



Relatório: Remote Method Invocation (RMI) e Remote Procedure Call (RPC)

Daniel Yuji Yamada nºUSP: 10430920

Arthur Prince de Almeida nºUSP:10782990

Rafael Nazima nºUSP: 11208311

Funcionamento

O EP visou implementar um exemplo de sistemas distribuídos sobre peças (parts) tendo como escolha o RMI (Remote Method Invocation), sendo uma API feita em java. Para isto a implementação pode ser dividida em duas partes principais: o servidor e o cliente. Com isso, podemos simular as diferentes parts, cada um pertencendo a um dado repositório de um dado servidor, além de atender requisições de uma outra simulação de um dado cliente.

A principal função do servidor é criar um repositório o qual será possível manipular várias peças. Quando inicializado, deve-se passar de parâmetro o nome do repositório de modo que não haja conflito de nome com outro repositório. Ao fazer isso dois objetos são criados: Parts, que é uma lista ligada de objetos Part, e PartCorrente, que a princípio é inicializada como null mas representa qual Part está rodando no seu repositório.

Para inicializar uma part, deve-se criar uma part no repositório corrente pelo cliente. Em seguida, obter o código usando a função “p” ou olhando no log do servidor. Com esse código será possível atualizar o atributo partCorrente usando a função “g” e passando-o como parâmetro.

O cliente é executado em main, onde, a princípio, se escreve o nome do repositório que o cliente deseja acessar. Caso deseje trocar de repositório basta usar a função “b” e passar o nome do novo repositório. Um objeto Repositorio é obtido pelo Naming.lookup a partir do nome do repositório procurado, ou seja, os nomes dos repositórios ficam registrados no stub de “PartRepository”.

É apresentado uma tabela de comandos o qual o cliente pode executar, caso esqueça algum comando é possível acessar essa tabela novamente usando a função h. As outras possibilidades de comandos são: b para se conectar a outro servidor com nome inserido e mudar o repositório corrente ; l para listar as peças do repositório corrente; s para mostrar os atributos da peça corrente; g que recebe o código da peça e a torna peça corrente; c para esvaziar a lista de sub-pecas corrente e apagar a peça corrente; a para adicionar a lista de sub-pecas corrente n unidades da peça corrente; d adiciona uma peça ao repositório corrente); e por fim q para encerrar a execução

A classe prompt guarda os métodos a ser executados por um repositório no main do cliente utilizando as informações obtidas pelos métodos do objeto repositório.

Estruturas

A implementação consiste em duas interfaces: Part, que representa cada peça, e PartRepository, que é essencialmente uma coleção de parts. Ambos são uma extensão da interface de Remote do java, o qual permite ao java identificar interfaces que conseguem realizar uma chamada de uma máquina virtual local ou não-local.

A estrutura do repositório é feita em árvore, ou seja, o repositório tem uma lista de peças que são considerados as parts pais. E cada part pode ou não ter subcomponentes. Isso significa que é possível entrar em uma part, que é sub-part de uma peça, e criar sub-part para essa sub-part. Por outro lado, o repositório apenas guarda a lista de pais, então para acessar uma sub-part é preciso buscar o código do nome dela e transformá-la em part corrente. Portanto têm-se uma estrutura em que o cliente guarda qual é o repositório corrente, o servidor/repositório guarda qual são as parts pais e a part corrente, e as parts guardam suas sub-componentes parts. Isso significa que quem manipula o repositório é apenas o servidor, e o cliente apenas envia os comandos.

Exemplos:

Inicialização dos servidores:

--servidor1

Escolha o nome do servidor:

rep1

--servidor2

Escolha o nome do servidor:

rep2

--Cliente

insira o nome do repositório

rep1

-----Lista de comandos-----

h (help - retorna essa lista de comando)

b (se conecta a outro servidor com nome inserido e muda o PartRepository corrente)

p (Retorna o código da peça)

l (Lista as Parts do PartRepository corrente)

s (Mostra atributos da Part corrente)

g (Escreve nome da Part a se tornar Part corrente)

c (Esvazia a lista de sub-Parts corrente e remove a part corrente)

a (Adiciona a lista de sub-Parts corrente n unidades da Part corrente)

d (Adiciona uma Part ao PartRepository corrente)

q (Encerra a execucao do cliente)

d

Insira o nome da Part a ser adicionada:

part

Insira a descricao da Part a ser adicionada:

test1

true

p

insira o nome da peca

part

nome da peca: part codigo da peca: 3433459

g

Insira o codigo da Part:

3433459

o codigo se refere a: part

s

-----Atributos da pessa corrente-----

Nome da Part corrente: part

Descricao da Part corrente: test1

Codigo da Part corrente: 3433459

Quantidade de sub-Parts da Part corrente: 0

a

Insira a quantidade da Part:

5

true

s

-----Atributos da pessa corrente-----

Nome da Part corrente: part

Descricao da Part corrente: test1

Codigo da Part corrente: 3433459

Quantidade de sub-Parts da Part corrente: 5

Nome da subPart da Part corrente: part/0

Nome da subPart da Part corrente: part/1

Nome da subPart da Part corrente: part/2

Nome da subPart da Part corrente: part/3

Nome da subPart da Part corrente: part/4

d

Insira o nome da Part a ser adicionada:

peca

Insira a descricao da Part a ser adicionada:

test2

true

p

insira o nome da peca

peca

nome da peca: peca codigo da peca: 3436819

g

Insira o codigo da Part:

3436819

o codigo se refere a: peca

s

-----Atributos da pessa corrente-----

Nome da Part corrente: peca

Descricao da Part corrente: test2

Codigo da Part corrente: 3436819

Quantidade de sub-Parts da Part corrente: 0

l

part

peca

c

true

l

part

b

Insira o nome do repositório:

rep2

d

Insira o nome da Part a ser adicionada:

part2

Insira a descrição da Part a ser adicionada:

test3

true

l

part2

q

--servidor1

Escolha o nome do servidor:

rep1

Part part adicionada no repositório com o código: 3433459

part corrente: part

5 sub-parts adicionadas em part

Part peça adicionada no repositório com o código: 3436819

part corrente: peça

lista apagada

--servidor2

Escolha o nome do servidor:

rep2

Part part2 adicionada no repositório com o código: 106437279

Ferramentas utilizadas:

- Linguagem: Java
- IDE: Eclipse