

CONCOURS EXTERNE ET INTERNE DES 14, 15 ET 16 JANVIER 2014

**POUR L'EMPLOI D'INSPECTEUR DES DOUANES ET DROITS INDIRECTS
DANS LA SPÉCIALITÉ « TRAITEMENT AUTOMATISÉ DE L'INFORMATION –
PROGRAMMEUR SYSTÈME D'EXPLOITATION »**

ÉPREUVE ÉCRITE D'ADMISSIBILITÉ N° 4 (FACULTATIVE)

(DURÉE : 2 HEURES - COEFFICIENT 3)

**TRADUCTION SANS DICTIONNAIRE D'UN TEXTE TECHNIQUE
RÉDIGÉ EN ANGLAIS**

AVERTISSEMENTS IMPORTANTS

L'usage de tout document autre que le support fourni est interdit.

L'usage de tout matériel autre que celui d'écriture est interdit.

Toute fraude ou tentative de fraude constatée par la commission de surveillance entraînera l'exclusion du concours.

Il vous est interdit de quitter définitivement la salle d'examen avant le terme de la première heure.

Le présent document comporte 2 pages numérotées.

JAVA LANGUAGE SPECIFICATION : EXCEPTIONS

When a program violates the semantic constraints of the Java programming language, the Java Virtual Machine signals this error to the program as an exception.

An example of such a violation is an attempt to index outside the bounds of an array. Some programming languages and their implementations react to such errors by peremptorily terminating the program; other programming languages allow an implementation to react in an arbitrary or unpredictable way. Neither of these approaches is compatible with the design goals of the Java SE platform : to provide portability and robustness.

Instead, the Java programming language specifies that an exception will be thrown when semantic constraints are violated and will cause a non-local transfer of control from the point where the exception occurred to a point that can be specified by the programmer.

An exception is said to be thrown from the point where it occurred and is said to be caught at the point to which control is transferred.

Programs can also throw exceptions explicitly, using throw statements (...).

Explicit use of throw statements provides an alternative to the old-fashioned style of handling error conditions by returning funny values, such as the integer value -1 where a negative value would not normally be expected. Experience shows that too often such funny values are ignored or not checked for by callers, leading to programs that are not robust, exhibit undesirable behavior, or both.

Every exception is represented by an instance of the class throwable or one of its subclasses (...). Such an object can be used to carry information from the point at which an exception occurs to the handler that catches it. Handlers are established by catch clauses of try statements (...).

During the process of throwing an exception, the Java Virtual Machine abruptly completes, one by one, any expressions, statements, method and constructor invocations, initializers, and field initialization expressions that have begun but not completed execution in the current thread. This process continues until a handler is found that indicates that it handles that particular exception by naming the class of the exception or a superclass of the class of the exception (...). If no such handler is found, then the exception may be handled by one of a hierarchy of uncaught exception handlers (...) - thus every effort is made to avoid letting an exception go unhandled.

The exception mechanism of the Java SE platform is integrated with its synchronization model (...), so that monitors are unlocked as synchronized statements (...) and invocations of synchronized methods (...) complete abruptly.

The Java Language Specification, Java SE 7 Edition (Java Series), Ch 11, ISBN-13: 978-0133260229