Pontifícia Universidade Católica do Rio Grande do Sul – PUCRS
Faculdade de Informática
Programação Distribuída – Turma 128 – Prof. Roland Teodorowitsch – 2014/1

Trabalho 2 – Elaboração de uma Aplicação Distribuída em Java Web Services para um Jogo da Velha com Teste Automatizado (DEFINIÇÃO)

Objetivos:

- exercitar a programação distribuída usando Web Services na linguagem Java;
- desenvolver uma aplicação formada por dois programas, um servidor (executado por um processo) e um cliente (eventualmente executado por vários processos) que interagem entre si para a solução de determinado problema;
- desenvolver uma aplicação servidora capaz de receber acessos concorrentes, gerenciando vários jogos simultaneamente, sem apresentar falhas de consistência.

Descrição:

- Neste trabalho deverá ser implementada uma aplicação distribuída em Java Web Services que permita o gerenciamento de várias partidas de Jogo da Velha (conforme regras definidas no Trabalho 1 desta disciplina), cada um com 2 jogadores, simultaneamente. A aplicação deverá ser formada por dois programas: um servidor (executado por um processo) e um cliente (eventualmente executado por vários processos). O servidor deverá funcionar de forma muito parecida com o servidor implementado como Trabalho 1 desta disciplina. No entanto, antes do registro normal de um usuário para iniciar uma partida, o servidor deverá aceitar a especificação de um identificador para determinado usuário. Esta especificação (de identificador) servirá apenas como ferramenta de teste automatizado do servidor. O programa cliente, por sua vez, terá uma estrutura diferente da estrutura do cliente tradicional para execução de um Jogo da Velha (portanto, diferente da estrutura do cliente sugerida para o Trabalho 1 desta disciplina). A nova estrutura deverá ser capaz de ler da entrada padrão (teclado) a especificação de um conjunto de chamadas remotas que devem ser iniciadas no servidor. E deverá exibir na saída padrão (vídeo) o resultado da execução de cada uma destas chamadas. Para iniciar uma partida de Jogo da Velha no servidor remoto, o cliente precisa fazer o registro de seu usuário, recebendo como resposta um identificador (que será usado nas demais chamadas). O uso de determinado nome de usuário deve ser feito de forma única, sem permitir dois usuários com o mesmo nome e reservando-se o direito de uso de determinado nome a quem registrar-se antes no servidor.

Especificação de Entradas e Saídas de um Cliente:
O programa cliente deverá ser capaz de ler a especificação de um conjunto de chamadas remotas da entrada padrão (teclado) e deverá ser capaz de escrever as respectivas respostas na saída padrão (vídeo). Entradas e saídas serão sempre fornecidas em formato texto (codificação ASCII, sem acentos).

A especificação da entrada começa com o número total de operações remotas que o cliente deve enviar para o servidor. E a seguir são apresentadas, cada uma em uma linha, as operações. Considera-se 7 operações:

- operação 0 – preRegistro (usada para viabilizar o teste): informa ao servidor o nome de usuário e o respectivo identificador que o servidor deverá utilizar para este usuário;
- operação 1 – registraJogador (relacionada ao jogo propriamente dito): recebe o nome do usuário e retorna como resposta o identificador (valor inteiro) que identifica este usuário de forma única no sistema (este identificador será utilizado nas chamadas subsequentes);

- operação 2 – temPartida (relacionada ao jogo propriamente dito): recebe o identificador do usuário e retorna um valor inteiro que pode ser -2 (tempo de espera esgotado), -1 (erro), 0 (ainda não há partida), 1 (sim, há partida e o jogador inicia jogando com “X”) ou 2 (sim, há partida e o jogador é o segundo a jogar, usando “O”);
- operação 3 – ehMinhaVez (relacionada ao jogo propriamente dito): recebe o identificador do usuário e retorna um valor inteiro que pode ser -1 (erro), 0 (não), 1 (sim), 2 (é o vencedor), 3 (é o perdedor) ou 4 (houve empate);
- operação 4 – obtemGrade (relacionada ao jogo propriamente dito): recebe o identificador do usuário e retorna uma cadeia de caracteres (String) vazia (em caso de erro) ou uma cadeia de caracteres com a representação da grade de jogo apresentada de forma linear, sem grade (conteúdo das posições de 0 até 8, sendo que usa-se: “.” para posição vazia, “X” para jogada do jogador 1 e “O” para jogada do jogador 2);
- operação 5 – enviaJogada (relacionada ao jogo propriamente dito): recebe o identificador do usuário e a jogada (valor inteiro de 0 a 8, correspondendo à posição da jogada na grade), retornando um valor inteiro que pode ser 2 (partida encerrada, o que ocorreria caso o jogador demorasse muito para enviar a sua jogada, o que não será testado nesta implementação), 1 (tudo certo), 0 (posição ocupada) ou -1 (erro);
- operação 6 – obtemOponente (relacionada ao jogo propriamente dito): recebe o identificador do usuário e retorna uma cadeia de caracteres (String) vazia (em caso de erro) ou uma cadeia de caracteres com o nome do oponente.

A seguinte sequência corresponde a um exemplo de entrada que reproduz um jogo simples, com quatro jogadores:

```
35
0    jogador1:10
0    jogador2:20
1    jogador1
2    10
1    jogador2
2    10
2    20
6    10
6    20
3    10
3    20
```

```
4    10
4    20
5    10:4
4    10
5    20:4
5    20:8
4    10
5    10:0
4    20
3    10
3    20
5    20:5
```

```
4    10
3    10
3    20
5    10:2
4    20
5    20:7
4    10
5    10:1
4    10
4    20
3    10
3    20
```

Nesta entrada há 35 chamadas remotas especificadas.
As primeiras 2 operações (tipo 0) informam ao servidor que: para jogador 1 o registro deverá retornar o identificador 10 e para jogador 2 o registro deverá retornar o identificador 20. Estas 2 chamadas retornam 0 em caso de sucesso ou 1 em caso de fracasso.
As operações tipo 1 fazem o registro dos jogadores, recebendo como resposta os valores pré-cadastrados.
As operações tipo 2 testam se há partida. No exemplo, depois do “jogador1” se cadastrar ele usa esta operação e recebe como resposta que “não” há partida. Depois do “jogador2” se cadastrar, executam-se 2 operações tipo 2, que agora retornaram que há partida para ambos.
A operação tipo 6 é usada para obter o nome do oponente.
A operação tipo 3 é usada para que o jogador verifique se é a sua vez e também para verificar o estado da partida (finalizada com vitória, empate ou derrota).

A operação tipo 4 retorna uma representação do conteúdo da grade do Jogo da Velha.
E a operação tipo 5 é usada para enviar jogadas.

Para cada entrada o cliente deverá gerar uma saída. Esta saída conterá uma linha para cada operação com o resultado ou resposta da respectiva operação.

Para o exemplo de entrada especificado anteriormente, espera-se a seguinte saída:

```
0                   | .........         | 1
0                   | 1                 | 0
10                  | ....X....         | 1
0                   | 0                 | X.X.XO..O
20                  | 1                 | 1
1                   | ....X...O         | X.X.XO.OO
2                   | 1                 | 1
jogador2            | X...X...O         | XXX.XO.OO
jogador1            | 0                 | XXX.XO.OO
1                   | 1                 | 2
0                   | 1                 | 3
.........           | X...XO..O         |
```

Avaliação:
O programa cliente será testado para vários conjuntos de entradas e, caso gere as saídas esperadas para todos os exemplos de entrada, obterá a nota máxima. O requisito mínimo para entrega e apresentação corresponde a apresentar corretamente os resultados esperados para o exemplo de entrada apresentado nesta definição. Nos demais casos será avaliado tanto o número de entradas acertadas quanto o nível de erro apresentado.

Outras Especificações:

- O trabalho poderá ser feito por grupos de no máximo 2 pessoas;
- Não incluir nenhum código-fonte copiado. A implementação da lógica do Jogo da Velha é relativamente simples e não há necessidade de inclusão de código-fonte elaborado por terceiros (por exemplo, obtidas de colegas ou da Internet). Em caso de uso de código-fonte não desenvolvido pelos alunos, será atribuída a nota 0 (ZERO) ao trabalho.

Data de Entrega:

- 21h15min do dia 1 de julho de 2014.

Formato de Entrega:

- Entregar os arquivos referentes ao código-fonte. Apresentar e descrever verbalmente o funcionamento do programa ao professor.