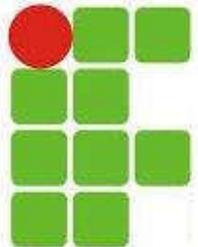
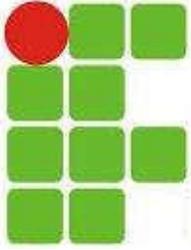

ALGORITMOS

Professor: Diego Oliveira



**Aula 07 -
Constantes, Variáveis e Memória**





Constantes

- Constantes são espaços de memória onde se pode armazenar um valor, de um tipo, que não será alterado ao longo do código
- Utilizam-se constantes para números conhecidos fixos, evitando assim os erros
- Uma constante é criada utilizando-se a palavra-chave 'final'
- O valor de uma constante só pode ser atribuído uma única vez!



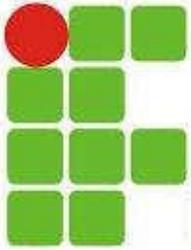


Constantes

- No exemplo abaixo uma constante PI é criada, depois o seu valor é impresso e após isso tenta-se alterar o valor de PI para um número mais preciso, resultando em um erro:

```
public class Constantes {  
    public static void main(String[] args) {  
        final double PI = 3.14;  
        System.out.println("O valor de PI é " + PI);  
        PI = 3.1415;  
    }  
}
```





Variáveis

- Como vimos na aula passada, as variáveis possuem um tipo e um valor e podem ter esse valor alterado a qualquer momento no código:

```
public class Variaveis {  
    public static void main(String[] args) {  
        int idade = 30;  
        System.out.println("Idade = " + idade);  
        idade = 31;  
        System.out.println("Idade = " + idade);  
    }  
}
```





Variáveis

- O valor da variável pode ser alterado, porém seu tipo NUNCA poderá ser pois Java é uma linguagem fortemente tipada:

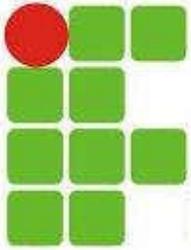
```
public class Variaveis {  
    public static void main(String[] args) {  
        int idade = 30;  
        System.out.print(idade);  
        idade = "trinta";  
        System.out.println("Idade = " + idade);  
    }  
}
```

incompatible types: String cannot be converted to int

(Alt-Enter shows hints)



- Acima a idade foi criada como inteiro e tentou-se colocar uma String, gerando erro

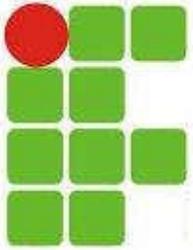


Memória

- Cada tipo de variável ocupa um valor na memória RAM e tem um range de valores:

Tipo	Início	Fim	Memória
boolean	true	false	1 bit
byte	-128	127	8 bits
short	-32768	32767	16 bits
int	-2147483648	2147483647	32 bits
long	-9223372036854775808L	9223372036854775807L	64 bits
float	1.40129846432481707e-45	3.40282346638528860e+38	32 bits
double	4.94065645841246544e-324D	1.79769313486231570e+308D	64 bits
char	0	65535	16 bits





Ler Valores do Teclado

- Para ler valores digitados pelo usuário vamos utilizar Scanner:

```
4 import java.util.Scanner;
5 public class LerTeclado {
6     public static void main(String[] args) {
7         Scanner leia = new Scanner(System.in);
8         int idade;
9         System.out.print("Digite sua idade: ");
10        idade = leia.nextInt();
11        System.out.println("Sua idade é " + idade);
12    }
13 }
```



- Na linha 4 importamos a classe Scanner, que fará a leitura do teclado

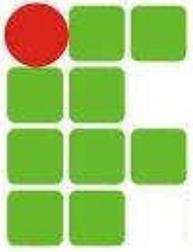


Ler Valores do Teclado

- Ao importar um código, significa que ele foi criado por outra pessoa e será utilizado dentro do seu código.
- Na linha 7 está sendo criado um objeto da classe Scanner, para que possamos fazer a leitura dos dados
- **Mais detalhes sobre importação e criação de objetos serão vistos nas aulas da disciplina de Programação Orientada a**



Objetos

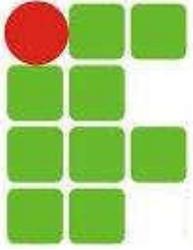


Ler Valores do Teclado

- Exemplo com vários tipos:

```
Scanner leia = new Scanner(System.in);
int idade;
String nome;
double peso;
boolean isAluno;
char genero;
System.out.print("Digite sua idade: ");
idade = leia.nextInt();
System.out.print("Digite seu nome: ");
nome = leia.nextLine();
System.out.print("Digite seu peso (ex. 70.5): ");
peso = leia.nextDouble();
System.out.print("Você é aluno? (true or false) ");
isAluno = leia.nextBoolean();
System.out.print("Digite seu genero (m ou f):");
genero = leia.next().charAt(0); //só primeira letra 9
```





Exercício

- Escreva um código que leia do teclado o valor do RAIO de um círculo e calcule:
 - Diâmetro
 - Perímetro
 - Área

Utilize $\text{PI} = 3.14$ como uma CONSTANTE
- Leia também o valor da Aresta de um cubo e calcule:
 - Área da Superfície
 - Volume
 - Diagonal ($a * \sqrt{3}$)

Utilize $\sqrt{3} = 1.73$ como uma CONSTANTE





Perguntas?

