

Mão na massa: Praticando char e String

Dessa vez, vamos praticar um pouco do uso dos tipos **char** e **String**.

1) Crie uma classe e estruture seu método main, na aula utilizamos o nome "**TestaCaracteres**"

2) Crie uma variável do tipo **char**, como fizemos abaixo:

```
public class TestaCaracteres{  
  
    public static void main(String[] args){  
  
        char letra = 'a';  
    }  
}
```

3) Imprima a variável no console:

```
public class TestaCaracteres{  
  
    public static void main(String[] args){  
  
        char letra = 'a';  
        System.out.println(letra);  
    }  
}
```

Nossa saída será: **a**

4) Como falado no vídeo, o tipo **char** guarda apenas um caractere da tabela **unicode**, por isso, quando passamos um valor numérico, teremos como saída o caractere correspondente ao número na tabela **unicode**. Escreva o seguinte código e veja:

```
public class TestaCaracteres{  
  
    public static void main(String[] args){  
  
        char letra = 'a';  
        System.out.println(letra);  
  
        char valor = 65; // Compila!  
        System.out.println(valor);  
    }  
}
```

A saída do nosso segundo `System.out.println` será: **A**

5) Para um segundo teste, escreva o seguinte:

```
public class TestaCaracteres{

    public static void main(String[] args){

        char letra = 'a';
        System.out.println(letra);

        char valor = 65;           // Compila!
        System.out.println(valor);

        valor = valor + 1;        // Não compila!
        System.out.println(valor);
    }
}
```

Nosso código não compila, quando escrevemos essa soma, o Java automaticamente transforma o resultado para o **maior** tipo, que nesse caso é **int**, já que o número **1** é um inteiro.

6) Faça um casting para que esse código compile, para isso devemos escrever:

```
public class TestaCaracteres{

    public static void main(String[] args){

        char letra = 'a';
        System.out.println(letra);

        char valor = 65;           // Compila!
        System.out.println(valor);

        valor = (char) (valor + 1); // Compila!
        System.out.println(valor);
    }
}
```

Repare que envolvemos a soma com parênteses, dessa forma, dizemos para o Java que queremos transformar o resultado dessa soma em **char**.

7) Crie agora uma variável do tipo **String**, repare que esse tipo é diferente dos que tínhamos usado até o momento, além disso, escrevemos com a inicial maiúscula.

```
public class TestaCaracteres{

    public static void main(String[] args){

        char letra = 'a';
        System.out.println(letra);

        char valor = 65;           // Compila!
        System.out.println(valor);

        valor = (char) (valor + 1); // Compila!
        System.out.println(valor);
    }
}
```

```

        String palavra = "alura cursos online de tecnologia";
    }
}

```

Para **String**, usaremos obrigatóriamente aspas duplas.

8) Imprima a **String** no console. Nossa código terá a seguinte cara:

```

public class TestaCaracteres{

    public static void main(String[] args){

        char letra = 'a';
        System.out.println(letra);

        char valor = 65;           // Compila!
        System.out.println(valor);

        valor = (char) (valor + 1);
        System.out.println(valor);

        String palavra = "alura cursos online de tecnologia";
        System.out.println(palavra);
    }
}

```

9) Concatene sua **String** com o número **2020** da seguinte forma:

```

public class TestaCaracteres{

    public static void main(String[] args){

        char letra = 'a';
        System.out.println(letra);

        char valor = 65;           // Compila!
        System.out.println(valor);

        valor = (char) (valor + 1); // Compila!
        System.out.println(valor);

        String palavra = "alura cursos online de tecnologia";
        System.out.println(palavra);

        palavra = palavra + 2020;
        System.out.println(palavra);
    }
}

```

Repare que o número 2020 aparece no final da frase, isso acontece porque que o Java converte automaticamente o número para **String**, diferente do que acontecia com os tipos primitivos que vimos anteriormente.

