

Curso online

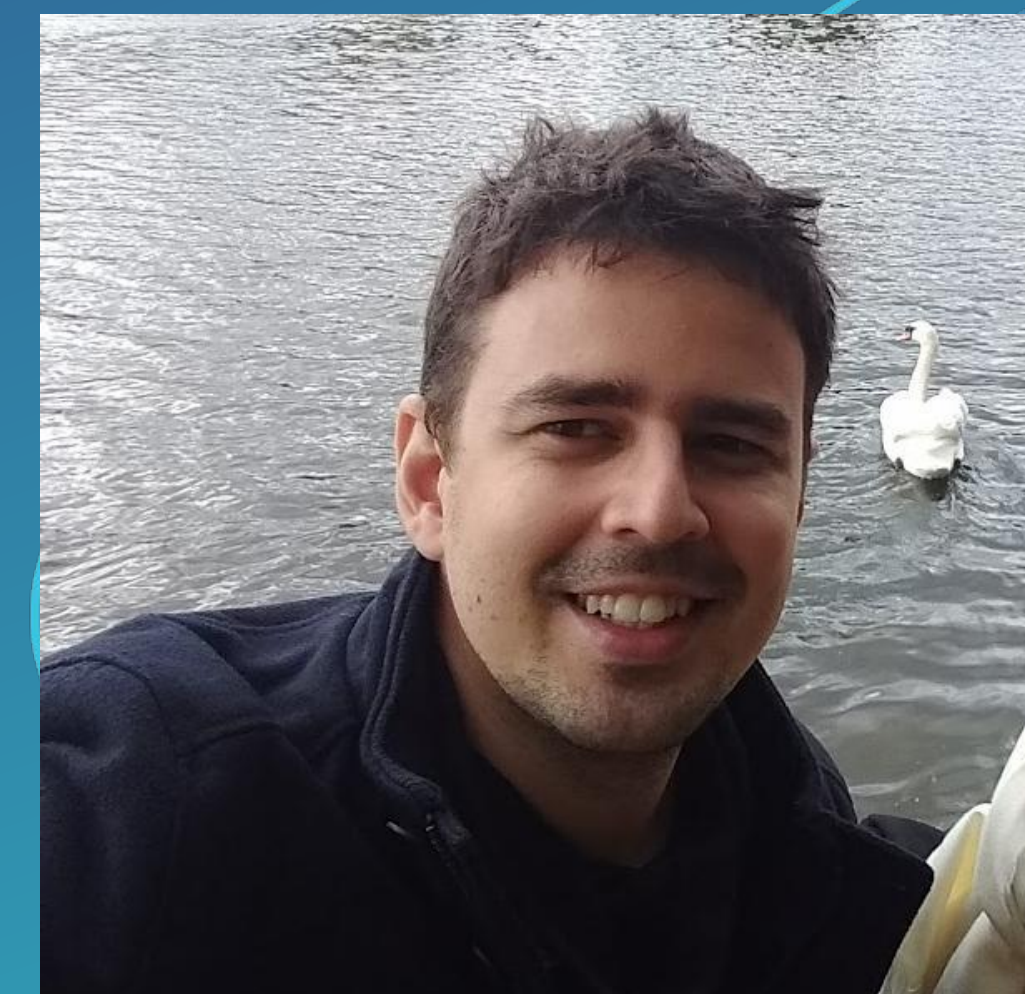
Formação em desenvolvimento WEB

Lógica de Programação

Conteúdo 03 - Classes, Objetos, Atributos e Métodos

Thiago Alves

Doutor



INSTITUTO FEDERAL

Ceará

Polo de Inovação

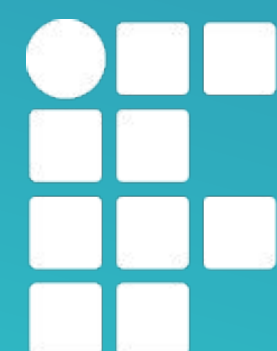


Agenda

- **Introdução**
- **Definição de Classes**
- **Variáveis Locais**
- **Criação de Objetos e chamadas de métodos**
- **Um pouco sobre modificadores de acesso**
- **Definição de Construtores**



Introdução



INSTITUTO FEDERAL

Ceará

Polo de Inovação



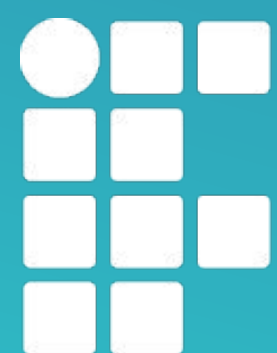
Introdução

Nós já usamos classes, objetos e métodos existentes

```
System.out.print("oi")  
Scanner input = new Scanner(System.in) ;
```

Agora, vamos criar nossas classes e métodos

Classes, atributos e métodos



INSTITUTO FEDERAL
Ceará
Polo de Inovação



Definição de Classe

```
public class Account {  
    private String name;  
  
    public void setName(String name) {  
        this.name = name;  
    }  
  
    public String getName() {  
        return name;  
    }  
}
```

Atributos, Métodos e Acesso

Deve ser armazenada em um arquivo com o mesmo nome da classe e terminando com a extensão .java

Variáveis de instância (atributos) são declaradas dentro da declaração de classe, mas fora dos métodos.

Atributos e métodos com modificador de acesso `private` só podem ser acessados por métodos da classe na qual foram declarados.

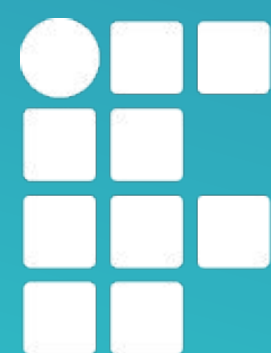
Variáveis Locais

O parâmetro `name` do método `setName` é local ao método

A variável local `name` ofusca o atributo `name` da classe

Usamos o `this.name` para referenciar o atributo `name` da classe

Objetos e chamadas de métodos



INSTITUTO FEDERAL
Ceará
Polo de Inovação



Criação de Objetos e Chamadas de Métodos

```
public class AccountTest {  
    public static void main(String[] args) {  
        Scanner input = new Scanner(System.in);  
        Account myAccount = new Account();  
        System.out.printf("Initial name is: %s%n%n", myAccount.getName());  
        System.out.println("Please enter the name:");  
        String theName = input.nextLine();  
        myAccount.setName(theName);  
        System.out.println();  
        System.out.printf("Name in object myAccount is:%n%s%n", myAccount.getName());  
    }  
}
```

Criação de Objetos e Chamadas de Métodos

O comando `new Account ()` cria um objeto da classe `Account` e atribui à uma variável

Classes no mesmo diretório são importadas implicitamente

Chamada do método `getName` usando o objeto `myAccount`, seguido por um separador ponto (`.`), e depois o nome do método `getName`

Tipos primitivos e tipos referência

```
int numero = 1;
int numero2 = numero;
numero = 2;
System.out.printf("Primerro numero: %d%n", numero);
System.out.printf("Segundo numero: %d%n", numero2);
```

```
Account myAccount = new Account();
myAccount.setName("Java");
Account myAccount2 = myAccount;
myAccount.setName("Thiago");
System.out.printf("Nome em myAccount: %s%n", myAccount.getName());
System.out.printf("Nome em myAccount2: %s%n", myAccount2.getName());
```

Modificadores de Acesso



INSTITUTO FEDERAL

Ceará

Polo de Inovação



Modificadores de Acesso

```
public class AccountModifier {  
  
    public void setAccountName(String name, Account myAccount) {  
        myAccount.name = name; // error  
        //myAccount.name is private  
    }  
  
    public String getAccountName(Account myAccount) {  
        return myAccount.name; // error  
        //myAccount.name is private  
    }  
}
```


Modificadores de Acesso

```
public class AccountModifier {  
  
    public void setAccountName(String name, Account myAccount) {  
        myAccount.setName(name); // OK  
        // method setName is public  
    }  
  
    public String getAccountName(Account myAccount) {  
        return myAccount.getName(); // OK  
        // method getName is public  
    }  
}
```

Construtores



INSTITUTO FEDERAL

Ceará

Polo de Inovação



Construtores

Atributo name é inicializado com null

```
Account myAccount = new Account();  
System.out.printf("Initial name is: %s%n%n", myAccount.getName());
```

E se quisermos fornecer um nome na criação de um objeto da classe?

Construtores

```
public class Account {  
  
    private String name;  
  
    // constructor initializes name with parameter name  
    public Account(String name) {  
        // constructor name is class name  
        this.name = name;  
    }  
    // . . .  
}
```

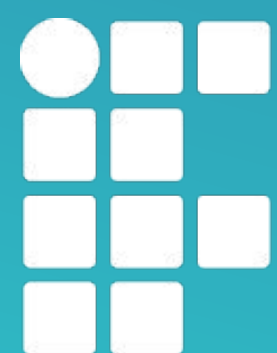
Construtores

```
public class AccountTest {  
    public static void main(String[] args) {  
        Account myAccount = new Account("Thiago Alves");  
        System.out.printf("Initial name is: %s%n%n", myAccount.getName());  
        System.out.println("Please enter the name:");  
        String theName = input.nextLine();  
        myAccount.setName(theName);  
        System.out.println();  
        System.out.printf("Name in object myAccount is:%n%s%n", myAccount.getName());  
    }  
}
```

Construtores

```
public class AccountTest {  
    public static void main(String[] args) {  
        Account myAccount1 = new Account("Thiago Alves");  
        Account myAccount2 = new Account("João");  
        System.out.printf("Initial name is: %s%n%n", myAccount1.getName());  
        System.out.printf("Initial name is: %s%n%n", myAccount2.getName());  
        System.out.println("Please enter the name:");  
        String theName = input.nextLine();  
        myAccount2.setName(theName);  
        System.out.println();  
        System.out.printf("Name in object myAccount1 is:%n%s%n", myAccount1.getName());  
        System.out.printf("Name in object myAccount2 is:%n%s%n", myAccount2.getName());  
    }  
}
```


Exercícios



INSTITUTO FEDERAL

Ceará

Polo de Inovação



Exercício

Adaptar a classe `Account` para armazenar o saldo da conta. Sua classe deve ter um construtor que recebe um nome e um saldo inicial e o construtor deve validar se o saldo inicial é maior ou igual a 0. Caso o saldo inicial do construtor seja negativo, inicializar o saldo com 0. Você também deve criar um método para fazer um depósito de forma que o depósito só seja efetivado se for válido, ou seja, se o depósito for maior que 0. Além disso, você deve criar um método `get` para recuperar o saldo. Em seguida, crie uma classe como programa principal para usar a classe `Account` com valores válidos e inválidos de saldo.

Exercício

Modifique a classe `Account` para fornecer um método chamado `saque` para retirar dinheiro de uma conta. Garante que a quantidade a ser retirada não exceda o saldo da conta. Se exceder, o saldo deve permanecer inalterado e o método deve mostrar uma mensagem "Valor de saque excede o saldo da conta". Cria uma classe com um método `main` para testar o seu método `saque`.

Exercício

Crie uma classe `Employee` que inclui três atributos: um primeiro nome (`String`), um último nome (`String`) e um salário mensal (`double`). Forneça um construtor que inicializa os três atributos. Forneça métodos `set` e `get` para cada atributo. Se o salário mensal não for positivo, não inicialize o seu valor. Escreva um programa `EmployeeTest` que demonstra as capacidades da classe `Employee`. Crie dois objetos `Employee` e mostre o salário anual de cada objeto. Depois dê a cada empregado um aumento de 10% e mostre o salário anual dos empregados novamente.

Exercício

Construa uma classe chamada Date que inclui três atributos: mes (int), dia (int) e ano (int). Forneça um construtor que inicializa os três atributos e assume que os valores fornecidos são corretos. Forneça métodos set e get para cada atributo. Forneça um método displayDate que mostra o dia, mês e ano separados por barra (/). Escreva um programa principal para demonstrar as características da classe Date

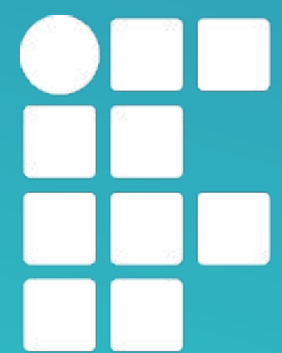
Exercício

Construa uma classe chamada Date que inclui três atributos: mes (int), dia (int) e ano (int). Forneça um construtor que inicializa os três atributos e assume que os valores fornecidos são corretos. Forneça métodos set e get para cada atributo. Forneça um método displayDate que mostra o dia, mês e ano separados por barra (/). Escreva um programa principal para demonstrar as características da classe Date



Obrigado

Professor
Thiago



INSTITUTO FEDERAL
Ceará
Polo de Inovação

