

## PROJETO PRÁTICO – PROGRAMAÇÃO ORIENTADA A OBJETOS

### Objetivos:

- Elaborar um programa que cumpra os pré-requisitos enumerados no enunciado.
- Comentários no código-fonte são encorajados e alvo de avaliação.
- Otimização é alvo de avaliação.
- **É obrigatório o programa executar** (sem erros de compilação).
- Deve ser usada linguagem Java para a resolução do presente projeto prático.
- Deve ser entregue na tarefa do Microsoft Teams. Pasta do Projeto comprimida (.zip) com o nome TP\_POO\_(nome\_formando)

### EXERCÍCIOS

Após ter terminado o curso de **Software Developer**, foi contactado por um estúdio de videojogos para desenvolver o novo Human Life Simulator. Este jogo trata-se de uma Pessoa que deve fazer escolhas, crescer e chegar ao seu objetivo de vida.

Para isso:

- Crie uma classe abstrata “**Pessoa**” com os atributos:
  - nome (String) – Nome da Personagem
  - dinheiro (double) – Dinheiro que a Pessoa ganhou
- Crie a subclasse **Jogador** com os atributos:
  - objetivoVida (Objetivo) – Objetivo do Jogo
  - profissão (Profissao) – Profissão Atual da Pessoa
  - necessidadeSono (int) – Necessidade de Dormir
  - necessidadeRefeicao (int) – Necessidade de Comer/Beber
  - necessidadeSocial (int) – Necessidade de Intergari/Divertir
  - estatuto (int) – Somatório dos Estatutos de todas as suas Propriedades
  - educação (int) – Nível de Formação da Pessoa
  - propriedades (ArrayList<Propriedade>) – Todas as Propriedades
  - família (ArrayList<NPC>) – Família do Jogador
  - Crie o método mostrarDetalhes que escreva na consola todos os detalhes da pessoa.
- Crie a subclasse **NPC** que age como outras Pessoas do jogo, não controláveis, com os atributos:
  - EstatutoMinimo (int)

Pode criar um jogador só com singleton

- Crie a classe **Propriedade** deverá ter atributos:
  - nome (String)
  - custo (double)
  - estatuto (int)
- Desenvolva as subclasses de propriedade: Imóvel, Veículo e AcessórioModa.
- A classe **Imóvel** deverá ter os atributos:
  - capacidadePessoas (int)
- A classe **Veículo** deverá ter os atributos:
  - marca (String)
  - modelo (String)
- A classe **AcessórioModa** deverá ter os atributos:
  - marca (String)
  - formal (boolean)

?? CONSTRUTOR CORRETO ??  
BOOLEAN
- Seguidamente desenvolva a classe **Profissão** que deverá ter os atributos:
  - nome (String)
  - salarioDia (double)
  - formal (boolean)
  - estatuto (int)
  - nívelMinimoEducacao (int)

?? CONSTRUTOR CORRETO ??  
BOOLEAN
- A classe **Imóvel** deverá ter os atributos:
  - capacidadePessoas (int)
- A enumeração **Objetivo** deverá descrever quais os Objetivos de Vida que uma Pessoa pode ter, como por exemplo: Milionário, Família Completa, Celebridade, Professor, etc..
- Crie a classe **Shopping** que vai simular um conjunto de lojas, como tal, deve ter os atributos:
  - ArrayList<Propriedade> coisasParaComprar
- Para shopping, desenvolva o método vender que recebe a Pessoa como parâmetro e pergunta qual secção do shopping a pessoa quer aceder (imobiliária, stand ou fashion outlet), de acordo com a escolha imprima na consola 10 itens disponíveis da secção adequada (mesmo que tenha mais no array, apenas 10 aleatórios devem aparecer). Após esta impressão, pergunta ao jogador qual quer comprar e, caso tenha dinheiro, permite a compra e adiciona ao array de propriedades. (Pode dividir em mais métodos, por exemplo: hallShopping, imprimirImoveis, imprimirStand, imprimirFashion, vender, ...)

- Posteriormente, desenvolva a classe **Sims**, que vai servir **o propósito de classe de Jogo**. Nesta classe, desenvolva o método `criarPessoa()` que permite criar a Pessoa a partir de feedback da consola. O atributo dinheiro começa a 0, a profissão começa a nula, as necessidades começam a 100 (limite máximo), o estatuto começa a 0, e educação começa a 0 e as propriedades também vazias.
- Seguidamente, crie o método `jogo()` onde estão todas as **instâncias de todos os objetos, onde está o fluxo da aplicação**.
- Este terá um ciclo para cada dia da vida da pessoa, cada dia é composto por 4 momentos (manhã, meio-dia, tarde, noite), e em cada momento o jogador pode escolher o que quer fazer, as opções são:
  - Ir trabalhar: aumenta dinheiro do jogador com base no salário/dia da sua profissão.
  - Dormir: repõe a necessidade de sono de volta a 100.
  - Ter uma refeição: repõe a necessidade de refeição de volta a 100 e diminui 5 dinheiros.
  - Falar com alguém/Jogar Computador/Praticar Hobby: repõe a necessidade social de volta a 100.
  - Ir às compras: invoca o método `vender` do shopping.
  - Ter formação: aumenta a educação em 2.
  - Visitar as propriedades: lista todas as propriedades do jogador.
  - Procurar nova profissão: lista as profissões disponíveis, e de acordo com dois fatores o jogador será aceite ou não num emprego:
    - Se contém, pelo menos, um acessório de moda formal no caso de o emprego ter o atributo `formal` a `true`.
    - Se contém o estatuto no nível mínimo de entrada no emprego.
    - Se contém a educação no nível mínimo de entrada no emprego.
- Atente que, a **cada iteração do ciclo, a necessidade sono diminui 25 pontos, a necessidade refeição diminui 20 pontos** e **a necessidade social diminui 15 pontos**.
- **Se a Pessoa tiver alguma das necessidades abaixo de 25, o jogador não poderá fazer nenhuma escolha, que não a escolha que satisfaz essa necessidade.**

- Para além dessa rotina, este ciclo deve ter escolhas de vida obrigatórias, como por exemplo:
  - Dia 5 deve escolher se quer ir para a universidade, caso sim, a educação aumenta 50, mas contrai uma dívida de 3.000 dinheiros.
  - Dia 22 deve escolher se quer casar, caso sim, pergunta com quem listando todos os NPCs no jogo. O jogador pode selecionar qualquer um, desde que tenha uma propriedade que albergue 2 ou mais pessoas. E terá também de ter o estatuto mínimo para casar com determinado NPC. Se conseguir casar aumenta ao dinheiro do jogador todo o dinheiro que esse NPC tem. Para além disso, a cada dia que passa, o dinheiro aumenta 30/dia. Também desbloqueia uma nova opção no menu para ter/adotar filhos, que de acordo com um fator aleatório permite acrescentar um NPC aleatório sem dinheiro ou estatutoMinimo ao Array da família (verifique que a propriedade com maior capacidade permita albergar a nova Pessoa). (Só terá esta opção disponível no menu até ao dia 60).  
Para cada Pessoa no Array Família, o jogador terá de pagar 10 dinheiros/dia.  
Atente que, o limite mínimo de dinheiro é -3250. Se o jogador ultrapassar esse limite devido ao custo de alimentação, a SS retira todos os filhos.
  - Acrescente mais eventos bons e maus para dar dinâmica ao jogo (deve ter, pelo menos, 6).
- Pode acrescentar mais opções ao menu à sua vontade, sendo o limite a criatividade... Por exemplo: jogar no euromilhoes, que diminui 10 dinheiros e pergunta ao jogador um número, se esse número coincidir com um aleatório entre 1 e 200, aumenta 1.000.000 dinheiros.
- O ciclo vai correr x vezes, sendo x o número de dias de vida da pessoa. Vamos ter em conta o número 100 (exemplo), ao fim da iteração do ciclo deve analisar tudo do Jogador e perceber se alcançou o seu objetivo de vida. Se sim, então ganha o jogo, se não, então deve permitir jogar novamente com a mesma Personagem ou começar o Jogo com uma nova Personagem. Por exemplo: no início o jogador escolheu o objetivo de Milionário, então deve ter a soma de todo o valor das suas propriedades mais o seu dinheiro atual acima de 1.000.000. Ou então escolheu Família Completa, deve ter no seu Array de Família 5 ou mais Pessoas.
- O Jogador não pode ter dívidas no momento do fim do jogo, caso contrário perderá sempre.
- Ponto de Valorização: Carregar o ArrayList do Shopping através de ficheiro.
- Deve gerar o **JAVADOC**.

**Bom trabalho! 😊**