

Primeiro Trabalho - Sexta Etapa (parte a)

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

Adapte o programa desenvolvido na etapa 2 do mercante, da seguinte maneira: utilize o framework mercante para ler pelo teclado uma string contendo um programa do robô e chamar uma nova função `executaInstrucao` que executará a instrução.

Detalhamento

Os arquivos `programa.c` e `programa.h` foram renomeados para `ator01.c` e `ator01.h` e transferidos para o diretório `ator`.

Considere a seguinte função que foi incluída no arquivo `ator01.c`:

```
void executaInstrucao(char* strInstrucao, TipoInstrucao codInstrucao,  
                    char *paramStr, int paramInt)
```

Ela é responsável por receber dados sobre uma instrução e executá-la. Por esta razão, a função `programa()` não deve mais executar diretamente as instruções, mas deve fazê-lo através da função `executaInstrucao`. Ao ler a string contendo a sequência de instruções pelo teclado, a sua função `programa()` deve separar cada instrução e acionar a função `executaInstrucao` passando os seguintes parâmetros:

`strInstrucao` → a instrução sem os parâmetros

`codInstrucao` → o código da instrução definido por uma constante conforme a tabela a seguir;

`paramStr` → o parâmetro da instrução em formato string original;

`paramInt` → parâmetros inteiros convertidos ou número da mercadoria ou planeta quando for o caso (veja detalhes na tabela a seguir).

Exemplo: para a instrução → `viaja Tander`

`strInstrucao` → `viaja`

`codInstrucao` → `0 (instrViaja)`

`paramStr` → `Tander`

`paramInt` → `5`

Se a instrução não for conhecida mesmo assim deve ser acionado o `executaInstrucao`. Neste caso, o `codInstrucao` deve ser `instrDesconhecida (-1)`, mas os parâmetros `strInstrucao` e `paramStr` devem conter a instrução e parâmetro respectivamente. O parâmetro `paramInt` deve ter o `paramStr` convertido para número se `paramStr` contiver um inteiro.

A tabela a seguir detalha como você deve tratar cada instrução. Observe que foi incluída uma nova instrução chamada `saca`, em que é informado o valor a ser sacado.

Instrução	intCodigo	paramInt	Descrição
<code>viaja-<número_planeta></code>	<code>instrViaja</code>	<code><número_planeta></code>	
<code>viaja-<nome_planeta></code>	<code>instrViaja</code>	<code><número_planeta></code>	Na tradução o <code><nome_planeta></code> informado deve ser convertido para <code><número_planeta></code> . Para encontrar o número do planeta a partir do nome, o vetor de planetas (obtido por <code>getNomePlanetas()</code>) deve ser percorrido.
<code>indica-<número_mercadoria></code>	<code>instrIndica</code>	<code><número_mercadoria></code>	
<code>indica-<nome_mercadoria></code>	<code>instrIndica</code>	<code><número_mercadoria></code>	Na tradução o <code><nome_mercadoria></code> informado deve ser convertido para <code><número_mercadoria></code> . Para encontrar o número da mercadoria a partir do nome, o vetor de planetas (obtido por <code>getNomeMercadorias()</code>) deve ser percorrido.
<code>compra-<quantidade></code>	<code>instrCompra</code>	<code><quantidade></code>	
<code>vende-<quantidade></code>	<code>instrVende</code>	<code><quantidade></code>	
<code>saca-<valor></code>	<code>instrSaca</code>	<code><valor></code>	Saca o valor em dinheiro informado.
<code><instrução_desconhecida></code>	<code>instrDesconhecida</code>	<code><valor></code>	Se o parâmetro que acompanha for inteiro, <code><valor></code> será a string convertida para inteiro, senão valor será 0.

Entrada

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

Saída

O seu programa não deve gerar saídas. Elas serão geradas pelo sistema de execução de instruções.

Entrega

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