

Our House

Trabalho Prático de Avaliação 1

Beja, 26 de Outubro de 2009

*Our house it has a crowd
There's always something happening
And it's usually quite loud
Our mum, she's so house-proud
Nothing ever slows her down
And a mess is not allowed
– Our House - Madness*

http://www.youtube.com/watch?v=KwIe_sjKeAY

1 Introdução

Uma das principais características deste enunciado é a de ser, digamos, algo extenso. Tal significa que deve ser lido mais do que uma vez e com muita atenção. O que é pedido está classificado em três tipos de **REQUISITOS**:

1. **ESSENCIAIS**
2. **NÃO ESSENCIAIS**
3. **QUE IMPLICAM PENALIZAÇÕES.**

Os essenciais permitem obter 10 valores, mas apenas desde que não estejam presentes motivos para penalizações, ou seja, se cumprir totalmente todos os requisitos que implicam penalizações. Para não ter penalizações basta ter cuidado a ler o enunciado. Assim, poderá garantir que o programa entregue não irá oferecer motivos para a aplicação de penalizações, as quais podem fazer baixar a nota para níveis certamente indesejados. Os requisitos não essenciais permitem obter até 17 valores. Os 18, 19 e 20 valores ficam apenas para quem fez tudo o que é pedido de forma absolutamente impecável e inventou algo mais para fazer, os chamados **EXTRAS**. Esses extras têm de ter uma dificuldade superior ao que é pedido embora devam continuar na linha do que é pedido. Não vale a pena fazer extras sem ter **TUDO** o que é pedido feito. Também não vale a pena fazer os requisitos não essenciais sem ter feito os essenciais. Em resumo:

- Só pode ter mais do 10 valores se tiver cumprido totalmente os requisitos essenciais e não tiver penalizações.
- Só pode ter mais do 17 valores se tiver cumprido totalmente os requisitos essenciais e também os não essenciais e não tiver penalizações.

Para mais informações pode e deve ler as regras de avaliação no guia de funcionamento da unidade curricular em <http://estig.e-learning.ipbeja.pt/mod/resource/view.php?id=17518>.

2 A casa

O objectivo geral do trabalho é escrever um programa na linguagem de programação Java™, utilizando o ambiente portablep1 (Java™ SDK + BlueJ + Biblioteca ACM) que produza o desenho da Fig. 1

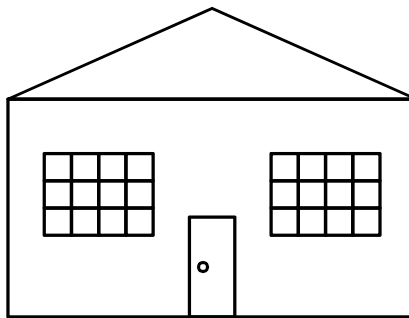


Figura 1: O desenho a produzir pelo programa.

Como ficará claro com os requisitos, o maior desafio neste primeiro trabalho não será fazer um qualquer programa que produza o desenho, mas sim fazer um programa *elegante* que produza o desenho.

Seguem-se os requisitos para o trabalho. Antes de escrever código, leia-os atentamente mais do que uma vez. Antes de entregar o trabalho leia-os, pelo menos, mais uma vez.

3 Requisitos Essenciais

O incumprimento de qualquer destes requisitos implica uma classificação negativa no trabalho e a NÃO contabilização dos requisitos não essenciais.

Req. 1 - A casa (4,0 valores) O programa deve produzir um desenho idêntico ao da Fig 1, feito por uma única tartaruga, mais especificamente por um único objecto da classe `GTurtle`. Note em especial que as posições e tamanho relativos das janelas e da porta devem corresponder às da figura.

Req. 2 - Janelas com métodos (2,0 valores) Deve ter um método em que a tartaruga desenha uma janela. O método deve ser chamado duas vezes: uma vez para cada janela.

Req. 3 - Paredes, texto e porta (3,0 valores) Defina métodos para desenhar a porta (incluindo a maçaneta circular), o telhado e as paredes.

Req. 4 - Utilização de constantes (1,0 valores) O programa deve definir e utilizar constantes simbólicas (nomes de constantes) em lugar de constantes literais (números). Para tal deve definir constantes utilizando a palavra reservada **final**. Por exemplo: `final int FLOOR_CENTER = 400;`

4 Requisitos Não Essenciais

Estes requisitos só são contabilizados se os requisitos essenciais estiverem totalmente correctos. Por outras palavras, não os faça sem ter feito todos os essenciais.

Req. 5 - Ciclos (3,0 valores) Utilize um ciclo para desenhar os vários quadrados que constituem cada janela. Ou seja, no método do requisito 2, que desenha uma janela, deve utilizar um ciclo.

Req. 6 - Ciclos (2,0 valores) No final do desenho, deve surgir no ecrã a distância (em pontos) percorrida pela tartaruga. Para tal utilize um objecto da classe `GLabel`, tal como o que é utilizado pelo programa inicialmente gerado no `portablep1` para escrever IPB-ESTIG no ecrã.

Req. 7 - Ciclos (2,0 valores) O programa deve simular uma situação em que a tartaruga anda no chão e não voa! Ou seja, para mudar de sítio sem desenhar, deve levantar a caneta (*penUp*) e ir para esse outro sítio onde então poderá baixar a caneta (*penDown*) para continuar o desenho. Em particular, a tartaruga nunca deve utilizar o método `setLocation`; o método `add` da classe `GraphicsProgram` só deve ser utilizado uma vez.

5 Requisitos que implicam penalizações

O incumprimento de um ou mais dos seguintes requisitos implica a atribuição da penalização especificada e a automática impossibilidade de obter uma nota superior a 17.

Req. 8 - Regras de estilo e elegância do código (-4,0 valores) O código entregue deve respeitar as regras de estilo, nomeadamente **todas** as seguintes:

Comentários Antes de cada método deve escrever um comentário javadoc (`/** */`) que explique o que o método faz, os respectivos parâmetros e valor devolvido (se existentes). Os comentários podem estar em português mas tente colocá-los em inglês. Os comentários têm de incluir, se aplicáveis, a *tag* `@return`, bem como uma *tag* `@param` para cada parâmetro.

Identificadores em inglês Os nomes de todas as variáveis, métodos e classes devem estar em inglês.

Nomes das variáveis, constantes e classe Utilização de letras minúsculas/-maiúsculas e informação transmitida pelos nomes; por exemplo, `turtle` é um melhor nome para uma tartaruga do que `a` ou `xyz`.

Alinhamento das chavetas Cada uma por baixo da correspondente.

Os espaçamentos Depois das vírgulas e antes e depois dos operadores.

Indentação coerente e para cada bloco.

Utilização do `this` Utilização da referência **this** antes do nome das operações que se aplicam ao objecto da janela ("Mundo"). Por exemplo, `this.add(turtle)`.

Req. 9 - Auto-avaliação (-4,0 valores) No ficheiro `.zip` entregue deve conter um ficheiro de texto com o nome "auto-aval.txt" que indica quais os requisitos que estão **totalmente** cumpridos (os únicos que contam como cumpridos) e a

classificação resultante. No mesmo ficheiro **deve indicar quantas horas gastou a fazer este trabalho, incluindo o tempo em aulas e fora das aulas (trabalho autónomo)**.

Req. 10 - Identificação (-4,0 valores) O ficheiro com código (ficheiro *.java) tem de conter, em comentário no código Java™, o nome e número de aluno do autor.

Req. 11 - Quantidade de autores (-20,0 valores) O trabalho a entregar tem de ser realizado por um único aluno.

Req. 12 - Nome do projecto (-3 valores) O nome do projecto em eclipse (abaixo referido como nomeDoProjecto) tem de respeitar o seguinte formato. Note que Primeiro e Ultimo representam o primeiro e o último nome do autor. Numero representa o número de aluno do autor.

Numero_PrimeiroUltimo_TP1_P1_2009-2010

Por exemplo: 1232_AnaGomes_TP1_P1_2009-2010

O trabalho é entregue compactando a directoria do projecto BlueJ num ficheiro .zip de forma a que este fique com o nome nomeDoProjecto.zip.

Req. 13 - Entrega (até -20,0 valores) O trabalho entregue tem de ser uma directoria do projecto BlueJ pronto a funcionar, sob a forma de um único ficheiro zip contendo toda a directoria mais o ficheiro de auto-avaliação. Antes de entregar, verifique que sabe pôr a funcionar o código no ficheiros zip entregue. Para tal parta desse ficheiro, descompacte-o, e leia-o no BlueJ. Tal poderá ser requerido na apresentação individual do trabalho. Se não conseguir pôr a funcionar o conteúdo do ficheiro zip entregue (no moodle e por e-mail) a classificação no trabalho poderá ser de zero valores.

A entrega tem de ser feita num ficheiro no formato zip com o nome nomeDoProjecto.zip e por duas vias:

1. Na página da disciplina.
2. Por *e-mail*, respeitando as seguintes regras: A entrega por *e-mail* é feita para trabalhos.p1ARROB@gmail.com. O *e-mail* a enviar deve conter em *attach* um único ficheiro no formato *zip*. O *subject* do *e-mail* deve ser o nomeDoProjecto.
3. Pode entregar mais do que uma vez, desde que dentro do prazo. A última entrega dentro do prazo é a única que conta.

Req. 14 - Data limite de entrega (-4,0 valores ou -20,0 valores) O trabalho deve ser entregue no moodle e por e-mail até às **23:55 de 2 de Novembro de 2009**. Os trabalhos entregues entre as 23:55 de 2 de Novembro e as 23:55 de 3 de Novembro de 2009 terão uma penalização de 4,0 valores. **Os trabalhos que não tenham sido entregues até às 23:55 de 3 de Novembro de 2009, nem no moodle nem por e-mail, serão considerados como não entregues e avaliados com zero valores.**

Finalmente, note que necessita de realizar mais do que o pedido para obter mais de 17 valores. A criatividade também pode justificar uma melhor classificação pelo que extras originais e sofisticados serão uma boa aposta. Note que estas adições só contam para a classificação do trabalho se forem consideradas suficientemente significativas e se **todos** os requisitos estiverem completamente cumpridos e sem penalizações.

6 Nota importante

Todas as contribuições para o trabalho que não sejam da exclusiva responsabilidade dos autores têm de ser identificadas (com comentários no código e referências no relatório) e a sua utilização bem justificada e expressamente autorizada pelo professor responsável. Justificações insatisfatórias, ausência de autorização, ou ausência de referências para trabalhos ou colaborações externas utilizadas serão consideradas fraude sempre que os trabalhos sejam considerados demasiado semelhantes para terem sido criados de forma independente. **Tal terá como consequência a reprovação na unidade curricular de todos os alunos envolvidos.** Excepcionalmente, poderão ser penalizados apenas os alunos que se declarem como únicos culpados. Assim, nenhum aluno deve dar cópia do seu código (ainda que em fase inicial) a outro. Por essa mesma razão não é boa ideia partilhar código com os colegas, quer directamente quer através do fórum. Naturalmente, cada aluno pode trocar impressões e esclarecer dúvidas com todos os colegas, mas deve saber escrever todo o código sozinho. Lembre-se que será avaliado num exame prático final em frente a um computador. A classificação neste trabalho prático fica dependente de uma eventual apresentação individual do mesmo, tal como previsto no guia da unidade curricular.

Finalmente, antes de entregar leia com MUITA atenção todo o enunciado. A falha de parte do exigido num requisito implica o não cumprimento desse requisito e consequente penalização. A programação é também a atenção aos detalhes.

Bom trabalho!

João Paulo Barros