

Projekt	Autorzy	Grupa
Szachy	Bartosz Surma i Krzysztof Surdy	2ID15B

## 1. O projekcie

Tematem naszego projektu była gra szachy w oparciu o architekturę klient serwer w języku JAVA. Po kompilacji i spakowaniu do archiwum jar projekt składa się z trzech modułów: klienta, serwera oraz biblioteki zawierającej niezbędne klasy dla modułu klienta i serwera. Do poprawnej pracy wymagana jest obecność plików properties (serwer.properties[serwer], databaseConnection.properties[serwer], client.properties[klient]) w folderze w którym znajdują się JARy.

## 2. Klient

Aplikacja kliencka została wykonana przy użyciu biblioteki Java Swing. Składa ona się z kilku części odpowiedzialnych za:

- interfejs graficzny (klasy JMainFrame, NewJFrame, Square, Szachownica),
- logikę aplikacji (klasy Manager, Player)
- komunikację sieciową z serwerem (klasa Client, która zawiera klasę wewnętrzną MyListener)

Klasa Client wysyła wiadomości do serwera, natomiast posiada również klasę wewnętrzną pracującą w osobnym wątku która nasłuchuje wiadomości od serwera. W archiwum JAR części klienckiej znajdują się również niezbędne pliki graficzne.

## 3. Serwer

Część serwerowa pracuje w trybie konsolowym. Dla każdego nowego połączenia tworzy ona nowy wątek (tzw. miniserwer) w którym będzie przebiegać dalsza część obsługi połączenia z klientem. Po uruchomieniu znajduje się ona w fazie ciągłego nasłuchu nadchodzących nowych połączeń od klientów. Składa się z kilku części odpowiedzialnych za:

- połączenie z bazą danych (klasa DatabaseConnector)
- wysyłanie wiadomości do strumieni wyjściowych i wyjściowych (klasa Streamer)
- logikę działania serwera oraz komunikacji między użytkownikami (Klasy Serwer z klasą wewnętrzną MyConnect, GameRoom)

Po stronie serwera można wywołać jedną z poniższych komend (wpisując w konsole uruchomionego serwera)

- list - wypisuje wszystkich połączonych klientów
- disconnect *user* - rozłącza klienta *user* od serwera
- disconnect all - rozłącza wszystkich klientów
- exit - zamyka serwer i rozłącza wszystkich klientów

## 4. MyLib

Jest to część (tzw. biblioteka) która dostarcza niezbędnych klas wspólnych dla części serwerowej oraz klienckiej. Składa się z dwóch głównych części dostarczających:

- klas służących do komunikacji sieciowej (klasy serializowane)
  - są to klasy używane przez klienta oraz serwer do wysyłania wiadomości różnego typu. Każda posiada zbiór stałych definiujących rodzaj przesyłanej akcji (Klasy AuthMessage, ConfigMessage, MoveMessage, ReturnMessage, StatusMessage, TextMessage oraz klasa po której dziedziczą wszystkie pozostałe Message)
- klas reprezentujących figury na szachownicy, oraz ich pozycje
  - Posiadają one m.in metody odpowiedzialne za generowanie możliwych ruchów figury

## 5. Kompilacja

Do kompilacji wszystkich modułów można użyć poniższego skryptu batch

```
:: Kompilacja i pakowanie biblioteki MyLib
javac pl/boardPieces/*.java pl/boardPieces/chessmans/*.java
pl/communication/serializableMessage/*.java
jar -cvf MyLib.jar pl/boardPieces/*.class pl/boardPieces/chessmans/*.class
pl/communication/serializableMessage/*.class

::Kompilacja i pakowanie modułu serwera
javac pl/communication/database/*.java pl/communication/serwer/*.java
pl/communication/writerReader/*.java
jar -cvfm serwer.jar manifest2.txt pl/communication/database/*.class
pl/communication/serwer/*.class pl/communication/writerReader/*.class

::Kompilacja i pakowanie modułu klienta
javac -classpath MyLib.jar pl/gui/*.java pl/communication/*.java
pl/logic/management/*.java
jar -cvfm client.jar manifest1.txt pl/gui/*.class pl/communication/*.class
pl/logic/management/*.class pl/image/*.*
```

## 6. Uruchomienie

Aby uruchomić serwer można skorzystać z poniższego skryptu:

```
::Uruchomienie serwera i pozostawienie sterowania w konsoli
start java -jar Serwer.jar
```

Aby uruchomić klient można użyć skryptu:

```
::Uruchomienie klienta i zamknięcie konsoli
start javaw -jar Client.jar
```

Aby uruchomić archiwum serwera w folderze powinny znajdować się pliki:

- databaseConnection.properties
- serwer.properties

- MyLib.jar
- ojdbcx.jar

Natomiast aby uruchomić moduł klienta, w folderze powinny znajdować się pliki:

- klient.properties
- MyLib.jar