

Sumário

- Classes Abstratas

Documentação complementar Java:

- [Classes Abstratas](#)
- [Classe Object](#)

Nota: Gere o JavaDoc para o(s) projeto(s) utilizado(s) na resolução desta ficha de trabalho.

Parte 1

Considere uma empresa de desenvolvimento de *software*. De um modo geral pode afirmar-se que independentemente do tipo de funções que um colaborador da mesma desempenhe será sempre necessário obter a seguinte informação de uma pessoa


- Nome (*name*)
- Data de nascimento (*birthDate*)
- Morada (*address*)
- Número de cartão de cidadão (*citizenCard*)
- Número de identificação fiscal (*vatNumber*)
- Salário base (*baseSalary*)

No contexto da empresa, os colaboradores podem ser Administrativo, Programador Java, ou Gestor de Projeto. Tenha em atenção as seguintes características específicas para cada

- Administrativo (*Administrative*)
 - Sigla (*initials*)
 - Tipo de contrato (*contractType*): Parcial (*partial*) ou Integral (*integral*)
 - Habilitações literárias (Licenciatura, Mestrado ou Doutoramento)
 - Data da contratação (*beginContract*)
 - Data de fim de contrato (*endContract*)
 - Pode não estar definida (contrato sem termo) se o contrato for a tempo integral
- Programador Java (*JavaProgrammer*)
 - Código de funcionário (*code*)
 - Número de anos de programação Java (*programmingYears*)
 - Nome do projeto que se encontra a desenvolver (*projectName*)
 - Tipo de programador (*programmerType*): Júnior ou Sénior
- Gestor de Projeto (*ProjectManager*)
 - Código de funcionário (*code*)
 - Conjunto de projetos que gere (*projects*)
 - Data da contratação (*contractDate*)

Todas as pessoas que colaboram na instituição têm um salário que varia de acordo com as funções que desempenha. Assim, considere que:

- Administrativo: o salário inicial é calculado de acordo com a percentagem do contrato estabelecido. Sobre este valor atente que deve somar mais 10%, 20% ou 30% de acordo com as habilitações literárias.
- Programador: o salário inicial é calculado como sendo o 105% do salário base. Sobre este valor é adicionado 5% ou 15% de acordo com o tipo de programador e 0.5% por cada ano como programador.
- Gestor: o salário inicial é calculado como sendo o 115% do salário base. Por cada ano de contrato é ainda adicionado 1.5% do salário inicial e 1% por cada projeto que este se encontra responsável.

 <p>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</p>	<p>LEI/LSIRC PP - Paradigmas de Programação 2º Semestre ■ Docentes: RJS, BMO, CDF, MFG, OAO Ficha Prática 9 – 2023/2024</p>
--	--

Na resolução dos exercícios propostos considere que deve:

- Garantir o encapsulamento de todas as classes criadas
- Criar os métodos de acesso necessários para as classes criadas
- Criar, num *package* específico, as enumerações necessárias para suportar o problema apresentado
- Criar métodos específicos para manipulação de coleções, ou seja, não deverá ser permitido o acesso direto às variáveis que representam coleções, devendo existir métodos para adicionar, remover, editar e listar elementos.

Exercício 1

- 1.1) Crie um diagrama de classes de forma a estruturar e representar todas as classes e relações entre elas. O diagrama deve ajudá-lo a estruturar a resolução deste problema antes de o codificar no editor de desenvolvimento.
- 1.2) No `package pp_fp09.programming`, implemente o código Java para representar a estrutura descrita anteriormente.
- 1.3) Nas várias classes criadas adicione métodos que permitam imprimir os valores atribuídos nas variáveis das diversas instâncias.
- 1.4) Crie a classe `ProgrammerDemo` de forma a testar as classes implementadas. Inicialize alguns elementos relativos aos diferentes tipos de pessoas que fazem parte da instituição.

Exercício 2

- 2.1) Crie uma classe `ProgrammingManagement` que permita armazenar um conjunto de colaboradores. Implemente os métodos necessários para adicionar, remover e listar os colaboradores. Teste a classe com pelo menos de 5 colaboradores de tipos diferentes.
- 2.2) Crie um método que permita listar apenas gestores de projeto.

Parte 2

Considere o diagrama de classes apresentado na Figura 1 que exemplifica uma forma (*Shape*). Implemente as classes de acordo com o diagrama. Tenha em consideração a existência de métodos e classes abstratas na resolução do problema.

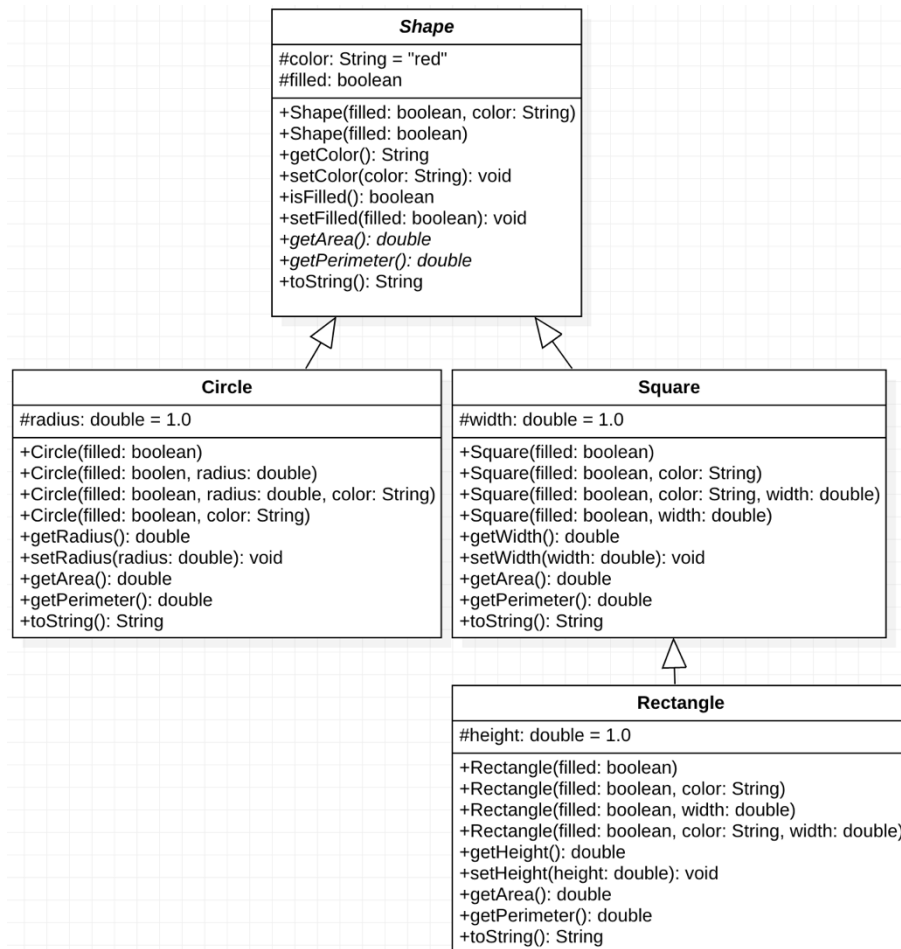


Figura 1 - Exemplo de diagrama de classes

Exercício 1

- 1.1) No package `pp_fp09.shapes`, implemente o código Java para representar a estrutura descrita anteriormente.
- 1.2) Nas várias classes criadas adicione métodos que permitam imprimir os valores atribuídos nas variáveis das diversas instâncias.
- 1.3) Crie a classe `ShapeDemo` de forma a testar as classes implementadas. Inicialize alguns elementos relativos aos diferentes tipos de `shapes`.
- 1.4) Crie uma classe `ShapeManagement` para guardar coleções de `Shapes`, com os métodos adicionar, remover e encontrar. Adicionalmente crie uma função para retornar `shapes` de um tipo específico (ex: `rectangle`) que se encontrem na coleção. Nota: não deverá permitir a duplicação de `shapes`.