Zadání 'jádra' úloh

Hra piškvorky

Zadání:
Implementujte server, který nabídne hráči (klientovi) možnost vytvořit a hrát hru piškvorky. Na jednom serveru bude moci být spuštěno vícero (instancí) her. Každá z her bude mít na serveru soubor popisující její průběh (z něhož půjde rekonstruovat dosavadní průběh, případně zrekonstruovat poslední validní stav).

Funkce poskytované serverem:
Server bude klientovi poskytovat následující funkce:

- `gameState[] showGames()`
  - kde `gameState` je struktura (objekt) skládající se ze jména hry (string) a id stavu (integer – 0 = OPEN = čerstvě vytvořená hra bez hráčů, 1 = WAITING = čeká se na další hráče, 2 = IN PROGRES = hra je hrána, 3 = ENDED = hra je ukončena)
- `boolean createGame(gname, x, y, z, p)`
  - kde `gname` je jméno hry (string), `x` je vodorovný rozměr hracího pole (integer), `y` je svislý rozměr hracího pole (integer), `z` je počet vítězných žetonů v řadě, `p` je počet hráčů a návratová hodnota udává jestli se podařilo hru vytvořit (minimální požadavek na hru je 3x3, na 3 vítězné pro 2 hráče)
- `long joinGame(gname, pname) throws NoGameException, FullGameException`
  - kde `gname` je jméno hry (string), `pname` je jméno hráče (string) a návratová hodnota je náhodně vygenerovaný identifikátor daného hráče
- `void joinGame(gname, pname, pid) throws NoGameException, FullGameException`
  - kde `gname` je jméno hry (string), `pname` je jméno hráče (string) a `pid` je identifikátor hráče (long)
- `integer doMove(gname, pid, x, y) throws WrongTurnException`
  - kde `gname` je jméno hry, `pid` je identifikátor hráče, `x` je souřadnice vodorovné osy, `y` je souřadnice svislé osy a návratová hodnota je 1 pokud hráč vyhrál
- `gameStatus getGameStatus(gname)`
  - kde `gname` je jméno hry a `gameStatus` je struktura mající položky `name` (string), `state` (viz id stavu u funkce showGames), `player[]` (string[]), `winner` (int), `board` (datová struktura reprezentující hrací pole)
- … další funkce jež uznáte za vhodné

Vnitřní struktury
Soubor popisující hru na serveru by se měl jmenovat jako hra samotná s příponou `.gme` a měl by mít následující tvar:

```
game;jméno_hry;vodorovný_rozměr;svislý_rozměr;počet_vítězných;počet_hráčů
player;číslo_hráče;jméno_hráče;id_hráče
...
turn;číslo_tahu;číslo_hráče;vodorovný_rozměr;svislý_rozměr
turn;číslo_tahu;číslo_hráče;vodorovný_rozměr;svislý_rozměr
...
```

Klient
Klient bude mít možnost ovládat server pomocí příkazů, které budou mít stejný název jako výše uvedené požadované funkce serveru. Povinností je dávkové zpracování (interaktivní konzole je volitelná).

Dávkové zpracování

```
./server [IP_adresa] [cislo_portu]
```

, kde *IP_adresa* a *cislo_portu* specifikuje, kde má server naslouchat

```
./client IP_adresa cislo_portu konf_soubor
```

, kde *IP_adresa* a *cislo_portu* specifikují, kde je potřeba hledat server a v souboru je posloupnost příkazů pro klienta. Příkazy mohou mít například následujících tvar:

```
showgames
creategame test001;3;3
joingame test001;pl1;123456
joingame test001;pl2;234567
domove test001;123456;2;2
domove test001;234567;1;1
domove test001;123456;1;2
domove test001;234567;3;3
getGameStatus test001
domove test001;123456;3;2
getGameStatus test001
```