**構文エラー**

構文エラーとはあらかじめ定義された構文の誤った使用による例外です

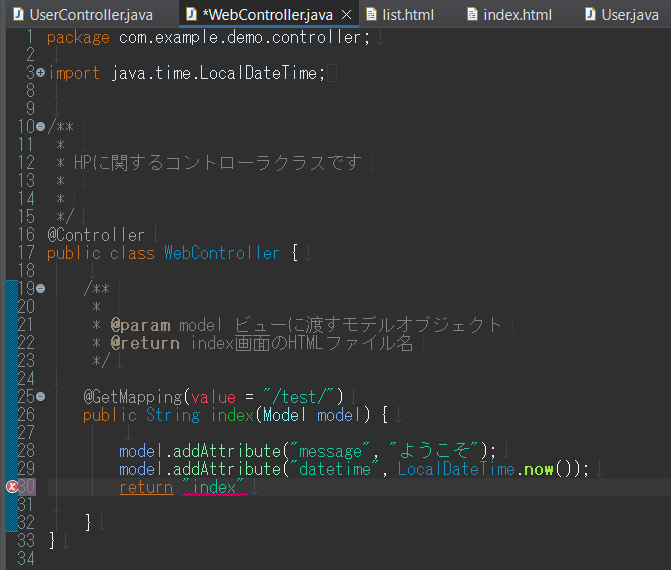
ソースコードのコンパイルまたは、構文解析中に検出されます

分かりやすく書くと「プログラムの書き方がおかしいですよ」エラーです

本来であれば、return "index";　が正しい構文となりますが

ここでは return “index”　の部分に「;」を抜いているので構文エラーとなります。

従ってretun “index”; と治せば解決します



**java.lang.ClassNotFoundException: com.example.web.WebAppApplication**

このエラーコードは com.example.web.WebAppApplicationはクラスが見つからないことを示す

エラーメッセージです。このエラーメッセージは、実行時に指定されたクラスが見つからない場合に

表示されます。

簡単に言うと「存在しないクラスを参照しようとしている」という事になります

作成したクラスを消したり、実行中にクラス名を替えたりすると発生します

ex1

@GetMapping(value = "/ex1")

public String NullPointerException() {

String value = Math.random() < 1 ? null : "a";

System.out.println(value.toLowerCase());

return "";

}

これはNullPointerExceptionを発生させるメソッドです

変数やオブジェクトがnullの状態でメソッドやフィールドにアクセスしようとした場合に発生する例外です

メソッド名はNullPointerException()で

Math.random()が返す乱数が1より小さい場合、変数valueはnullになります

value.toLowerCase()というメソッドの呼び出しにより

valueの参照がnullのままであるため、NullPointerExceptionとなります

具体的に

String value = Math.random() < 1 ? null : "a"　の部分は三項演算子というものを使用した条件式になっていて

三項演算子とは、公式を書くと

「条件式 ? 式１ : 式２」といった形式になっていて

条件式が「true」であれば式１の結果が代入され、条件式が「false」であれば式２が代入されるものです

Math.random()は0以上1未満のランダムな値を返します、この場合は

Math.random() < 1　となっていて常にtrueになっています

つまり、条件付き代入では常に式1の結果が代入されています

条件付き代入の結果が変数valueに代入されています

簡潔にいうとこの部分は常にaを代入するという意味になっています

valueがnullの時にtoLowercaseメソッドを呼び出すとNullPointerExceptionが出ます

なぜそうなるのかというと、nullの状態の文字列に対して、メソッドを呼び出す事はできないからです

ex2

これはNumberFormatExceptionを発生させるコードです

value変数には文字列「a」が代入されています

次にInteger.parseInt(value)を呼び出して文字列を変数に変換しようとしています

代入されているのが「a」なので整数じゃないので例外が起き、NumberFormatExceptionが発生します

文字列ではなく整数にすると解決します、123など

要点は数値ではなく文字列を変数に変換しようとした場合に発生するエラーです

ex3

IndexOutOfBoundsExceptionという例外を発生させています

配列やリストなどの範囲外の要素にアクセスしようとしたら発生します

空のArrayList　オブジェクトを作成

list.get(0)はリストから要素を取得しています、0の要素が存在しない場合IndexOutOfBoundsが出ます

存在しない要素にアクセスしようとして発生するエラーですので

リストに要素を追加して存在する要素にアクセスできるようにします

範囲を見ているIndexOutOfBoundsに対応するため、if文などを追加して範囲内のインデックスのみにアクセスするようにすると解決します