

Trabalho Prático 2 – Cartas na mesa

Valor: 10 (+1) pontos

Data de devolução: 13/10/2014

O objetivo deste trabalho prático é familiarizar o aluno com as etapas de análise e projeto de um sistema referente a um jogo de cartas.

Instruções:

O aluno deverá implementar as entidades necessárias para a implementação de um jogo de cartas (Cartas, Mão, Baralho, etc), assim como as funcionalidades relacionadas à um jogo específico. Abaixo é apresentada uma lista (não exclusiva) de sugestões de jogos que podem ser implementados. A aplicação deverá ser 100% funcional, com a implementação de todas as regras específicas do jogo.

Sugestões de jogos de cartas:

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

ATENÇÃO: No máximo 4 duplas poderão escolher o mesmo jogo!

O calendário de atividades do trabalho é mostrado abaixo:

Atividade	Data
Definição do jogo (Moodle)	Até 22/09
Entrega do trabalho	13/10/2013

Tente utilizar os conceitos de orientação a objetos vistos em sala de aula (herança, polimorfismo, etc), técnicas de modularização, e possíveis padrões de projeto que sejam aplicáveis. Também serão avaliados critérios como criatividade na solução e também a implementação de funcionalidades extras.

A interface de jogabilidade da aplicação deverá ser feita via terminal de comando. Possíveis arquivos necessários durante a etapa de inicialização do sistema deverão ser fornecidos pela dupla.

Exemplo de jogabilidade de uma partida de Blackjack.

```
Bem-vindo ao Blackjack PM 2014/2!

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
```

Documentação:

Deve ser composta por três documentos:

- Diagrama de classes (UML)
- Diagrama de atividades (UML)
- Relatório

Entre outras coisas, o relatório deve conter:

1. **Introdução:** descrição do problema 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.

O que deve ser entregue:

Envie um arquivo ZIP com o nome no formato 'tp1-nome1-nome2.zip', contendo os seguintes arquivos:

- Arquivo README com os nomes completos dos alunos da dupla.
- O código fonte do programa em, Java bem indentado e comentado. Deve ser fornecido junto com o fonte um arquivo Makefile com as opções 'make' e 'make run'.
- A documentação do trabalho bem escrita e detalhada.

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, indentação e comentários no programa também serão avaliados.
- O trabalho poderá ser feito em dupla.
- Trabalhos copiados serão penalizados conforme anunciado.
- Penalização por atraso: $(2^d - 1)$ pontos, onde d é o número de dias de atraso.

Critérios de avaliação:

- Funcionamento correto (3 pts).
- Uso correto dos conceitos de OO (4 pts).
- Documentação (3 pts).
- Criatividade, extras (1 pt)