

FUNDAÇÃO INSTITUTO DE EDUCAÇÃO DE BARUERI

Unidade “Brasílio Flores de Azevedo”

PDMO (Programação para Aplicativos)

Aula01-3 – for() / while{ }

Prof Leomar Duarte

### Estrutura de Repetição

Os laços de repetição (looping) permitem a repetição da execução de um bloco de instruções em um programa. As estruturas utilizadas no java são **for** e **while**.

### Estrutura for

```
for (inicialização; condição; incremento ou decremento) {  
    <Instruções>  
}
```

- **Inicialização:** Valor inicial da variável de controle do laço
- **Condição:** Contém uma expressão booleana (verdadeiro(true) ou falso (false)), que será usada para controlar a continuidade do laço.
- **Incremento ou Decremento:** Variável de controle que será acrescentada ou subtraída.

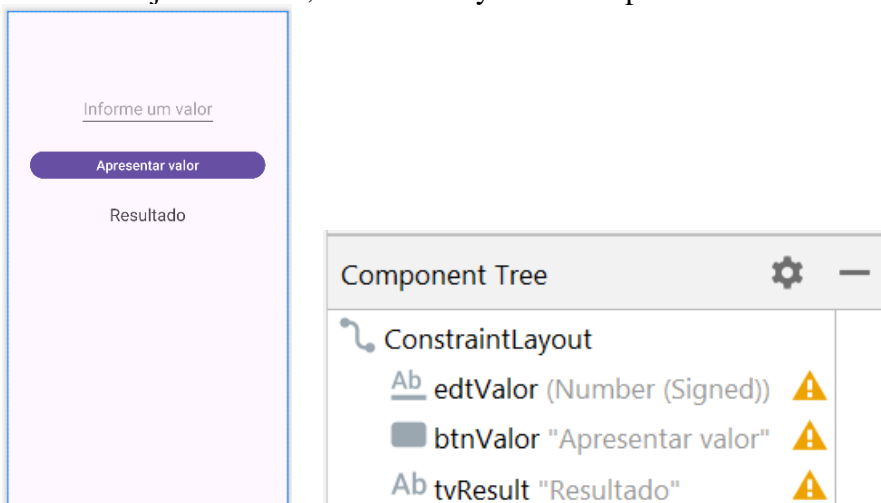
Exemplo:

//Início do cont em 1, onde o laço é executado 10 vezes enquanto a condição for verdadeira

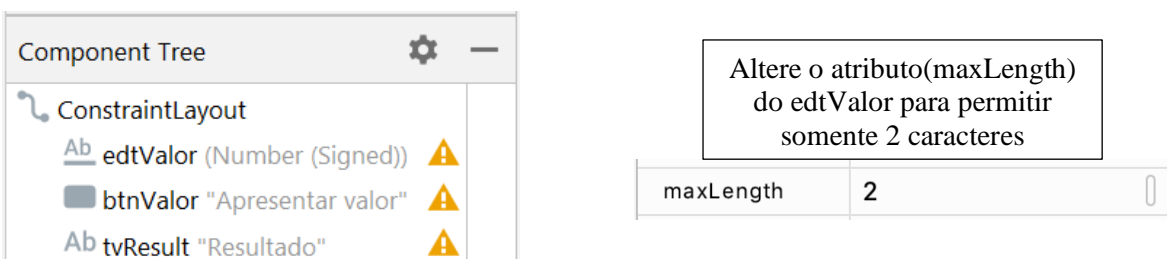
```
for (int cont=1; cont<=10; cont++) {  
    <Instruções>  
}
```

### Passos para o LayOut

- 1) Crie um novo projeto em branco em Android com o nome: Testando\_Repeticao
- 2) Apagar o Hello World
- 3) Insira os objetos abaixo, conforme layout e Component Tree:



- 4)
- 5) Seguem os IDs e tipos dos objetos:



- 6)

## Passos para o código Java

- 1) Clique em  MainActivity.java
- 2) Crie as variáveis para cada objeto da Activity anterior, digitando abaixo da class:

```
public class MainActivity extends AppCompatActivity {
```

5 usages

```
    EditText edtValor;
```

2 usages

```
    Button btnValor;
```

3 usages

```
    TextView tvResult;
```

4 usages

```
    String aux;
```

- 3)
- 4) Faça o link entre as variáveis acima com os IDs de cada objeto que está na Activity, utilize o método para inicializar após o onCreate...

@Override

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);
```

```
    inicializar();
```

- 5)
- 6) Crie o método abaixo de tudo, porém antes da ultima chave:

```
private void inicializar() {  
    edtValor = findViewById(R.id.edtValor);  
    btnValor = findViewById(R.id.btnValor);  
    tvResult = findViewById(R.id.tvResult);  
  
    aux = "";  
    tvResult.setText("");  
    edtValor.requestFocus();  
}
```

- 7) }


- 8) Incluir o código dentro do try e catch abaixo. Haverá erro(em vermelho) no alert() e contagem(), porque não existem. Serão explicados abaixo:

```
btnValor.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        try {  
            int valorInformado = Integer.parseInt(edtValor.getText().toString());  
            if (valorInformado > 10) {  
                alert("Informe valor até 10");  
                inicializar();  
            } else {  
                contagem(valorInformado);  
                aux = "";  
            }  
        } catch (Exception e) {  
            alert("Informe um valor, não deixe em branco!!!");  
            inicializar();  
        }  
    }  
}
```

- 9) }

10) Clique no método alert até que seja o oferecido a lâmpada em vermelho, clique em Create method...



11) 

12) Selecione a opção: 

13) Será criado o método acima da última chave do seu app:

```
private void alert(String s) {  
}
```

14)

15) Digite o código entre as chaves do método alert(String s) como segue:

```
private void alert(String s) {  
    edtValor.setError(s);  
    inicializar();  
}
```

16)

17) Faça o mesmo processo para o método contagem(qtde).

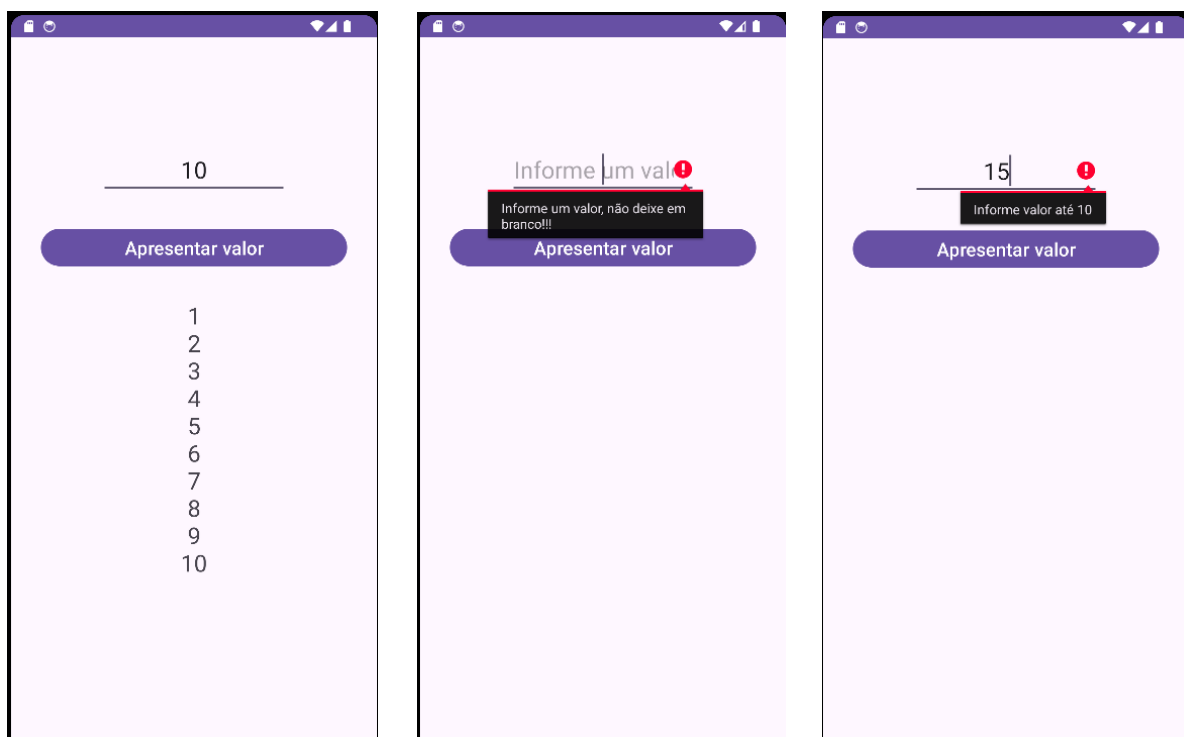
18) Insira os códigos abaixo:

```
private void contagem(int valorInformado) {  
    for (int cont = 1; cont <= valorInformado; cont++) {  
        aux += cont + "\n";  
    }  
    tvResult.setText(aux);  
    edtValor.requestFocus();  
}
```

19)

20) Execute o App, teste e analise o código para entendimento.

### Segue o Código Completo e Proposta da Tela



## Código Completo

```
public class MainActivity extends AppCompatActivity {  
    5 usages  
    EditText edtValor;  
    2 usages  
    Button btnValor;  
    3 usages  
    TextView tvResult;  
    4 usages  
    String aux;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        inicializar();  
    }  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        inicializar();  
  
        btnValor.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View view) {  
                try {  
                    int valorInformado = Integer.parseInt(edtValor.getText().toString());  
                    if (valorInformado > 10) {  
                        alert(s: "Informe valor até 10");  
                        inicializar();  
                    } else {  
                        contagem(valorInformado);  
                        aux = "";  
                    }  
                } catch (Exception e) {  
                    alert(s: "Informe um valor, não deixe em branco!!!");  
                    inicializar();  
                }  
            }  
        });  
    }  
}
```

```

private void contagem(int valorInformado) {
    for (int cont = 1; cont <= valorInformado; cont++) {
        aux += cont + "\n";
    }
    tvResult.setText(aux);
    edtValor.requestFocus();
}

2 usages
private void alert(String s) {
    edtValor.setError(s);
    inicializar();
}

4 usages
private void inicializar() {
    edtValor = findViewById(R.id.edtValor);
    btnValor = findViewById(R.id.btnValor);
    tvResult = findViewById(R.id.tvResult);

    aux = "";
    tvResult.setText("");
    edtValor.requestFocus();
}
}

```

## Repetição While

### Estrutura while

Igual ao laço for, porém avalia o resultado da expressão (condição, da mesma forma que o if) antes de executar as instruções do bloco {}, assim é possível que as instruções nunca sejam executadas, caso a condição seja inicialmente falsa.

```

while (condição) {
    <Instruções>
}

```

## Método com Retorno

- 1) Vamos criar outro método igual ao contagem() porém com retorno de valor;
- 2) Digite o código e o método a seguir =>contagemComRetorno(), abaixo do contagem(valorInformado).
- 3) Coloque // para comentar a contagem, porque não será utilizado.

```

} else {
    //contagem(valorInformado);
    tvResult.setText(contagemComRetorno(valorInformado));
}
4)
5) Complete o método:

```

Teste, Análise e Estude o Código.  
 Verifique se entendeu o código.  
 Faça alterações para entendimentos.

## **Resumo**

### **Método com retorno:**

Executa e retorna o resultado, onde você deverá utilizá-lo no programa principal, neste caso foi o `tvResult.setText`.

### **Método sem retorno (void),**

Não tem retorno de valores, deverá ser capaz de executar toda a tarefa solicitada no método.