Zarządzanie Informacją 2

Sprawozdanie z Laboratorium 3
Igor Bębenek
Grupa 332

Cały kod:

```java
package org.example;
import java.util.Random;
import java.sql.*;

public class Main {
    public static void main(String[] args) {
        try (Connection conn = DriverManager.getConnection( url: "jdbc:mysql://localhost:3306/lab3",  user: "root",  password: "")) {
            System.out.println("Połączono");

            Statement stmt = conn.createStatement();

            stmt.execute( sql: "CREATE TABLE IF NOT EXISTS tabela ("+ "id INT NOT NULL AUTO_INCREMENT PRIMARY KEY, " + "liczba INT NOT NULL, "
                    + "tekst VARCHAR(255) NOT NULL)");

            stmt.executeUpdate( sql: "DELETE FROM tabela");

            String[] texts = {"Pies", "Kot", "Papuga", "Królik", "Żółw", "Mysz", "Koń", "Słoń", "Lew", "Paw"};
            PreparedStatement ps = conn.prepareStatement( sql: "INSERT INTO tabela (liczba, tekst) VALUES (?, ?)");
            Random random = new Random();

            for (String text : texts) {
                ps.setInt( parameterIndex: 1, random.nextInt( bound: 100));
                ps.setString( parameterIndex: 2, text);
                ps.executeUpdate();
            }
            System.out.println("Wartości z tablicy dodane");

            ResultSet rs = conn.createStatement().executeQuery( sql: "SELECT * FROM tabela");
            while (rs.next()) {
                System.out.println(rs.getInt( columnLabel: "id") + " - " + rs.getInt( columnLabel: "liczba") + " - " + rs.getString( columnLabel: "tekst"));
            }

        } catch (SQLException e) {
            System.err.println("Nie udało się połączyć z bazą danych lub wykonać operacji.");
            System.err.println(e.getMessage());
        }
    }
}
```

1. Połączenie z bazą danych:

```java
try (Connection conn = DriverManager.getConnection( url: "jdbc:mysql://localhost:3306/lab3",  user: "root",  password: "")) {
```

Połączenie z bazą danych za pomocą klasy DriverManager zlokalizowane na localhost, na porcie 3306, nazwa bazy danych lab3, użytkownik root, bez hasła

2. Obsługiwania wyjątku SQLException

```java
} catch (SQLException e) {
    System.err.println("Nie udało się połączyć z bazą danych lub wykonać operacji.");
    System.err.println(e.getMessage());
}
```

Własny komunikat oraz komunikat getMessage()

3. Zapytanie Statement, stworzenie tabeli, stworzenie tablicy, wstawianie rekordów, pobranie danych i wyświetlenie na konsoli

```
10          Statement stmt = conn.createStatement();
11
12          stmt.execute( sql: "CREATE TABLE IF NOT EXISTS tabela ("+ "id INT NOT NULL AUTO_INCREMENT PRIMARY KEY, " + "liczba INT NOT NULL, "
13                  + "tekst VARCHAR(255) NOT NULL)");
14
15          stmt.executeUpdate( sql: "DELETE FROM tabela");
16
17          String[] texts = {"Pies", "Kot", "Papuga", "Królik", "Żółw", "Mysz", "Koń", "Słoń", "Lew", "Paw"};
18          PreparedStatement ps = conn.prepareStatement( sql: "INSERT INTO tabela (liczba, tekst) VALUES (?, ?)");
19          Random random = new Random();
```

W linii 10 tworzymy obiekt stmt interfejsu Statement, następnie za pomocą metody execute tworzymy tabelę z trzema kolumnami: z auto-inkrementującym id, liczba – int oraz tekst typu varchar(255).

W linii 15 wykonujemy polecenie usuwające wszystkie rekordy z tabeli „tabela”.

W linii 17 tworzymy tablicę nazwaną texts typu String w której zawarliśmy różne nazwy, które w przyszłości dodamy do kolumny „tekst”

W kolejnej linii tworzymy obiekt ps typu PreparedStatement za pomocą którego przygotujemy polecenie wstawiające do tabeli rekordy do „liczba” oraz „tekst”.

```
19          Random random = new Random();
20
21          for (String text : texts) {
22              ps.setInt( parameterIndex: 1, random.nextInt( bound: 100));
23              ps.setString( parameterIndex: 2, text);
24              ps.executeUpdate();
25          }
26          System.out.println("Wartości z tablicy dodane");
27
28          ResultSet rs = conn.createStatement().executeQuery( sql: "SELECT * FROM tabela");
29          while (rs.next()) {
30              System.out.println(rs.getInt( columnLabel: "id") + " - " + rs.getInt( columnLabel: "liczba") + " - " + rs.getString( columnLabel: "tekst"));
31          }
32
```

W linii 19 tworzymy obiekt random typu Random, który służy do generowania liczb losowych, które będą wstawiane do kolumny „liczba” w tabeli „tabela”.

Iterujemy przez array „texts”. Za pomocą setInt ustawiamy wartość losową int od 0 do 99, dla drugiego parametru ustawiamy element z arraya „texts” i na końcu pętli wykonujemy metodę executeUpdate(), wykonującą polecenie wstawiające do tabeli.

W linii 28 tworzymy obiekt rs typu ResultSet, tworzymy oraz wykonujemy wyświetlanie całej tabeli „tabela”.

Za pomocą while tworzymy pętlę, która będzie się wykonywać do momentu gdy metoda next() zwróci false, czyli gdy zabraknie wierszy w tabeli „tabela”. Wypisujemy po kolei kolumny id, liczba oraz tekst.

Wyniki z konsoli:

```
C:\Users\igorb\.jdks\openjdk-23\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2024.2.3\lib\idea_rt.jar=53650:C:\Program Files\JetBrains\IntelliJ IDEA 2024.2.3\bin" -Dfile.encoding=UTF-8 -Dsun.stdout.encod
Połączono
Wartości z tablicy dodane
21 - 94 - Pies
22 - 1 - Kot
23 - 56 - Papuga
24 - 43 - Królik
25 - 60 - Żółw
26 - 40 - Mysz
27 - 49 - Koń
28 - 42 - Słoń
29 - 81 - Lew
30 - 98 - Paw

Process finished with exit code 0
```