

# JChat

## Dokumentácia

Ondřej Kipila

6. mája 2014

## 1 Zadanie projektu

Program bude určený na zabezpečenie komunikácie viacerých používateľov v reálnom čase. Prostredníctvom programu bude možné posilať niekoľko typov správ (textové, obrázkové. . . ), pričom budú používateľom zobrazované rôznym spôsobom podľa typu správy.

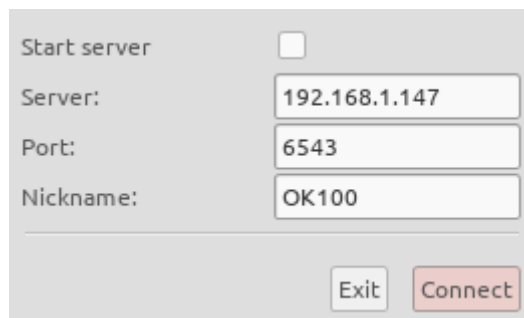
Používatelia budú mať možnosť vytvárať a pripájať sa do „miestností“, pričom správy posielané v určitej miestnosti budú dostupné len pre používateľov, ktorí sú k danej miestnosti pripojení. Program bude umožňovať aj posielanie správ určených len pre konkrétneho používateľa (pre ostatných nebude viditeľná).

Používatelia tiež budú mať rozdielne oprávnenia, napr. na zablokovanie určitého užívateľa, atď.

Program bude pracovať v textovom aj grafickom režime.

## 2 Návod na použitie programu

Po spustení programu sa zobrazí prihlasovacie okno (obr. 1). Pri výbere voľby „Start server“ bude aplikácia spustená v režime server-klient, v opačnom prípade bude spustená v režime klient. Po stlačení tlačidla „Connect“ prebehne prihlásenie a zobrazí sa hlavné okno aplikácie (obr. 2).



Start server ☐

Server: 192.168.1.147

Port: 6543

Nickname: OK100

Exit Connect

Obr. 1: Prihlasovacie okno



Obr. 2: Hlavné okno aplikácie

Hlavné okno je rozdelené na niekoľko častí. V hornej časti okna je hlavná ponuka a záložky miestností. V strednej časti sa zobrazujú správy vo vybranej miestnosti.

V dolnej časti okna sa nachádza textové pole na odosielanie textových správ. Tlačidlo „Send image...“ v pravom dolnom rohu je určené na odosielanie grafických správ.

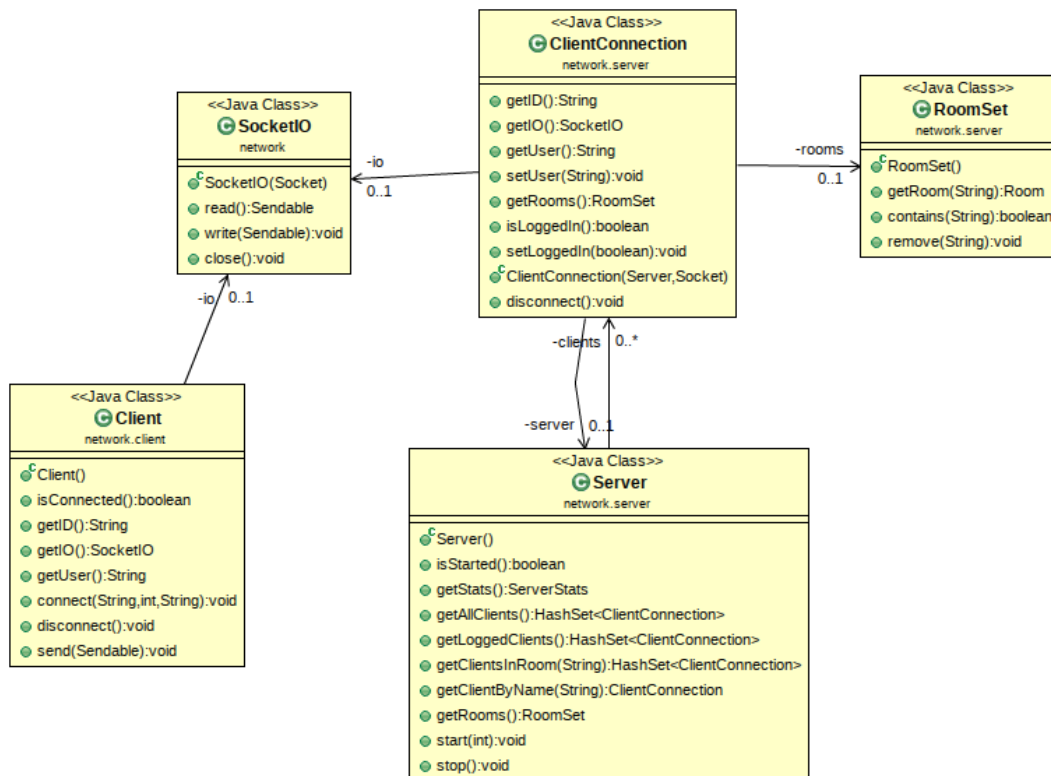
V textovom poli je tiež možné používať rôzne príkazy (tab. 1), ktoré sú tiež dostupné z ponuky „Commands“.

Príkaz	Syntax príkazu	Popis príkazu
<b>block</b>	<code>/block MENO-POUZIVATELA ZABLOKOVAT-ODBLOKOVAT</code>	Zablokovanie/odblokovanie používateľa v danej miestnosti. Parameter <b>MENO-POUZIVATELA</b> udáva meno používateľa, parameter <b>ZABLOKOVAT-ODBLOKOVAT</b> udáva, či bude používateľ zablokovaný alebo odblokovaný ( <b>true</b> pre zablokovanie, <b>false</b> pre odblokovanie). Príkaz je dostupný len pre správcu miestnosti (používateľ, ktorý ju vytvoril).
<b>join</b>	<code>/join NAZOV-MIESTNOSTI</code>	Pripojenie používateľa k miestnosti. Parameter <b>NAZOV-MIESTNOSTI</b> udáva názov miestnosti ku ktorej bude používateľ pripojený. V prípade, že miestnosť ešte nebola vytvorená, používateľ sa automaticky stane jej správcou.
<b>leave</b>	<code>/leave</code>	Odpojenie používateľa od miestnosti. Používateľ sa odpojí od miestnosti v ktorej odoslal tento príkaz.
<b>rooms</b>	<code>/rooms</code>	Vypíše zoznam všetkých miestností.
<b>stats</b>	<code>/stats</code>	Vypíše informácie o serveri.
<b>users</b>	<code>/users</code>	Vypíše zoznam používateľov ktorí sú pripojení k miestnosti, v ktorej bol príkaz odoslaný.

Tabuľka 1: Prehľad príkazov

## 3 Štruktúra systému

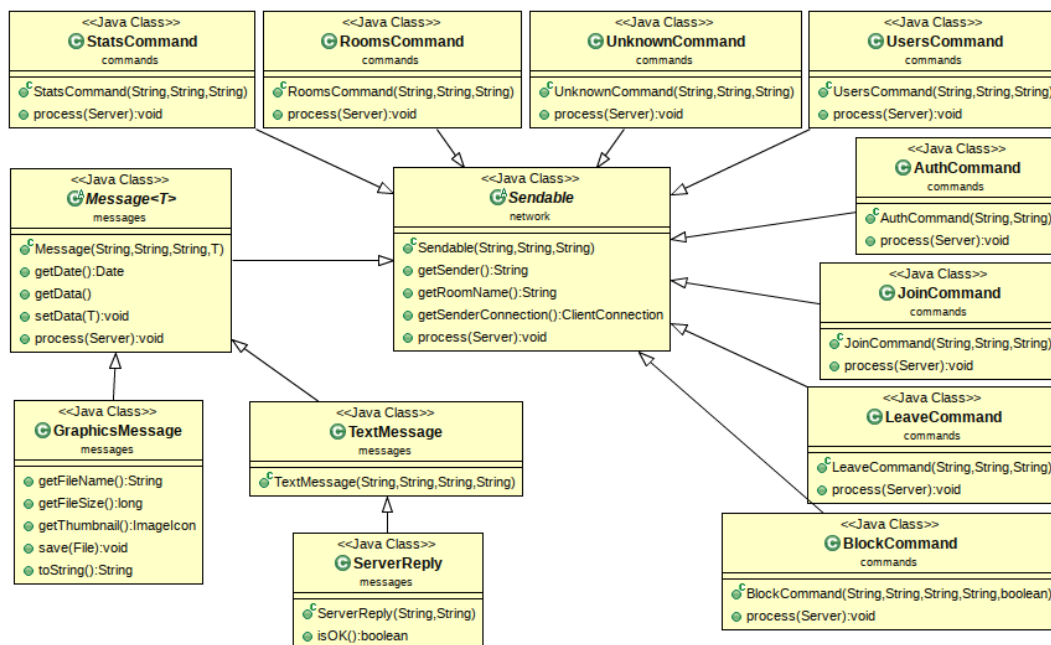
### 3.1 Sieťová časť



Obr. 3: Diagram najdôležitejších tried

- **Server** — Hlavná trieda servera. Server je schopný spracovať max. 10 súčasných požiadaviek na pripojenie. Server si uchováva referenciu na objekt typu **ClientConnection** pre každého pripojeného klienta.
- **Client** — Hlavná trieda klienta. Príjem a odosielanie správ je realizovaný prostredníctvom zapúzdreného objektu typu **SocketIO**. Príjem správ zo servera prebieha v samostatnom vlákne.
- **SocketIO** — Zabezpečuje príjem a odosielanie serializovateľných objektov prostredníctvom socketu.
- **ClientConnection** — Pripojenie servera a klienta. Príjem správ od klienta prebieha v samostatnom vlákne.
- **RoomSet** — Obsahuje zoznam miestností ku ktorým je daný používateľ pripojený.

### 3.1.1 Správy

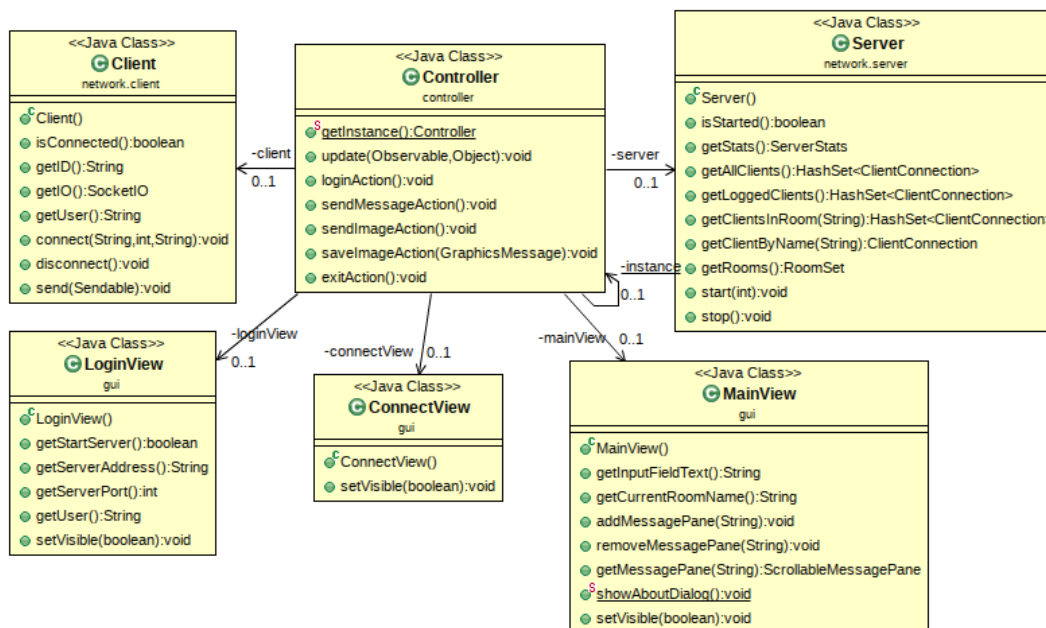


Obr. 4: Diagram najdôležitejších tried

- **Sendable** — Abstraktná trieda objektu ktorý je možné posilať.
- **Message** — Abstraktná trieda správy generického typu.
- **GraphicsMessage** — Grafická správa.
- **TextMessage** — Textová správa.
- **ServerReply** — Textová správa posiadaná serverom.
- **AuthCommand** — Príkaz pre prihlásenie.
- **BlockCommand** — Príkaz pre zablokovanie/odblokovanie používateľa.
- **JoinCommand** — Príkaz pre pripojenie k miestnosti.
- **LeaveCommand** — Príkaz pre odpojenie od miestnosti.
- **RoomsCommand** — Príkaz pre výpis miestností.
- **StatsCommand** — Príkaz pre výpis informácií o serveri.
- **SyntaxErrorCommand** — Príkaz s chýbajúcimi parametrami je nahradený týmto príkazom.

- UnknownCommand — Neznámy príkaz je nahradený týmto príkazom.
- UsersCommand — Príkaz pre výpis používateľov.

### 3.2 Aplikačná časť



Obr. 5: Diagram najdôležitejších tried

- **Controller** — Zabezpečuje prepojenie grafického používateľského rozhrania a modelu (klient, server).
- **Client** — Hlavná trieda klienta.
- **Server** — Hlavná trieda servera.
- **LoginView** — Prihlasovacie okno.
- **ConnectView** — Okno zobrazené v priebehu prihlasovania.
- **MainView** — Hlavné okno aplikácie.

## 4 Kritériá hodnotenia

**Hlavné kritériá** Aplikácia podľa môjho názoru splňuje všetky hlavné kritériá hodnotenia.

- Aplikácia zodpovedá spresnenému zadaniu, jediná chýbajúca vlastnosť je textové používateľské rozhranie, ktoré nakoniec nebolo realizované kvôli nemožnosti zobrazit' grafické správy
- Sú použité objektovo-orientované mechanizmy Javy (dedenie, polymorfizmus, zapuzdrenie, agregácia)
- Zdrojový kód obsahuje bežné aj Javadoc komentáre

### Dodatočné kritériá

- Použité návrhové vzory — MVC, factory, observer, proxy, singleton
- Kód je organizovaný do balíkov
- Sú ošetrené mimoriadne stavy, vrátane vlastných výnimok
- Grafické používateľské rozhranie je oddelené od aplikačnej logiky pomocou návrhového vzoru MVC
- Využitie viacnítovosti, dynamické vytváranie vlákien za behu aplikácie
- Použitie pokročilých mechanizmov Javy — RTTI, vnhiezdené triedy, genericosť vo vlastných triedach