

Chapter 11

データベースプログラミング(2)

以下の指示に従って、クラス、オブジェクトを作成・実行せよ。なお、文中では個別に指示しないが、変更を行ったら適宜コンパイルすること。本課題はレポート提出を求める。レポートには、各設問についてその結果（作成したプログラムの実行結果のスクリーンショット）および考察を含めた説明を記述すること。

締め切りはScombに設定する記述までとする（課題提出欄を確認すること）。Scombにて提出すること（シェアフォルダやメールなど他の方法による提出は原則認めない）。レポートをPDF形式で提出し、加えてソースファイルを提出せよ。ソースファイル名は本資料で指定した名称のまま提出すること。他のファイル名による提出は加点しない。データベースサーバーはakita.data.ise.shibaura-it.ac.jpである。なお、このサーバーは学内でのみアクセス可能である。

11.1 チャットプログラムの作成(データ入力部分編)

次の仕様のチャットプログラム（データ入力部分）をAWTおよびJDBCを使って作成せよ。クラス名をMyChatInputとする。必要なパッケージはimportすることを忘れないように。

- mainメソッドで“チャット（入力）”という名前のフレームを作成し、そこに“発言”ボタン(Buttonクラスのオブジェクトbutton)とTextFieldクラスのオブジェクトtxtFieldを加えること。レイアウトは各自、自由に行ってかまわない。

- “発言”ボタン(button)が押されたら、T_LOGテーブルに次のデータを挿入せよ。ただし、ボタンを押して実行するイベント駆動プログラミングについては以前作成したMyCount.javaのソースを参考にし、データの挿入方法はPreparedStatementクラスを用いることとするため先回の演習のJDBC_PREPARE.javaのソースを参考にせよ。

- データを挿入するためのSQLは

```
INSERT INTO T_LOG VALUES (?, ?, ?, ?)
```

とする。PreparedStatementクラスを用いてデータを挿入する。

- ID列には学籍番号、NAME列には氏名、MESSAGE列はメッセージをそれぞれ文字列としてセットする。
- WRITTENDATEはtimestamp型があるが、これは日時を保持する型である。この列にはデータ挿入時の日付・時間を入れる。PreparedStatementクラスのオブジェクトの名前をprestmtとすると、WRITTENDATE列が4列目であるから、現時刻は

```
prestmt.setTimestamp(4, new Timestamp(System.currentTimeMillis()))
```

とするとセットできる。

- 入力が出来たら、テーブルT_LOGの全データを画面に表示するプログラムを作成し、自分が入力したレコードがデータベースに入力されているかを確認せよ。

ただし、データベースおよびテーブルの設定は以下の通りである。

- データベースアクセスのポート番号は5432である。
- データベース名はfirstdbである。
- データベース接続ユーザーはshibaura、パスワードはtoyosuとする。(ここまで先回と同じ)
- 接続すべきテーブルはT_LOGであり、T_LOGは次のSQLで生成されたものとする。なお、このテーブルは既に作成済みである。作成しようとしなないこと。

```
CREATE TABLE T_LOG(
    ID VARCHAR(10) NOT NULL,
```

```

NAME VARCHAR(30),
MESSAGE VARCHAR(100),
WRITTENDATE TIMESTAMP NOT NULL
)

```

11.2 チャットプログラムの作成(画面表示部分編)

次の仕様のチャットプログラムをAWTおよびJDBCを使って作成せよ。クラス名をMyChatOutputとする。必要なパッケージはimportすること。

- mainメソッドで“チャット（出力）”という名前のフレームを作成し、TextAreaクラスのオブジェクトtxtAreaを加えよ。レイアウトは自由に行ってよい。
- 10秒¹待つごとにtxtAreaの値をクリアし、最新20件のレコードをテーブルT_LOGから取得する。取得したレコードから次のフォーマットの文字列を作成する。1レコードあたりのデータから
 - (2018-07-03 11:30:10.671 (Masaomi Kimura)) 今日は晴れているよ。

という文字列を生成し、この単位で改行を行った20行の文字列をtxtAreaにセットすること。

なお、SQLは SELECT * FROM T_LOG ORDER BY WRITTEN-DATE DESC とすること。このSQLを実行し結果セットを得て上記文字列を作成する際、最大20回だけResultSetクラスのオブジェクトのnextメソッドを呼び出すこと。

- 定期的に処理を実行する場合の処理は以下を参考にすること。

```

import java.util.Timer;
import java.util.TimerTask;

{
    . . .
}

```

¹これを短くするとデータベースの負荷が増加してしまうので注意。

```

    Timer timer = new Timer();
    timer.scheduleAtFixedRate(
        new MyTask(),
        初回のタスク実行までの時間(ms),
        実行するタスクの間隔(ms));
    . . .
}

class MyTask extends TimerTask {
    public void run() {
        // 実行したい処理
    }
}

```

- プログラムを実行し、MyChatInputから入力されたメッセージが正しく含まれるかを確認せよ。