

# CHECKLIST DE PROGRAMAÇÃO POR OBJECTOS EM JAVA5

ABRIL DE 2008

MATÉRIA LECCIONADA:

PARADIGMA DA POO, ENCAPSULAMENTO, PROGRAMAÇÃO E CONCEPÇÃO EM  $x.f()$  vs.  $f(x)$   
OBJECTOS, MENSAGENS, CLASSES, INSTÂNCIAS, MÉTODOS DE INSTÂNCIA, CONSTRUTORES  
MÉTODOS ADICIONAIS STANDARD: `toString()`, `equals()` E `clone()`  
COLECÇÕES DE JCF, TIPOS PARAMETRIZADOS, `Collection<E>`  
ITERADOR `for(each)` E `Iterator<E>`

1.- AS MINHAS CLASSES TÊM UM IDENTIFICADOR COMEÇADO SEMPRE POR LETRA MAIÚSCULA, AINDA QUE OS OUTROS CARACTERES SEJAM EM GERAL MINÚSCULOS, MAS NÃO NECESSARIAMENTE (CF. `Ponto2D`, `Aluno`, `Plano`, ETC.)

SIM     NÃO

2.- AS MINHAS VARIÁVEIS DE INSTÂNCIA, OS MEUS MÉTODOS DE INSTÂNCIA E AS MINHAS VARIÁVEIS LOCAIS TÊM SEMPRE UM IDENTIFICADOR COMEÇADO POR LETRA MINÚSCULA, AINDA QUE OS OUTROS CARACTERES POSSAM SER MAIÚSCULOS:

SIM     NÃO

3.- AS MINHAS CONSTANTES SÃO SEMPRE IDENTIFICADAS POR LETRAS MAIÚSCULAS, POR VEZES USANDO `_`, E SÃO SEMPRE DECLARADAS COM O ATRIBUTO `final`:

SIM     NÃO

4.- AS MINHAS CLASSES POSSUEM SEMPRE, PARA ALÉM DO CONSTRUTOR VAZIO, PELO MENOS, MAIS 2 CONSTRUTORES: POR PARTES E DE CÓPIA.

SIM     NÃO

5.- OS MEUS CONSTRUTORES NUNCA DEIXAM NENHUMA VARIÁVEL DE INSTÂNCIA POR INICIALIZAR.

SIM     NÃO

6.- NAS MINHAS CLASSES, OS MÉTODOS DE INSTÂNCIA QUE SÃO PARA SER USADOS EXTERNAMENTE (API) SÃO TODOS DE TIPO DE ACESSO `public`.

SIM     NÃO

7.- NAS MINHAS CLASSES, OS MÉTODOS QUE NÃO SÃO PARA SER USADOS EXTERNAMENTE (API) E SÃO APENAS AUXILIARES À ESCRITA DO CÓDIGO DE OUTROS MÉTODOS SÃO TIPO DE ACESSO `private`.

SIM     NÃO

8.- NAS MINHAS CLASSES, OS MÉTODOS DE INSTÂNCIA OU SÃO DE CONSULTA, E DEVOLVEM UM RESULTADO DE TIPO `T`, OU SÃO DE MODIFICAÇÃO DO ESTADO E DEVOLVEM `void`. NÃO USO MÉTODOS MISTOS, EMBORA SAIBA QUE ALGUNS MÉTODOS DE JAVA SÃO DESTE TIPO (CF. `boolean add(Object o)`), ETC.

SIM     NÃO

9.- AS MINHAS CLASSES DE NOME `X` POSSUEM TODAS UM MÉTODO `public X clone()`, UM MÉTODO `public String toString()` E UM MÉTODO `boolean equals(Object o)`.

SEMPRE     NEM SEMPRE     NUNCA

10.- SEI QUE SE NAS MINHAS CLASSES NÃO PROGRAMAR OS MÉTODOS ANTERIORES, SERÃO USADOS MÉTODOS COM A MESMA ASSINATURA, MAS GENÉRICOS E POUCO INTERESSANTES QUE ESTÃO PROGRAMADOS NA CLASSE `Object`:

SIM     NÃO

11.- AS VARIÁVEIS DE INSTÂNCIA DEFINIDAS NAS MINHAS CLASSES TÊM TODAS ATRIBUTO DE ACESSO `private`, E EU SEI EXACTAMENTE PORQUE DEVEM SER DECLARADAS ASSIM.

SIM     NÃO

12.- SEI QUE NÃO SEREI APROVADO À DISCIPLINA SE INTRODUIZIR CÓDIGO DE I/O COM O UTILIZADOR NOS MÉTODOS DE INSTÂNCIA, E QUE TODO O I/O COM O UTILIZADOR DEVE SER PROGRAMADO NO PROGRAMA PRINCIPAL APENAS, DE FORMA GRÁFICA OU NÃO:

SIM     NÃO

13.- ESCREVO O CÓDIGO DOS MEUS MÉTODOS DE FORMA INDENTADA E COMENTO-O USANDO FORA DO CORPO DOS MÉTODOS DIRECTIVAS COMO `/** ... */` PARA GERAÇÃO DE DOCUMENTAÇÃO EM `javadoc`, E `//` NO CORPO DESTES.

SEMPRE     NEM SEMPRE     NUNCA

14.- AS MINHAS VARIÁVEIS SÃO SEMPRE INICIALIZADAS ANTES DE SEREM UTILIZADAS, EM ESPECIAL AS QUE SÃO DE TIPO REFERENCIADO, QUE INICIALIZO A `null` SE NÃO REFEREM UMA COLECÇÃO, OU USANDO UM CONSTRUTOR ADEQUADO CASO SE TRATE DE UMA COLECÇÃO.

SIM     NÃO

15.- SEMPRE QUE UM MÉTODO DEVOLVE UMA PARTE OU O TODO DE UMA VARIÁVEL DE INSTÂNCIA, USO O MÉTODO `clone()` DEFINIDO NA CLASSE RESPECTIVA PARA QUE SEJA DEVOLVIDA UMA CÓPIA DE TAL OBJECTO, E SEI PORQUE O DEVO FAZER SEMPRE.

SIM     NÃO

16.- SEMPRE QUE UM MÉTODO RECEBE UM PARÂMETRO OBJECTO QUE VAI SER USADO PARA ALTERAR O TODO OU PARTE DE UMA DAS VARIÁVEIS DE INSTÂNCIA, ANTES DE REALIZAR A MODIFICAÇÃO FAÇO UMA CÓPIA DE TAL OBJECTO USANDO O RESPECTIVO MÉTODO `clone()`, E SEI PORQUE O DEVO FAZER SEMPRE.

SIM     NÃO

17.- SEMPRE QUE OS MEUS MÉTODOS USAM COLECÇÕES PARÂMETRO CUJOS OBJECTOS VÃO MODIFICAR AS MINHAS VARIÁVEIS DE INSTÂNCIA, NÃO USO OPERAÇÕES PRÉ-DEFINIDAS DAS APIS DE COLECÇÕES, TAIS COMO OS CONSTRUTORES QUE ACEITAM COLECÇÕES OU `addAll()`, E PROGRAMOAS MODIFICAÇÕES USANDO `for(each)` E `clone()`.

SIM     NÃO

18.- SEI PROGRAMAR “POR CONTRATO” E SEI DISTINGUIR UMA OPERAÇÃO TOTAL DE UMA OPERAÇÃO PARCIAL. SEMPRE QUE TENHO QUE PROGRAMAR UMA OPERAÇÃO PARCIAL CRIO A RESPECTIVA `pré-condição` ESCRIVENDO UM MÉTODO QUE TESTA A CONDIÇÃO DE PARCIALIDADE E QUE DEVOLVE `boolean`, E COLOCO-A NA API.

SIM     NÃO

19.- SEI CRIAR PROGRAMAS PRINCIPAIS QUE QUANDO INVOCADOS NO AMBIENTE BLUEJ CRIAM NO RESPECTIVO “INSTANCE BENCH” INSTÂNCIAS COMPLEXAS DE UMA QUALQUER CLASSE COMPLEXA (QUE DEMORARIAM MUITO TEMPO A CRIAR DIRECTAMENTE EM BLUEJ), MESMO USANDO **TOOLS/ USE CLASS LIBRARY**.

SIM     NÃO

20.- OS MEUS PROGRAMAS COMPILAM SEM “WARNINGS”. NÃO USO NUNCA COLECÇÕES “RAW” DE JAVA4 (CF. **new ArrayList()**; OU **new HashMap()**), NEM USO NENHUM DOS MÉTODOS **clone()** PRÉ-DEFINIDOS PARA COLECÇÕES. ASSIM, É POUCO PROVÁVEL TER “WARNINGS”.

SIM     NÃO

---

**NOTA:** NÃO HÁ UMA PONTUAÇÃO IGUAL PARA CADA QUESTÃO PORQUE O GRAU DE IMPORTÂNCIA É DIFERENTE, E TODOS OS ALUNOS SABEM DISSO. TRATA-SE ASSIM DE UMA AUTO-AVALIAÇÃO.

EM CASO DE DÚVIDA RELATIVAMENTE ÀS SUAS RESPOSTAS CONTACTAR O RESPONSÁVEL DA DISCIPLINA ([fmm@di.uminho.pt](mailto:fmm@di.uminho.pt)).