

Prática 6

Conteúdos que serão praticados:

- Criação de Classe
- **Definição** de métodos de Classe
- **Declaração** de variáveis de Classe
- **Instanciação** de objetos
- **Chamada** de métodos de Classe através de objetos instanciados
- **Atribuição** de valores para as variáveis de Classe
- **Acesso** aos valores das variáveis de Classe
- Entrada de dados com `Scanner`
- Saída de dados com `System.out.print`
- ArrayList de objetos
- Laços de repetição

Atividade 1

1. Crie um projeto Java Padrão e dê o nome de `Atividade1`
2. Crie uma classe `Usuario`
3. Declare as seguintes **variáveis de classe** para `Usuario`:
 1. `email`: tipo `String` e acesso `private`
 2. `nome`: tipo `String` e acesso `private`
 3. `senha`: tipo `String` e acesso `private`
4. Defina os seguintes métodos na classe `Usuario`:
 1. `atribuirEmail`: um parâmetro `String` e retorno `void`; recebe um valor do tipo `String` e armazena na variável de classe `email`
 2. `atribuirNome`: um parâmetro `String` e retorno `void`; recebe um valor do tipo `String` e armazena na variável de classe `nome`
 3. `atribuirSenha`: um parâmetro `String` e retorno `void`; recebe um valor do tipo `String` e armazena na variável de classe `senha`
 4. `pegarEmail`: sem parâmetros e retorno `String`; retorna o valor da variável de classe `email`
 5. `pegarNome`: sem parâmetros e retorno `String`; retorna o valor da variável de classe `nome`
 6. `pegarSenha`: sem parâmetros e retorno `String`; retorna o valor da variável de classe `senha`
5. Na `main`, declare uma variável local do tipo `ArrayList<Usuario>` e instancie um `ArrayList`
6. Faça um laço de repetição para instanciar 3 objetos da classe `Usuario` e peça para o usuário digitar um email, nome e senha para cada objeto; atribua esses valores às respectivas variáveis de classe `email`, `nome` e `senha` de cada objeto.

7. Faça **outro** laço de repetição para percorrer o `ArrayList` e mostrar na tela os valores das variáveis de classe `email`, `nome` e `senha` de todos os objetos

OBSERVAÇÕES:

- `Scanner` deve ser usado na `main`
 - **NÃO USE** `Scanner` nos métodos da classe `Usuario`
 - `System.out.print` deve ser usado na `main`
 - **NÃO USE** `System.out.print` nos métodos da classe `Usuario`
 - As variáveis de classe de `Usuario` **devem ser atribuídas e acessas usando os métodos da classe `Usuario`**
 - **NÃO ATRIBUA/ACESSE** os valores das variáveis de classe de `Usuario` diretamente (pois as variáveis têm acesso `private`)
-

Atividade 2

1. Crie um projeto Java e dê o nome de `Atividade2`
2. Crie uma classe e dê o nome de `Tarefa`
3. Declare as seguintes variáveis de classe para `Tarefa`:
 1. `titulo`: tipo `String` e acesso `private`
 2. `finalizada`: tipo `boolean` e acesso `private`
4. Defina os seguintes métodos de classe para `Tarefa`:
 1. `atribuirTitulo`: um parâmetro `String` e retorno `void`; recebe um valor do tipo `String` e armazena na variável de classe `titulo`
 2. `atribuirFinalizada`: um parâmetro `boolean` e retorno `void`; recebe um valor do tipo `boolean` e armazena na variável de classe `finalizada`
 3. `pegarTitulo`: sem parâmetros e retorno `String`; apenas retorna o valor da variável de classe `titulo`
 4. `pegarFinalizada`: sem parâmetros e retorno `boolean`; apenas retorna o valor da variável de classe `finalizada`
5. Na `main`, declare uma variável local do tipo `ArrayList<Tarefa>` e instancie um `ArrayList`
6. Faça um laço de repetição para instanciar 3 objetos da classe `Tarefa` e peça para o usuário digitar um título para cada objeto; atribua esse valor à variável de classe `titulo`; atribua o valor `false` à variável `finalizada`; faça isso para cada um dos objetos
7. Faça **outro** laço de repetição para percorrer o `ArrayList` e mostrar na tela os valores das variáveis de classe `titulo`, e `finalizada` de todos os objetos

OBSERVAÇÕES:

- `Scanner` deve ser usado na `main`
- **NÃO USE** `Scanner` nos métodos da classe `Tarefa`
- `System.out.print` deve ser usado na `main`
- **NÃO USE** `System.out.print` nos métodos da classe `Tarefa`
- As variáveis de classe de `Tarefa` **devem ser atribuídas e acessas usando os métodos da classe `Usuario`**

- **NÃO ATRIBUA/ACESSE** os valores das variáveis de classe de `Tarefa` diretamente (pois as variáveis têm acesso `private`)