



UNIVERSIDADE FEDERAL DO CARIRI
TECNÓLOGO EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

1ª ATIVIDADE AVALIATIVA

| | |
|------------------|---------------------------------|
| Professor | Ricardo Ferreira Vilela |
| Disciplina: | Programação Orientada a Objetos |
| Semestre Letivo: | 2023-1 |

Nesta lista de exercícios, você terá a oportunidade de praticar suas habilidades na Programação Orientada a Objetos. Os exercícios são baseados em um conjunto de classes previstos para um sistema capaz de armazenar companhias e pessoas, onde uma pessoa pode ser um funcionário de uma companhia ou proprietário da mesma.

Instruções Gerais:

- Certifique-se de que os nomes das classes, atributos e funções correspondam >>>**EXATAMENTE**<<< ao texto, pois serão avaliados por meio de testes automatizados;
- Cada classe solicitada deve ser criada em um arquivo separado;

Lembre-se de que o objetivo da atividade é, além da avaliação, uma prática na construção de código orientado a objetos, portanto, evite consultar soluções prontas.

EXERCÍCIOS

EXERCÍCIO 1: CLASSE PERSON

- a) Desenvolva uma classe denominada *Person* que inclua os seguintes atributos, cada um com seu tipo correspondente:

| Atributo | Tipo |
|--------------------|-----------------------------|
| <i>firstname</i> | <i>Cadeia de caracteres</i> |
| <i>lastname</i> | <i>Cadeia de caracteres</i> |
| <i>age</i> | <i>inteiro</i> |
| <i>address</i> | <i>Cadeia de caracteres</i> |
| <i>phoneNumber</i> | <i>Cadeia de caracteres</i> |
| <i>email</i> | <i>Cadeia de caracteres</i> |

- b) Crie um construtor para a classe *Person* que aceite todos os atributos como parâmetros, mantendo a mesma ordem em que foram apresentados na questão anterior.



UNIVERSIDADE FEDERAL DO CARIRI
TECNÓLOGO EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

- c) Crie um método na classe *Person* chamado *speak* que aceite uma mensagem como parâmetro (String message) e imprima a mensagem no console, conforme exemplo abaixo:

Paulo *says:* *Hello World!*

*Onde:

firstname = *Paulo*
message = *Hello World!*

- d) Crie um método na classe *Person* chamado *walk* que aceite uma distância em metros como parâmetro (int distanceInMeters) e imprima no console uma mensagem que inclua o primeiro nome da pessoa e a distância em metros. A mensagem deve indicar que a pessoa está caminhando essa distância, conforme exemplo a seguir:

Paulo *is walking* *10* *meters.*

*Onde:

firstname = *Paulo*
distanceInMeters = *10*

- e) Crie um método, do tipo inteiro, na classe *Person* chamado *increaseAgeByYears* que aceite um número de anos como parâmetro (int *yearsToAdd*), aumente a idade da pessoa em *yearsToAdd* e retorne a nova idade.
- f) Crie um método na classe *Person* chamado *eat* que aceite o nome de um alimento como parâmetro (String *food*) e imprima no console uma mensagem que inclua o primeiro nome da pessoa e o alimento, indicando que a pessoa está comendo esse alimento. Conforme apresentado abaixo:

Paulo *is eating* *cheese*

*Onde:

firstname = *Paulo*
food = *cheese*

- g) Crie um método na classe *Person* chamado *displayInfo* que imprima no console todas as informações da pessoa, exatamente como no exemplo abaixo:

First Name: *Paulo*
Last Name: *Silva*
Age: *20*
Address: *Avenue 5*
Phone Number: *88-99999999*
Email: *paulo@gmail.com*



UNIVERSIDADE FEDERAL DO CARIRI
TECNÓLOGO EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

EXERCÍCIO 2: CLASSE COMPANY

- a) Desenvolva uma classe chamada *Company* que inclua os seguintes atributos, cada um com seu tipo correspondente:
- *companyName*: Cadeia de caracteres
 - *employeeCount*: inteiro
 - *owner*: *Person* (onde *Person* é a classe que você definiu anteriormente)
- b) Crie um construtor para a classe *Company* que aceite os atributos *companyName*, *employeeCount* e *owner* como parâmetros, mantendo a mesma ordem em que foram apresentados na questão anterior.
- c) Crie um método na classe *Company* chamado *hireEmployee* que aceite um objeto *Person* como parâmetro para representar um novo funcionário. O método deve adicionar o novo funcionário à empresa e atualizar o contador de funcionários. Por exemplo, se você passar um objeto *Person* representando um funcionário chamado "Alice", o método deve imprimir no console uma mensagem indicando que Alice foi contratada e atualizar o contador de funcionários. A mensagem no console deve ser exibida conforme o exemplo abaixo:

Hired a new employee: Alice Souza

*Onde:

firstname = Alice
lastname = Souza

- d) Crie um método na classe *Company* chamado *scheduleMeeting* que aceite uma agenda como parâmetro (String agenda). O método deve imprimir no console uma mensagem indicando que uma reunião da empresa foi agendada com a agenda especificada. A mensagem no console deve ser exibida conforme o exemplo abaixo:

ABC Inc. Meeting Scheduled - Agenda: 09/27/2023 at 3pm

*Onde:

companyName = ABC Inc.
agenda = 09/27/2023 at 3pm



UNIVERSIDADE FEDERAL DO CARIRI
TECNÓLOGO EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

- e) Crie um método na classe *Company* chamado *displayCompanyInfo* que imprima no console todas as informações sobre a empresa, incluindo o nome da empresa, o número de funcionários e as informações do proprietário. O método deve chamar o método *displayInfo* da classe *Person* para exibir as informações do proprietário. A mensagem no console deve ser exibida **exatamente** como no exemplo abaixo:

Company Name: ABC Inc.
Number of Employees: 100
Owner:
First Name: Paulo
Last Name: Silva
Age: 20
Address: Avenue 5
Phone Number: 88-99999999
Email: paulo@gmail.com