Guião 4: *Draw a Grid*

Versão 1.1

INTRODUÇÃO

O objectivo deste guião é que resolva um problema do concurso de programação ACM – ICPC (International Collegiate Programming Contest). O problema escolhido é o "Draw a Grid" (apresentado em anexo).

Neste projecto vão ser disponibilizados 5 testes, sendo cada teste composto por 2 ficheiros texto: um ficheiro para os dados de entrada; outro para os dados de saída. O mooshak irá verificar 5 testes similares aos testes disponibilizados, garantindo que um programa que passe os 5 testes localmente também passará todos os testes no mooshak. Por esta razão, o número de submissões ao mooshak das soluções deste guião podem penalizar a sua nota neste trabalho. As primeiras cinco submissões de cada problema, não penalizam a cotação do problema, no entanto por cada submissão subsequente será descontado 10% da cotação do problema.

Como testar localmente o seu trabalho?

Há pelo menos duas formas de experimentar o seu trabalho. A primeira é executar o programa a partir do ambiente de desenvolvimento Eclipse, como provavelmente está habituado a fazer. A segunda é através da linha de comandos do seu sistema operativo. Quer num caso, quer noutro, desenvolverá o seu programa normalmente no Eclipse.

Primeiro de tudo deve criar na raiz do projecto uma directoria designada tests, executando os seguintes passos:

1. No Package Explorer, seleccionar a raiz do projecto.
2. Clicar no botão direito do rato e no menu de contexto, seleccionar New > Folder...
3. Criar a directoria tests.
4. Importar os ficheiros de teste (ficheiro tests.zip) seleccionando a raiz do projecto e após clicar no botão do rato...
5. Import...
6. Na janela de diálogo que surge, indicar a opção Archive File e de seguida procurar o ficheiro tests.zip.
7. De seguida selecionar a package poo e clicar no botão Finish.

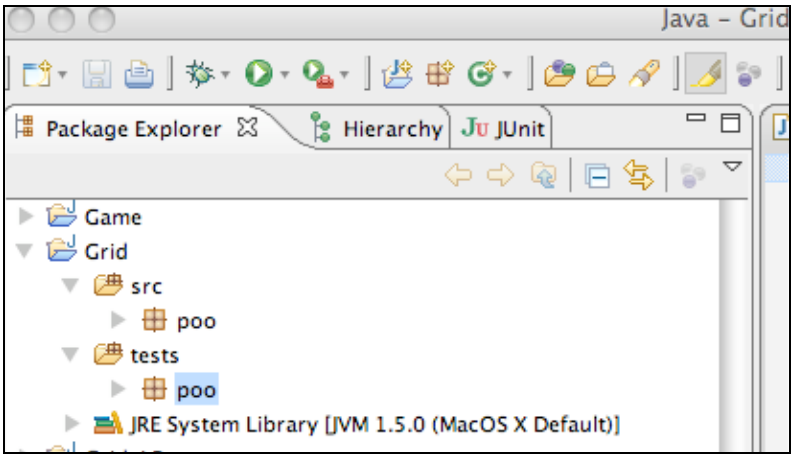


Figura 1 - Estrutura do projecto

Após estes passos a raiz do projecto deve ter a estrutura similar à apresentada na Figura 1.

Para testar o seu programa na consola do sistema operativo deve redireccionar o input, para obter os dados a partir do ficheiro de teste de entrada. Considerando que o input do primeiro teste se chama test01-in.txt e que está guardado na directoria tests, então deverá escrever o seguinte comando, dentro da directoria bin do seu projecto:

```
java –classpath . poo\Main < ..\tests\poo\test01-in.txt
```

O programa escreverá o resultado na consola. Para salvaguardar o output, pode fazê-lo redirigindo-o para um ficheiro, assim:

```
java –classpath . poo\Main < ..\tests\poo\test01-in.txt > ..\tests\poo\res01.txt
```

Para confirmar que o seu programa calcula os resultados esperados deve comparar o ficheiro res01.txt com o output esperado do primeiro teste: ficheiro test01-out.txt. Para tal, podemos inspeccionar visualmente o conteúdo de ambos, ou usar uma ferramenta conhecida como diff.

A vantagem de usar uma ferramenta é que ela nos permite detectar diferenças subtis que de outro modo nos podem passar despercebidas na inspecção visual dos resultados. A ferramenta que vamos usar recebe dois argumentos, com os nomes dos ficheiros que queremos comparar. Caso existam diferenças entre ambos os ficheiros, elas serão assinaladas.

```
diff ..\tests\poo\test01-out.txt ..\tests\poo\res01.txt
```

Caso os ficheiros test01-out.txt e res01.txt sejam iguais a ferramenta não dá nenhum output. Nesta situação, o seu programa passou com sucesso o primeiro teste, não existem diferenças entre o output obtido e o output esperado.

O sistema de avaliação automática de trabalhos (mooshak) faz isto mesmo. Pega no seu programa, gera um executável e depois executa o programa com vários ficheiros

de input. Para cada ficheiro de input, existe sempre um ficheiro com o output respectivo. Se o seu programa gerar o output esperado, passa no teste. Se o resultado for diferente, nem que seja num carácter, o programa falha o teste.

Instalação da ferramenta diff

O diff está disponível nas distribuições standard de sistemas operativos como o Linux, mas tem de ser instalado no Windows. Pode obter a ferramenta na Internet, na página http://gnuwin32.sourceforge.net/packages/diffutils.htm .

Preparar o zip para a submissão da tarefa A

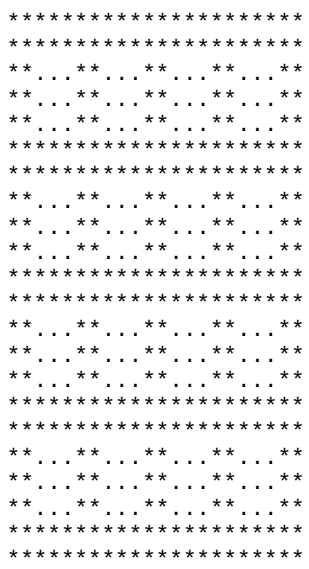
Depois de ter verificado que o seu programa satisfaz os testes disponibilizados, está na altura de o submeter ao mooshak. Para isso deve empacotar a directoria de código fonte poo (incluindo a própria directoria) num ficheiro zip. Não é necessário submeter a directoria de testes.

Problem A

Draw Grid

Input: Standard Input
Output: Standard Output

It is very easy to draw grids with ASCII characters. For example look at the picture below. It shows a (4x4) grid, where each smallest square is of size 3 and the thickness of drawing line is 2.



In this problem your job is very simple: Given the size of the grid, size of smallest square and thickness of drawing line you will just have to draw the grid.

Input

The input file contains at most 101 lines of inputs. Each line contains three integers S, T and N ($0 < S, T, N < 21$). Here S is the size of smallest squares, T is the thickness of drawing line and N is the size of the grid. Input is terminated by a set where the value of S, T and N is zero. This set should not be processed.

Output

For each set of input first produce the serial of output. In next several lines draw an (NxN) sized grid where each smallest square is of size (SxS) and the thickness of drawing line is T. Print a blank line after the output of each case. Note that line pixels are denoted with '*' (asterisk) and blank pixels are denoted with '.'.

Sample Input	Output for Sample Input
3 3 3 2 3 4 0 0 0	Case 1: ********** ********** ********** ***...***...***...*** ***...***...***...*** ***...***...***...*** ********** ********** ********** ***...***...***...*** ***...***...***...*** ***...***...***...*** ********** ********** ********** ***...***...***...*** ***...***...***...*** ***...***...***...*** ********** ********** ********** Case 2: ********** ********** ********** ***..***..***..***..*** ***..***..***..***..*** ********** ********** ********** ***..***..***..***..*** ***..***..***..***..*** ********** ********** ********** ***..***..***..***..*** ***..***..***..***..*** ********** ********** **********

Problemsetter: Shahriar Manzoor
Special Thanks: Md. Kamruzzaman