moodle)	Até 26/03/2018
Apresentação em sala de aula	16/04/2018
Entrega do trabalho	16/04/2018

Lembre-se, o objetivo não é apenas escrever um programa funcional, mas desenvolver um sistema confiável, reutilizável e de fácil manutenção e extensão! Logo, tente utilizar os conceitos de orientação a objetos e modularidade vistos até o momento em sala de aula. Também serão avaliados critérios como criatividade na solução, assim como a possível implementação de funcionalidades extras.

---

<sup>1</sup> <https://github.com/>

A interface de interação com a aplicação poderá ser feita via terminal de comando. Possíveis arquivos necessários durante a etapa de inicialização do sistema deverão ser fornecidos pelo grupo.

Exemplo de jogabilidade de uma partida de Blackjack via terminal.

```
Bem-vindo ao Blackjack PM 2018/1!

Quantos jogadores participação? <1 - 4>: 2
Nome do jogador 1: Joao
Nome do jogador 2: Maria

-- Embaralhando e separando cartas --

Joao: 6C 3C <9>
Maria: 3D JC <13>
Dealer: XX QS

Joao, voce deseja outra carta? <y/n> y
Joao: 6C 3C 6S <15>
Joao, voce deseja outra carta? <y/n> y
Joao: 6C 3C 6S 4H <19>
Joao, voce deseja outra carta? <y/n> n

Maria, voce deseja outra carta? <y/n> y
Maria: 3D JC JD <23>
Maria perdeu!

Dealer: 8H QS <18>

##### Joao é o vencedor! #####

Deseja jogar novamente? <y/n>: n
```

## 2. Documentação

Um relatório deve ser entregue ao final dessa etapa, de forma a deixar claro as decisões tomadas relativas aos casos e detalhes que porventura estejam omissos no enunciado. Adicione também comentários gerais sobre o trabalho e as principais dificuldades encontradas.

Para familiarizar o aluno com o processo inicial de modelagem, a estrutura do sistema deverá ser apresentada utilizando-se cartões CRC.

Além dos artefatos anteriormente mencionados, o grupo também deverá realizar a documentação do código utilizando Javadoc<sup>2</sup>. Nesse caso, apenas é necessário adicionar a funcionalidade/descrição do método e utilizar pelo menos as tags @param e @return.

---

<sup>2</sup> <https://en.wikipedia.org/wiki/Javadoc>

Entre outras coisas, o relatório deve conter:

1. Introdução: descrição do problema/jogo e visão geral sobre o funcionamento do programa.
2. Implementação: descrição sobre a implementação do programa. Devem ser detalhadas as estruturas de dados utilizadas, o funcionamento das principais funções, bem como decisões tomadas. Possível descrição de extras implementados.
3. Conclusão: comentários gerais sobre o trabalho e as principais dificuldades encontradas em sua implementação.
4. Bibliografia: bibliografia utilizada para o desenvolvimento do trabalho, incluindo sites da Internet se for o caso.

### 3. Comentários Gerais

- Comece a fazer este trabalho logo, enquanto o problema está fresco na memória e o prazo para terminá-lo está tão longe quanto jamais poderá estar.
- Clareza, endentação e comentários no programa também serão avaliados.
- O trabalho deverá ser feito em grupos de no mínimo 3 e no máximo 4 alunos.
- Trabalhos copiados serão penalizados conforme anunciado.
- Penalização por atraso:  $(2^d - 1)$  pontos, onde  $d$  é o número de dias de atraso.
- Deve ser fornecido com o código um arquivo Makefile com as opções 'make' e 'make run'.
- As entregas devem ser feitas utilizando o conceito de Release<sup>3</sup> do GitHub.

### 4. Critérios de avaliação

- Cartões CRC (2 pts).
- Funcionamento correto (2 pts).
  - Resposta correta, não apresenta crash.
- Uso correto dos conceitos de OO e Modularidade (4 pts).
  - Abstração, Encapsulamento, Herança e Polimorfismo.
  - Pacotes, Componentes reusáveis.
- Documentação e Estilo (2 pts).
  - Javadoc, comentários, endentação.
- Apresentação em sala (2 pts).
- Criatividade, extras (+1 pt).
- Participação (+1 pt).

---

<sup>3</sup> <https://help.github.com/articles/creating-releases/>