



# **Programação Orientada por Objetos**

## Grupo 21

Bernardo Nabais Gomes – 75573

José Manuel Dias – 75847

Diogo Ribeiro Ferreira - 76090

Lisboa, 15 de Maio de 2016

# 1. Herança e Polimorfismo

## 1.1. *Person, Dealer e Player*

Através da análise ao jogo implementado, existem ações e parâmetros comuns entre os intervenientes: contagem estatística, hit e a mão que está a ser jogada. Deste modo e tendo por base a terminologia do jogo real de casino criou-se a superclasse *Person* que contém todas as variáveis e métodos comuns aos jogadores. As subclasses *Dealer* e *Player* herdam todos os atributos, e definem os métodos específicos de cada um – extensão da *Person*. Esta divisão permite que vários *Players* e *Dealers* sejam instanciados, o que em conjunto com a definição da classe *Game* possibilita vários jogos em simultâneo.

## 1.2. *Game, Debug e Simulation*

Por forma a implementar o jogo no formato mais comum (interactivo), foi criada a classe *Game*, com os diferentes métodos a ela associados. Sendo a diferença para os restantes modos apenas a forma como o jogo recebe os comandos do jogador, foram criadas as classes *Debug* e *Simulation* que irão fazer override aos métodos da classe anterior. Assim, no modo de debug, os comandos irão ser retornados de uma linked list formada pelos valores de um ficheiro, e no modo simulation estes dependerão das estratégias seguidas pelo jogador.

Adicionalmente, fez-se override de mais métodos do *Game* no modo de simulação por forma a que os outputs do terminal não apareçam, sendo apenas mostradas as estatísticas no final da simulação.

# 2. Interfaces

## 1.1. *BetStrategy e ActionStrategy*

Criaram-se duas interfaces com argumentos de contagem distintos. Na estratégia de *Bet*, a contagem é calculada através da observação de cartas que vão saindo e o jogador é aconselhado no valor da sua aposta. Neste sentido, esta interface tem um único método de advice, sem qualquer argumento que será implementada pela estratégia *Ace-five strategy*.

Na *ActionStrategy* a contagem é através das cartas e vão saindo e das cartas em jogo nessa ronda, razão pela qual o método advice recebe a mão actual do jogador e a carta revelada pelo dealer.

Optou-se por fazer a divisão por uma questão de estruturação, legibilidade e reutilização do código para outras estratégias que possam implementar umas das duas interfaces anteriores.

### 3. *ArrayList* vs *LinkedList*

Para armazenamento das cartas no shoe, usou-se um *ArrayList* dado que o tamanho máximo é conhecido e calculado a partir do número de decks. A escolha teve em conta o contexto do Blackjack: o número de cartas distribuídas numa ronda é muito inferior ao número total de cartas, o que significa que existem poucas remoções ou inserções de cartas do shoe com *Linked List*, em que a complexidade é  $O(1)$ , ou seja, não existiria grande vantagem. Apesar do possível ganho em termos de memória pela remoção das cartas que iriam saindo, seria contra balanceado por cada elemento do *Arraylist* ocupar menos espaço em memória.

Para armazenar as mãos do jogador e cada carta em cada mão escolheu-se *LinkedList* pelo facto do número de cartas por mão e o número de mãos por jogador serem variáveis e serem alvo de sucessivas operações de inserção (hit e split) e remoção (split).

### 4. Interface gráfica

A interface gráfica foi feita com recurso ao Swing. Desenhou-se o ambiente visual, colocaram-se os botões tendo em conta os comandos executáveis a partir do terminal aos quais se associaram métodos no evento de clique, adicionou-se o campo de bet que recebe o único parâmetro alterável durante o jogo e adicionou-se uma caixa de diálogo que permite introduzir os parâmetros de jogo. As acções permitidas a cada jogada foram controladas através da mudança do parâmetro *Enable* dos botões.

Cada frame mostra apenas a mão corrente do jogador. Cada interveniente no jogo tem um painel associado onde é mostrada a mão com que está a jogar. Cada carta do jogo é associado a uma imagem através de um hashmap. Quando o jogador pede uma carta é criada uma nova label, associada à carta e adicionada ao painel.