

Seminar 9.2/Homework 2

Continue Seminar 9

Networking and Databases (Seminar 10)

Bingo server must be made using these two imports:

```
com.esotericsoftware.kryo.Kryo;  
com.esotericsoftware.kryonet.Server;
```

User must be able to buy new variations, fill them in by hand (using console or JoptionPane input) or by random generator and submit to the server. Server then sends back the results for the tickets saying how many right number were for each chosen variation.

Assignment 0 (Create project)

- Open Netbeans or Eclipse
- Create new project "Sem09_kryonet"
- Add "kryonet-2.21-all.jar" to the libraries
- Following class is given already:
 - Variation - responsible for encapsulating all the information about a variation

```
public class Variation {  
    private int variationNr;  
    private int correctNumbers = 0;  
    private ArrayList<Integer> selectedNumbers = new  
ArrayList<>();  
    private String clientEmail;  
    Private Datetime dateTime;  
  
    public Variation(){  
  
    }  
}
```

- Following classes must be created during this seminar
 - MainServer - the main class which hold the Bingo server
 - ClientApp - class for client to choose the variations he wants to sent to the server
 - KryoClient - responsible for client connection
 - KryoClientListener - responsible for listening the events sent to the client

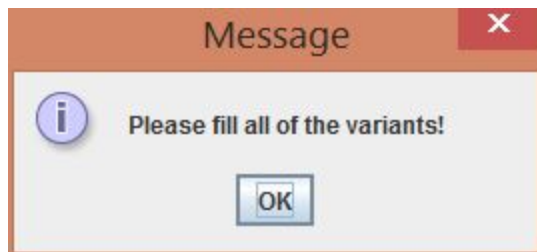
- KryoServer - responsible for server connection
- KryoServerListener - responsible for listening the events sent to the server
- Packet - Message wrapper when sending it across the net
- *Main.java - optional class to Launch the Client side application

Assignment 1 (Variations chooser by client)

- Create an input strategy for variation choose. For example:
 - Ask client to choose how many variants he wants to fill
 - Ask client which variants he wants to fill by own (others will fill randomly)
 - Start to select numbers 1-35 (5 numbers per variant) for each variant. Use console input stream or JOptionPane class for input of numbers (add verifications)
 - Start to generate variants which were checked as “randomly”
 - Submit the list of variants to Server. After that chosen numbers are sent to the server and following message box appears:



- In case where less than 5 numbers are chosen per variant, user must be notified using the message box:



Assignment 2 (Networking #1)

- To send some packets of information, we should create Packet class with subclass of Packet01Message. This class should contain String message. This is done so that the same package can contain different types of sent information (message/warning/...)
- ```
public class Packet {
 public static class Packet01Message {String message;}
}
```
- Create a new class with title “KryoServerListener” which extends Listener
    - Import the necessary libraries:
      - `import com.esotericsoftware.kryonet.Connection;`

- `import com.esotericsoftware.kryonet.Listener;`
- In the constructor generate random numbers used for this session in variable **winningNumbers**;
- Implement connected, disconnected and received functions:
  - `@Override`

```
public void connected(Connection c){...}
```

This should print in console that someone has connected and print the winning numbers

  - `@Override`

```
public void disconnected(Connection c){...}
```

This should print in console that somebody has disconnected

  - `@Override`

```
public void received(Connection c, Object o){...}
```

Server should handle two cases :

  - Received object is of type `Packet.Packet01Message` - print out in console the message;
  - Received object is of type `Variation` - compare the winningNumbers with chosen numbers in the `Variation` object, increase the according field in `Variation` object and finally, print out in console the `Variation` number and how many winning numbers it had.
- Create new class with title “KryoServer”
  - Copy the following lines to include the necessary libraries:
 

```
import com.esotericsoftware.kryo.Kryo;
import com.esotericsoftware.kryonet.Server;
```
  - Constructor must:
    - Create a new server:
 

```
server = new Server();
```
    - Add listener:
 

```
kyl = new kryoServerListener();
server.addListener(kyl);
```
    - Bind to a port:
 

```
server.bind(serverPort);
```
    - Register all the possible packets sent
 

```
registerPackets();
```
    - Start the server
 

```
server.start();
```
  - Create and implement function `registerPackets()` :
 

```
private void registerPackets(){...}
```

In this function you need to get a new Kryo object and register all the Packages that will be sent:

```
Kryo kryo = server.getKryo();
E.g. kryo.register(Packet.Packet01Message.class);
...
```

- In a very similar manner create classes “KryoClientListener” and “KryoClient”.

Differences in KryoClientListener:

```
received(Connection c, Object o){...}
```

Should print a message if server sends one.

In KryoClient:

- Constructor must:
  - Create a new client:  
`client = new Client();`
  - Add listener:  
`kcl = new KryoClientListener();`  
`client.addListener(kcl);`
  - Register all the possible packets sent  
`registerPackets();`
  - Start the client  
`client.start();`
  - Connect to the correct ip address and port  
`client.connect(500, ip, serverPort);`
- Send all the filled variations to the server:  
`client.sendTCP(...)`

## Assignment 3 (Networking #2)

- Implement functionality that allows to submit the chosen numbers to a server
- Submitted variations from client are saved in database (see Assignment 3)
- Each hour server generates lucky numbers and compare with numbers which are saved in database during this hour
- Server sends a response to a client and outputs the correctly guessed numbers in console (if client is still connected). If client is disconnected - send response to client email

## Assignment 3 (Database #1)

- Create a database with at least one table where all variations will be saved (variation ID, selected numbers, submitted date and time, client email). Database could be supplement with others tables (for example - table Client where all information about client is saved)
- Every time when clients send variations to server, it INSERT new record in database
- After lucky number generation by server, it SELECTs all records in this hour and processes the winner or winners.

Good Luck!