

Documento

Documentação linha por linha para o código em Java:

```
import java.util.Scanner;
```

Esta linha importa a classe `Scanner` da biblioteca padrão do Java. A `Scanner` é usada para receber entradas do usuário.

```
public class Main {
```

Define uma classe pública chamada `Main`, que é a classe principal do programa. No Java, o código precisa estar dentro de uma classe para ser executado.

```
    public static void main(String[] args) {
```

O método `main` é o ponto de entrada do programa. O Java executa este método primeiro quando o programa é iniciado. O parâmetro `String[] args` permite que argumentos de linha de comando sejam passados para o programa, embora não sejam usados neste código.

```
        Scanner scanner = new Scanner(System.in);
```

Aqui, criamos um objeto `Scanner` chamado `scanner`, que vai ler a entrada do usuário a partir do console (`System.in`).

```
        System.out.print("Digite uma palavra: ");
```

Imprime a mensagem `"Digite uma palavra: "` no console, solicitando que o usuário insira uma palavra.

```
        String palavra = scanner.nextLine();
```

Lê a próxima linha de entrada do usuário e a armazena em uma variável do tipo `String` chamada `palavra`.

```
palavra = palavra.toLowerCase();
```

Converte todos os caracteres da `palavra` para letras minúsculas, para que a criptografia seja case-insensitive (não depende de maiúsculas ou minúsculas).

```
StringBuilder resultado = new StringBuilder();
```

Inicializa um objeto `StringBuilder` chamado `resultado`. O `StringBuilder` é usado para construir uma string de forma eficiente, especialmente ao adicionar caracteres de forma iterativa.

```
for (char c : palavra.toCharArray()) {
```

Inicia um loop `for` que percorre cada caractere (`char`) da `palavra`. O método `toCharArray()` converte a string `palavra` em um array de caracteres, permitindo que cada caractere seja processado individualmente.

```
if (Character.isLetter(c)) {
```

Verifica se o caractere `c` é uma letra. Isso evita que caracteres não alfabéticos (como números ou símbolos) sejam processados.

```
int valor = c - 'a' + 1;
```

Converte o caractere `c` em um número correspondente à sua posição no alfabeto. Subtrair `'a'` de `c` calcula a diferença entre o valor ASCII do caractere `c` e o valor ASCII de `'a'`, dando uma posição de 0 a 25. Ao adicionar `1`, o valor final estará entre 1 e 26.

```
resultado.append(valor);
```

Adiciona (`append`) o valor calculado para o caractere `c` ao `StringBuilder` `resultado`.

```
}
```

Fecha o loop `for`, que percorreu todos os caracteres da `palavra`.

```
System.out.println("Criptografia: " + resultado.toString());
```

Imprime a string construída `resultado` no console, precedida do texto `"Criptografia: "`. O método `toString()` é usado para converter o conteúdo do `StringBuilder` em uma `String`.

```
scanner.close();
```

Fecha o objeto `scanner`, liberando o recurso associado (no caso, a entrada do console).

Resumo do funcionamento: Este programa solicita que o usuário digite uma palavra, converte-a em letras minúsculas, e substitui cada letra pela sua posição no alfabeto (ex.: A = 1, B = 2, etc.). O resultado é exibido como uma sequência numérica que representa a "criptografia" da palavra digitada.