



Webアプリケーション開発演習

コネクションプール

前回までの学習内容

フィルターとは

- ・サーブレットが動く前に動く
- ・3つの決まり

○Filterインターフェースを実装する！

○doFilter、init、destroyの3つのメソッドを実装する！

○@WebFilterをつけてURLとフィルターを関連づける！

- ・フィルターの使用用途

・複数の画面で共通の処理を行う場合

→文字化け対策やログインチェックでよくつかわれる

今日の目標！

コネクションフルを理解する

目次

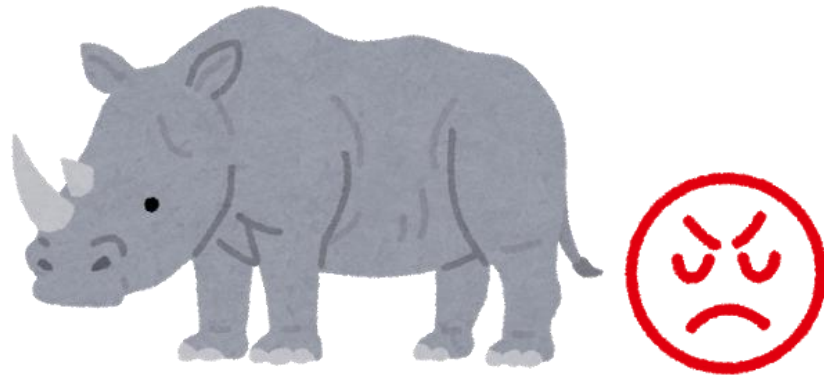
- **DBの接続方法の問題点**
- **コネクションプールとは？**
- **演習**

目次

- **DBの接続方法の問題点**
- コネクションプールとは？
- 演習

DBの接続方法の問題点

DBの接続方法について今まで学んだ方法をまずは再度、復習しましょう。



DBの接続方法の問題点

DBの扱いは↓のようなコードでした

```
////////////////////////////////////↓
```

```
//DBの接続↓
```

```
Class.forName("com.mysql.cj.jdbc.Driver");↓
```

```
con = DriverManager.getConnection(↓
```

```
> > "jdbc:mysql://localhost:3306/webtestdb?characterEncoding=UTF-8&serverTimezone=JST",↓
```

```
> > "root","password");↓
```

接続

```
////////////////////////////////////↓
```

```
//SELECT文の発行↓
```

```
stmt = con.prepareStatement("SELECT * FROM user_tbl");↓
```

```
rs = stmt.executeQuery();↓
```

SQLの組み立て
&実行

```
////////////////////////////////////↓
```

```
//DBから値を取得↓
```

```
while( rs.next() ){↓
```

```
> MemberDto member = new MemberDto();↓
```

```
> member.setMail(rs.getString("mail"));↓
```

```
> member.setName(rs.getString("name"));↓
```

```
> list.add(member);↓
```

```
}↓
```

結果の取得

DBの接続方法の問題点

実はこのコード、ある問題があります

```
////////////////////////////////////↓
//DBの接続↓
Class.forName("com.mysql.cj.jdbc.Driver");↓

con = DriverManager.getConnection(↓
> > "jdbc:mysql://localhost:3306/webtestdb?characterEncoding=UTF-8&serverTimezone=JST",
> > "root","password");↓

////////////////////////////////////↓
//SELECT文の発行↓
stmt = con.prepareStatement("SELECT * FROM user_tbl");↓

rs = stmt.executeQuery();↓

////////////////////////////////////↓
//DBから値を取得↓
while( rs.next() ){↓
> MemberDto member = new MemberDto();↓

> member.setMail(rs.getString("mail"));↓
> member.setName(rs.getString("name"));↓

> list.add(member);↓
}↓
```


DBの接続方法の問題点

DBの処理は

1. DBの接続
2. SQLの組み立て
3. SQLの実行
4. (必要であれば)結果の取得

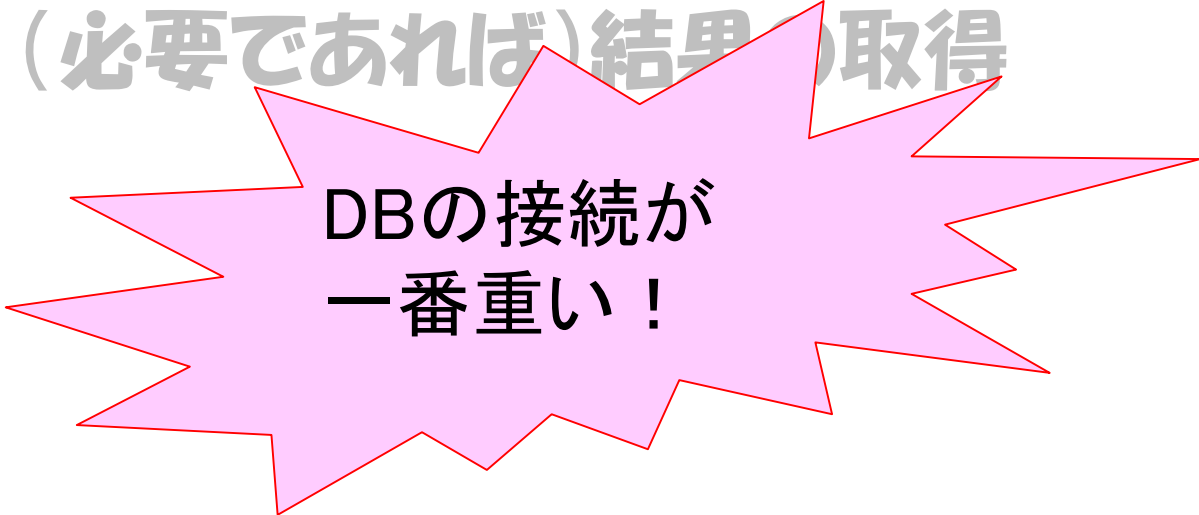


問題です。この処理の中で**一番重たい**
(遅い)処理は何でしょう？

DBの接続方法の問題点

正解は・・・

- 1. DBの接続の**
2. SQLの組み立て
3. SQLの実行
4. (必要であれば)結果の取得

A pink starburst shape with a red outline, containing the text "DBの接続が一番重い！".

**DBの接続が
一番重い！**

DBの接続方法の問題点

**Webアプリケーションでは
DBからの値の取得や設定を何度も
行います。**

**重たい処理を何度も行くと
全体のパフォーマンス(速度)に影響
します。**

これが今の接続方法の問題点

目次

- ・DBの接続方法の問題点
- ・コネクションプールとは？
- ・演習



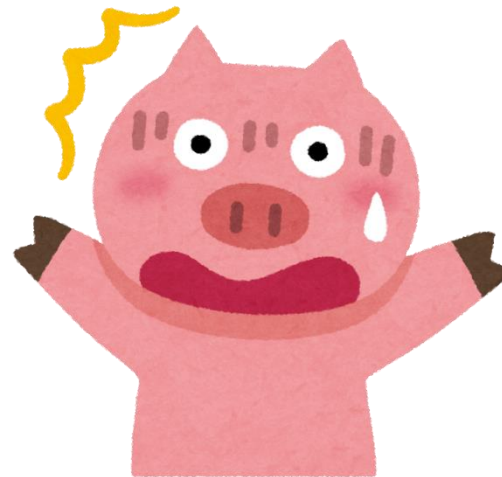
コネクションプールとは？

この問題を解決するのが今回のテーマ

コネクションプール

コネクションプールとは？

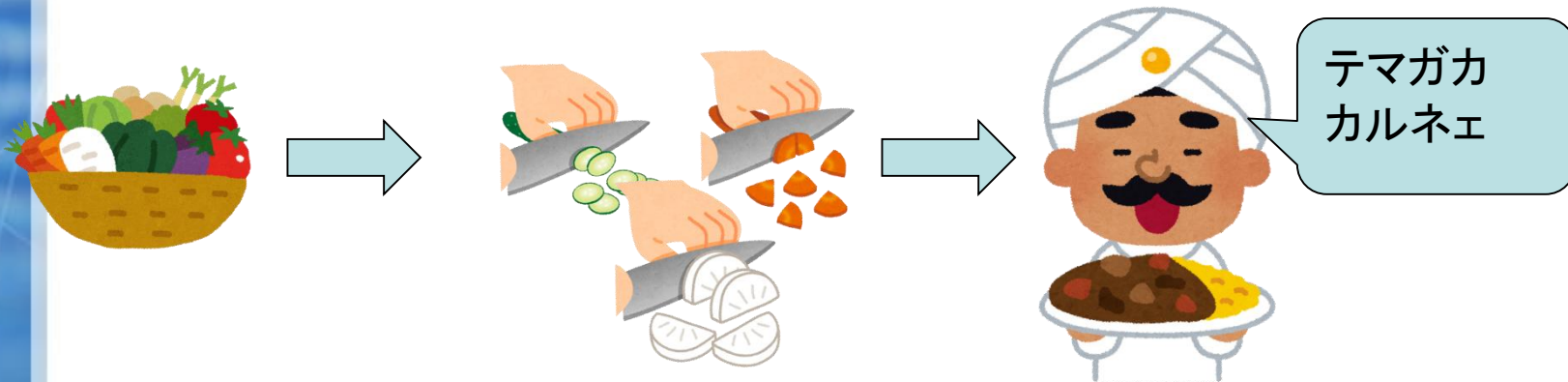
コネクションプールとはとは、データベースにコネクション(接続)する場合に、要求されるたびにコネクションを作成するのではなく、コネクションをプール(保持)して再利用するという機能です。



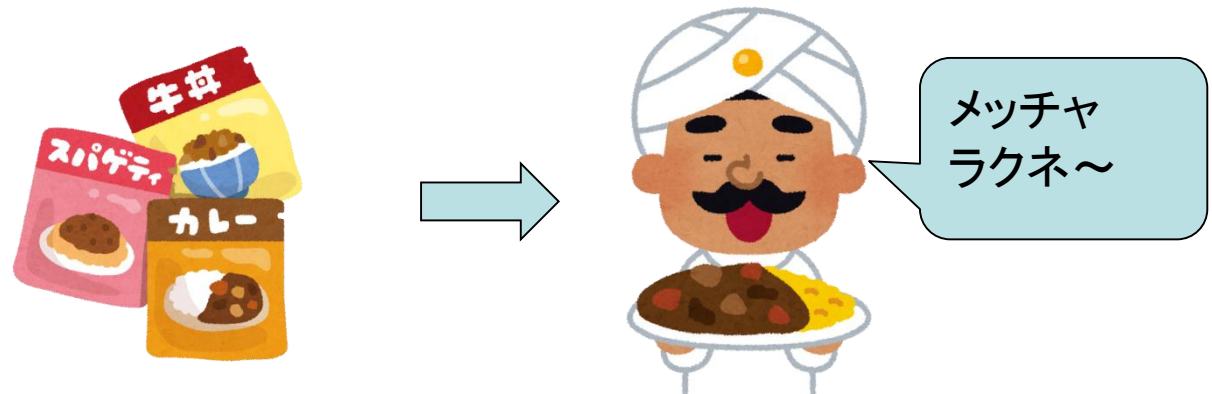
コネクションプールとは？

たとえるならば・・・

今までのやり方は、材料からカレーを作るやり方



コネクションプールは、レトルトパックで作るやり方



コネクションプールとは？

**早速コネクションプールを使った
プログラムを動かしてみよう！**

2019Web/演習/サーブレット演習/
connection_pooling.zip

**ダウンロードして、プロジェクトに
インポートして動かしてみよう**



コネクションプールの説明

**コネクションプールのソースコードを
確認しましょう**



コネクションプールの説明

以前と違う部分は↓の部分ですね

```
String jndi = "java:comp/env/jdbc/MySQL";  
  
context = new InitialContext();  
  
DataSource dataSource = (DataSource) context.lookup(jndi);  
  
con = dataSource.getConnection();
```

この部分を詳しく見ていきましょう

コネクションプールの説明

```
String jndi = "java:comp/env/jdbc/MySQL";  
context = new InitialContext();  
DataSource dataSource = (DataSource) context.lookup(jndi);  
con = dataSource.getConnection();
```

接続文字列がない！

コネクションプールの説明

接続文字列はいったいどこに??



コネクションプールの説明

接続文字列はここですっ！ `context.xml`

```
<?xml version="1.0" encoding="UTF-8"?>
<Context>
  <Resource
    name="jdbc/MySQL"
    auth="Container"
    type="javax.sql.DataSource"
    maxTotal="100"
    maxIdle="20"
    maxWaitMillis="-1"
    username="root"
    password="password"
    driverClassName="com.mysql.jdbc.Driver"
    url="jdbc:mysql://localhost:3306/webtestdb?serverTimezone=JST"
    validationQuery="SELECT 0"
  />
</Context>
```

コネクションプールの説明

コネクションプールでは、context.xmlという設定ファイルに「接続情報」を記述して**JNDI という仕組み**を使って、設定情報をJava側に読み込んで、DBに接続します！

JNDI=

Java Naming and Directory Interface



コネクションプールの説明

**設定値がたくさんありますね。
一応、どんな意味の設定値か見ましょう**



データベース接続の項目

設定値	説明	重要度
name	JNDIリソースの名前を指定。	高
auth	プログラムでログインの手順を書くか、コンテナ（Tomcat）がログインするか指定（通常はContainer）	低
type	JNDIリソースのクラス型、インタフェース型を指定する。	低
maxTotal	同時に割り当てることのできるアクティブな接続の最大数	中
maxIdle	プールに保持しておく最大のコネクション数 0の場合は無制限になる。	中
maxWaitMillis	利用可能なコネクションが存在しないときに待機する最大時間（ミリ秒単位）	中
username	DB接続時のユーザー名を指定	高
password	DB接続時のパスワードを指定	高
driverClassName	接続ドライバのクラス名	中
url	接続URL	高
validationQuery	接続検証用のクエリ	低

コネクションプールの説明

設定値はたくさんありますが 必ず知っておかないといけないのは3つ

```
<?xml version="1.0" encoding="UTF-8">
<Context>
  <Resource
    name="jdbc/MySQL"
    auth="Container"
    type="javax.sql.DataSource"
    maxTotal="100"
    maxIdle="20"
    maxWaitMillis="-1"
    username="root"
    password="password"
    driverClassName="com.mysql.jdbc.Driver"
    url="jdbc:mysql://localhost:3306/webtestdb?serverTimezone=JST"
    validationQuery="SELECT 0"
  />
</Context>
```

この設定の名前。

＜Resource＞タグを複数つくれば設定は何種類も作れる。

ユーザー名とパスワード
DBに接続するために必要

コネクションプールの説明

**先ほどの context.xml の設定値を
Javaのプログラムで読み込んでいるとい
うのを意識して、もういちどプログラムを
見てみましょう！**



コネクションプールの説明

```
String jndi = "java:comp/env/jdbc/MySQL";
```

```
context = new InitialContext();
```

初期化

```
DataSource dataSource = (DataSource) context.lookup(jndi);
```

```
con = dataSource.getConnection();
```

JNDIを使って接続するための設定を探す(lookup)する処理

見つけた設定のプールから接続(Connection)を取得(get)する処理

コネクションプールの説明

```
String jndi = "java:comp/env/jdbc/MySQL";
```

```
context = new InitialContext();
```

```
DataSource dataSource = (DataSource) context.lookup(jndi);
```

```
con = dataSource.getConnection();
```

<Resource>

```
name="jdbc/MySQL"
```

```
auth="Container"
```

```
type="javax.sql.DataSource"
```

```
maxTotal="100"
```

```
maxIdle="20"
```

```
maxWaitMillis="-1"
```

```
username="root"
```

```
password="password"
```

```
driverClassName="com.mysql.jdbc.Driver"
```

lookupで指定する引数の値と
context.xmlのnameで指定した
値は一致する必要がある！



コネクションプールの説明

まとめます

コネクションプールを使ったDB接続方法

1. `context.xml`に接続設定を記述する
2. `java`で`context.xml`で指定した`name`を指定して接続情報を`lookup`(探す)
3. 接続情報(`Connection`)を取得する

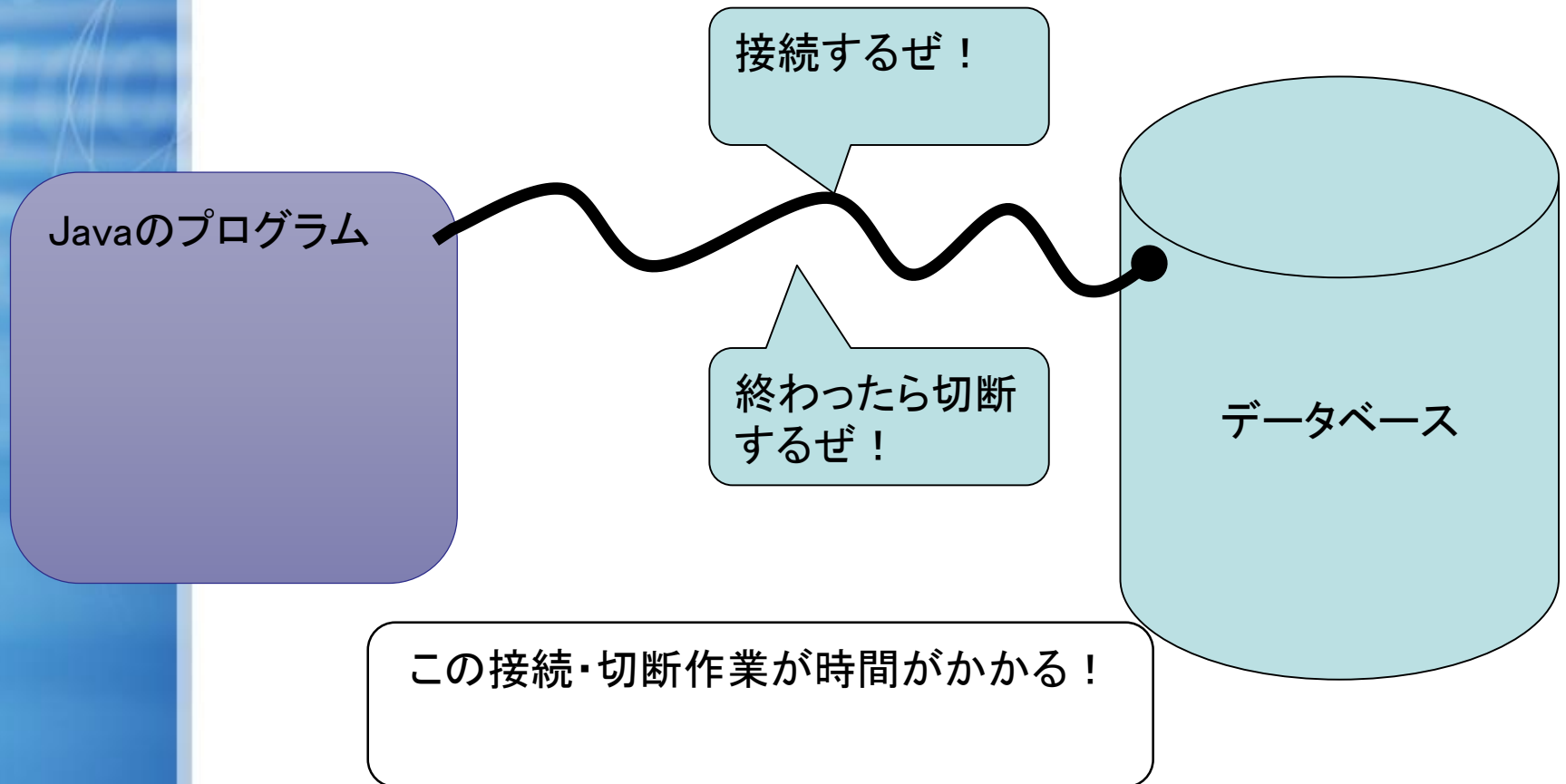
コネクションプールの説明

**理屈が頭に入った状態でもう一度
コネクションプールを絵で説明します。**



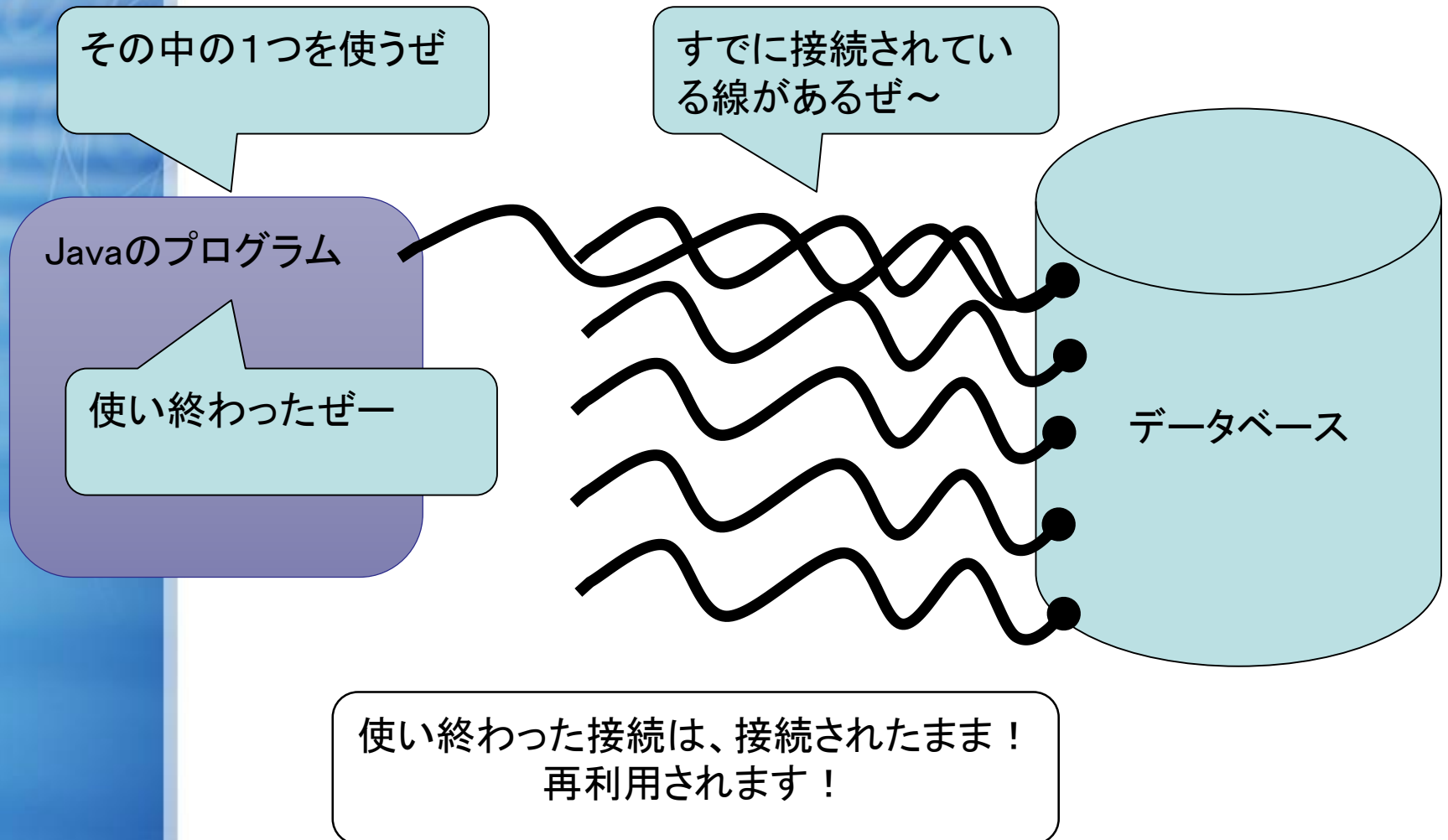
コネクションプールの説明

今までの接続・・・



コネクションプールの説明

コネクションプーリングを使うと...





目次

- ・コネクションフルの説明
- ・**演習**

目次

【演習】

**先日のフィルターの演習の課題を
コネクションプールを使ったプログラムに
書き直してください**

**提出はGoogleClassroom
期限は9月24日中です**