

Estruturas de Repetição – **While/Do**

Para utilizar a Função While/Do desenvolveremos um **DADO** Eletrônico.

Para desenvolver este Dado Eletrônico, será necessário:

Criar um Variável do Tipo char.

```
14 public class Teste2 {  
15  
16     /**  
17      * @param args the command line arguments  
18      */  
19     public static void main(String[] args) {  
20         // TODO code application logic here  
21  
22         //Início do Jogo  
23         char opcao;  
24         System.out.println("DADO ELETRÔNICO");
```

E uma Estrutura de Repetição. O código dentro desta Estrutura de Repetição será impresso e depois irá se repetir enquanto o teste retornar Verdadeiro.

```
do {  


Código

  
} while ( 

Teste

 );
```

Neste primeiro momento, temos o seguinte código:

```
14 public class Teste2 {
15
16     /**
17      * @param args the command line argument:
18      */
19     public static void main(String[] args) {
20         // TODO code application logic here
21
22         //Início do Jogo
23         char opcao;
24         System.out.println("DADO ELETRÔNICO");
25
26         System.out.println(" ");
27
28         do {
29
30
31
32
33
34
35
36
37     } while (
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

Código

Teste

Dentro da Estrutura de Repetição, crie uma Variável Inteira que receberá um número Aleatório de 0 a 5.

Em “*6” percebemos que serão impressos 6 dígitos, ou seja, 0, 1, 2, 3, 4 e 5; nos criando um problema, pois um Dado possui número de 1 a 6.

```
do {
    int dado = (int) (Math.random() * 6); //Gera um número aleatório de 0 a 5
} while (
    Teste
);
```

Para corrigir essa falha, realizaremos um Pré-Incremento, ou seja, incrementaremos o valor da Variável em uma unidade antes de realizar a impressão. Este primeiro valor é impresso assim que o programa é executado.

```
do {  
    int dado = (int) (Math.random() * 6); //Gera um número aleatório de 0 a 5  
    System.out.println("Número Sorteado: " + ++dado); //Incrementa o número a.  
  
} while ( Teste );
```

Contudo, para que o jogo continue (Isso que dizer, que o Dado seja lançado e verificado o novo valor sorteado.) é necessário que o usuário (Jogador) pressione a tecla “s”+Enter para confirmar a ação.

O bom funcionamento da interação com o usuário (Jogador) irá depender de duas linhas, sendo: uma linha para realizar a leitura do teclado `Scanner opcao_digitada = new Scanner(System.in);` e outra linha para fazer a Variável “opção” receber o valor inserido pelo teclado convertido em uma Variável do Tipo char `opcao = opcao_digitada.next().charAt(0);`.

```
do {  
    int dado = (int) (Math.random() * 6); //Gera um número aleatório de 0 a 5  
    System.out.println("Número Sorteado: " + ++dado); //Incrementa o número aleatório em 1 e imprime  
  
    System.out.println(" ");  
  
    System.out.print("Desejar lançar o dado (s/n)?");  
    Scanner opcao_digitada = new Scanner(System.in); //Realiza a leitura do teclado  
    opcao = opcao_digitada.next().charAt(0); //next().charAt(0) faz com que somente a primeira letra digitada seja lida  
} while ( Teste );
```

Ao inserir `Scanner opcao_digitada = new Scanner(System.in);`, será necessário realizar o `import java.util.Scanner;`, clicando na lâmpada de dica do NetBeans.

Por fim, devemos inserir o Teste, onde a execução desta Repetição continuará até que uma tecla diferente de “s” seja digitada.

```
do {  
    int dado = (int) (Math.random() * 6); //Gera um número aleatório de 0 a 5  
    System.out.println("Número Sorteado: " + ++dado); //Incrementa o número aleatório em 1 e imprime  
  
    System.out.println(" ");  
  
    System.out.print("Desejar lançar o dado (s/n)?");  
    Scanner opcao_digitada = new Scanner(System.in); //Realiza a leitura do teclado  
    opcao = opcao_digitada.next().charAt(0); //next().charAt(0) faz com que somente a primeira letra digitada seja lida  
} while (opcao == 's');
```

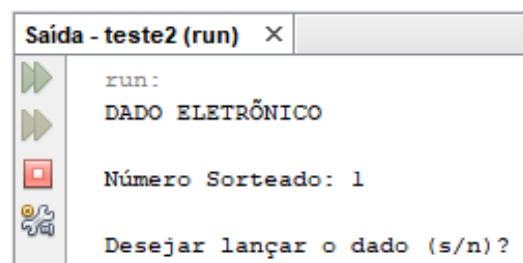
E, somente, para ficar mais apreciável, inserimos a impressão de “Fim de Jogo” e pulamos algumas linhas.



```
6 package teste2;
7
8 import java.util.Scanner;
9
10 /**
11  *
12  * @author Ricardo
13  */
14 public class Teste2 {
15
16     /**
17      * @param args the command line arguments
18      */
19     public static void main(String[] args) {
20         // TODO code application logic here
21
22         //Início do Jogo
23         char opcao;
24         System.out.println("DADO ELETRÔNICO");
25
26         System.out.println(" ");
27
28         do {
29             int dado = (int) (Math.random() * 6); //Gera um nú
30             System.out.println("Número Sorteado: " + ++dado);
31
32             System.out.println(" ");
33
34             System.out.print("Desejar lançar o dado (s/n)?");
35             Scanner opcao_digitada = new Scanner(System.in);
36             opcao = opcao_digitada.next().charAt(0); //next().c
37         } while (opcao == 's');
38
39         System.out.println("Fim de Jogos");
40     }
41
42 }
```

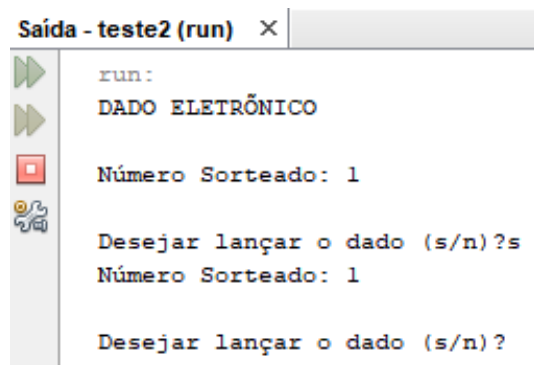
Execute o programa e realize os testes.

Neste primeiro momento, o Dado é lançado e o programa retorna o número sorteado. Abaixo, exibe o diálogo para que o usuário informe se deseja lançar o Dado novamente.



```
Saída - teste2 (run) X
run:
DADO ELETRÔNICO
Número Sorteado: 1
Desejar lançar o dado (s/n)?
```

Inserindo a letra “s”, o Dado será lançado mais uma vez e conforme anteriormente, informará o número sorteado e exibirá o diálogo se deverá lançar o Dado novamente.



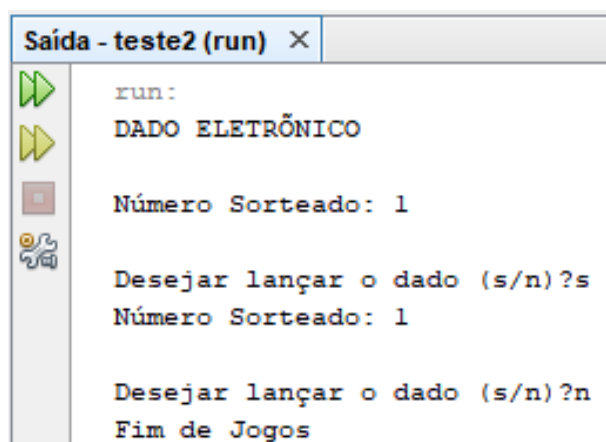
```
run:
DADO ELETRÔNICO

Número Sorteado: 1

Desejar lançar o dado (s/n)?s
Número Sorteado: 1

Desejar lançar o dado (s/n)?
```

Caso o usuário pressione a tecla “n” ou qualquer outra tecla, o jogo será encerrado.



```
run:
DADO ELETRÔNICO

Número Sorteado: 1

Desejar lançar o dado (s/n)?s
Número Sorteado: 1

Desejar lançar o dado (s/n)?n
Fim de Jogos
```