

Primeiro Trabalho - Sexta Etapa (parte b)

MC102 - Algoritmos e Programação de Computadores

Instituto de Computação
Universidade Estadual de Campinas

1º semestre de 2012

Turmas Q e R

Professor: André Santanchè

PED: Daniel Cintra Cugler

Resumo

Dando continuidade à parte desenvolvida na etapa 06a, implemente uma função de prestação de serviços de seu planeta, conforme descrito a seguir.

Detalhamento

O código do framework desta etapa não estará mais disponível para download em formato ZIP. Você deverá baixar o código através do Mercurial no projeto c2learn (<http://code.google.com/p/c2learn/>). Veja dois vídeos no site de trabalho que explica como fazê-lo através do TortoiseHg.

A partir de agora cada equipe terá um planeta com nome a sua escolha. Por esta razão, foi criada uma nova instrução chamada `registro` que deve ser chamada no início da função `programa()` (veja exemplo que acompanha esta etapa do trabalho). Esta função recebe dois parâmetros: nome do planeta e chave privada. O nome do planeta será escolhido por cada equipe e não pode haver repetição, entretanto, provisoriamente para esta etapa do trabalho deve ser usado o nome "Ator01". A chave privada será um sistema de segurança para impedir clonagem de identidade. Provisoriamente o valor é sempre 0.

O módulo `executaInstrucao` da etapa 06a foi adaptada para suportar novas instruções a ser criadas pelas equipes. Nesta nova versão, a cada vez que `executaInstrucao` não reconhece a instrução, ela despacha a instrução para uma função relacionada ao planeta corrente. Por exemplo, se o robô viajar para o planeta `Ator01` e disparar uma instrução desconhecida, esta instrução será encaminhada para a função do planeta `Ator01` para que ele a trate. Por esta razão, cada planeta terá uma função com seu nome prefixador por `ativa`. Então o planeta `Ator01` tem a função `ativaAtor01` que é disparada toda vez que houver uma instrução para ele. Esta função recebe quatro parâmetros:

- `strInstrucao` → string contendo a instrução;
- `paramStr` → parâmetro que acompanha a instrução em formato string;
- `paramInt` → parâmetro que acompanha a instrução em formato inteiro (se for possível converter);
- `valor` → se o robô sacar um valor antes de acionar uma instrução do planeta, este valor é automaticamente transferido para o respectivo planeta e é informado como parâmetro aqui.

Agora o planeta pode cobrar pelos seus serviços. Toda vez que o robô executa uma instrução `saca-<valor>` antes de chamar uma instrução do planeta, o respectivo valor é transferido para o planeta.

O objetivo desta etapa do trabalho é verificar se você entendeu o que foi explicado. Para isto você deve copiar para esta nova versão do framework o código que você implementou em `programa` na etapa 06a, para que o programa seja capaz de ler e interpretar instruções. Não se esqueça de manter a chamada 'a função que registra o planeta antes de ler as instruções.

Além disso, considere a seguinte sequência de instruções que está no Susy:

```
viaja-Ator01;saca-150;zumpt-X
```

O robô viajou para o planeta `Ator01` (seu planeta hipotético) sacou 150 e solicitou um serviço de nome `zumpt` com parâmetro `X` para o planeta `Ator01`. Estenda a função do planeta `Ator01` de modo que ele imprima na tela uma mensagem: `* recebi <valor> e executei o servico *`

Neste caso, `<valor>` é substituído pelo valor recebido.

Entrada

- Sequência de instruções separadas por ponto-e-vírgula.

Saída

Mensagem gerada pela função `ativaAtor01`.

Entrega

- Até o dia 07/05 o trabalho deverá ser submetido pelo Susy.