KATHOLIEKE UNIVERSITEIT LEUVEN

Department of Computer Science

Defensive Programming

- Specification
- Implementation
- Verification

Assignment

KATHOLIEKE UNIVERSITEIT LEUVEN

- Develop a class of rational numbers in which all exceptional cases are handled defensively
  - A constructor initializing a new rational number with given numerator and denominator
  - Inspectors returning the numerator, respectively the denominator of a rational number
  - An inspector checking whether two rational numbers have the same value
  - A method for multiplying a rational number with a given integer number
  - …
- An extended set of mathematical methods is given

Overview

KATHOLIEKE UNIVERSITEIT LEUVEN

- Defensive programming
  - Signal illegal invocations of methods by throwing exceptions
    - The user of a method can catch the exception, and proceed with a corrected invocation of the method
- Specification of exceptions
  - Specify all the exceptions that can be signaled by a method
    - The documentation will reveal conditions under which exceptions can or must be thrown
- Throwing and catching exceptions
  - Implement methods by throwing and catching exceptions whenever appropriate
- Epilogue

Defensive Programming

KATHOLIEKE UNIVERSITEIT LEUVEN

Task 1+2

- Signal illegal invocations of methods by means of exceptions
  - Users of a class can catch exceptions, and try to correct the error such that normal processing can be resumed
    - Users of a class must not be confused with end-users of software systems
  - In the implementation of a method, the developer can throw exceptions under conditions stated in the documentation
    - The documentation of a method is complemented with clauses describing under which conditions exceptions must be thrown
- Defensive programming is an alternative way to deal with special (abnormal) cases
  - Special cases can also be handled in the effect of the method (total programming) or by imposing preconditions (nominal programming)
    - No proper guidelines exist when to use these paradigms
  - Defensive programming is said to improve the robustness of code

Overview

KATHOLIEKE UNIVERSITEIT LEUVEN

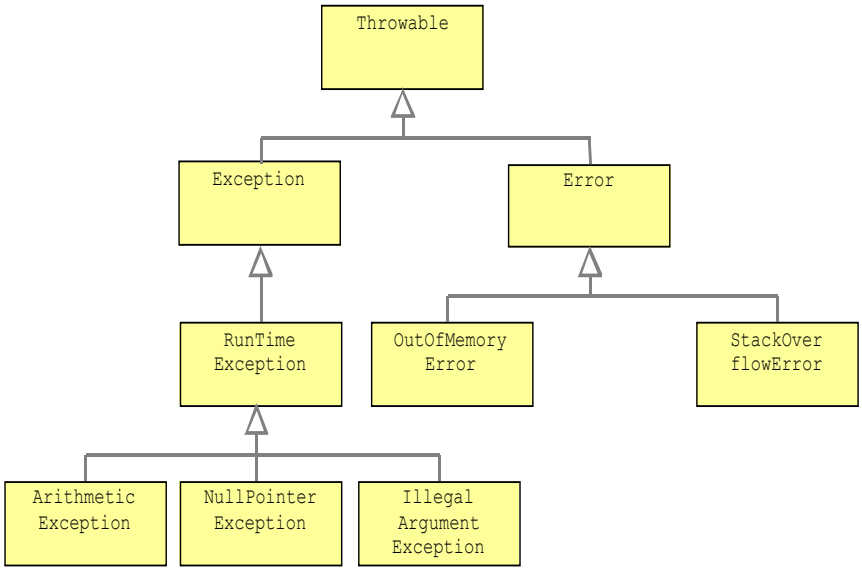
- Defensive programming
  - Signal illegal invocations of methods by throwing exceptions
    - The user of a method can catch the exception, and proceed with a corrected invocation of the method
- Specification of exceptions
  - Specify all the exceptions that can be signaled by a method
    - The documentation will reveal conditions under which exceptions must be thrown
- Throwing and catching exceptions
  - Implement methods by throwing and catching exceptions whenever appropriate
- Epilogue

Method Signature

KATHOLIEKE UNIVERSITEIT LEUVEN

- The signature of a method is complemented with a list of all exceptions that must be thrown by that method
  - Java distinguishes between checked exceptions and unchecked exceptions
    - Runtime exceptions and errors are unchecked exceptions; all other exceptions are checked exceptions
    - The Java compiler checks whether method invocations deal with checked exceptions
  - In this course, a method will list all the exceptions it can throw, except for errors
    - Java does not impose an enumeration of unchecked exceptions in the throws list
      - Only checked exceptions must be listed
    - We do not enumerate errors, because almost all methods can terminate with an error (stack overflow, out of object memory, …)

Hierarchy of Exception Classes



Specification of Exceptions

- The documentation of a method reveals the conditions under which an exception must be thrown
  - A method may not terminate in a normal way under conditions that exceptions must be thrown
    - If the conditions for throwing several exceptions are satisfied, one of them is thrown arbitrarily
  - Whenever a method terminates with an exception, the state of all objects involved is left untouched
- Documentation concerning exceptions is worked out in the heading of the method
  - The documentation starts with one of the standard tags "@throws" or "@exception" followed by the name of the exception class
    - The documentation further describes the conditions under which the exception must be thrown
  - *In the chapter on the substitution principle of Liskov, the formalism to document exceptions is further extended*

Exception Classes

KATHOLIEKE UNIVERSITEIT LEUVEN

- The definition of an exception class is similar to the definition of an ordinary class
  - Self-defined exception classes will specify the predefined class "RuntimeException" as their direct or indirect superclass
    - In Java, all exception classes must inherit directly or indirectly from the class "Throwable"
    - Self-defined exception classes defining checked exceptions would inherit from "Exception"
  - As for ordinary classes, exception classes may introduce their own methods and variables
    - Exception classes inherit methods related to a diagnostic message, to a cause and to the stack trace from their root class "Throwable"

Task 3

Overview

KATHOLIEKE UNIVERSITEIT LEUVEN

- Defensive programming
  - Signal illegal invocations of methods by throwing exceptions
    - The user of a method can catch the exception, and proceed with a corrected invocation of the method
- Specification of exceptions
  - Specify all the exceptions that can be signaled by a method
    - The documentation will reveal conditions under which exceptions can or must be thrown
- Throwing and catching exceptions
  - Implement methods by throwing and catching exceptions whenever appropriate
- Epilogue

Throw statement

KATHOLIEKE UNIVERSITEIT LEUVEN

- The Java Virtual Machine may throw exceptions during the execution of a program
  - These exceptions are typically errors or runtime exceptions
- A program may explicitly throw exceptions by means of throw statements
  - The throw statement involves an expression whose evaluation yields the object to be thrown

Catchers

KATHOLIEKE UNIVERSITEIT LEUVEN

- In Java, code can be included in try-catch-finally constructs
  - The "normal code" is worked out as part of the try-clause
    - If no exceptions are thrown, execution proceeds with the statements in the finally-clause, if any
    - If during the execution of the try-clause an exception is thrown, a proper catcher for that exception is searched
  - Catchers may be worked out, each of them trying to solve the problem signaled by particular exceptions
    - If the execution of a catcher does not terminate in an abnormal way, execution proceeds with the statements in the finally-clause, if any
    - If during the execution of a catcher an exception is thrown again, a search for a proper catcher for that exception is started
    - Since Java 1.7, a catcher can be used to catch several exceptions
  - A finally-clause may be worked out, involving statements that must be executed at all times
    - Finally-clauses are typically used to release resources such as files and network connections

Try-catch-finally

KATHOLIEKE UNIVERSITEIT LEUVEN

Task 4+5+6+7

```
try {
   statement1;
   …
   statementn;
}
catch (Exception11 | … | Exception1kexc) {
  statementc11; … statementc1k;
}
…
catch (Exceptionn1 | … | Exceptionnlexc) {
   statementc11; … statementc1l;
}
finally {
   statementf1; … statementfm;
}
```

Overview

KATHOLIEKE UNIVERSITEIT LEUVEN

- Defensive programming
  - Signal illegal invocations of methods by throwing exceptions
    - The user of a method can catch the exception, and proceed with a corrected invocation of the method
- Specification of exceptions
  - Specify all the exceptions that can be signaled by a method
    - The documentation will reveal conditions under which exceptions can or must be thrown
- Throwing and catching exceptions
  - Implement methods by throwing and catching exceptions whenever appropriate
- Epilogue

Summary

KATHOLIEKE UNIVERSITEIT LEUVEN

- Defensive Programming
  - Use exceptions to signal illegal invocations of methods
- Specification
  - The heading of a method enumerates all exceptions it can throw
  - The documentation of a method reveals the conditions under which a method must throw stated exceptions
- Implementation
  - Exceptions can be thrown explicitly by means of the throw statement or implicitly by the Java Virtual Machine
  - Exceptions can be handled in catchers, associated with try-catch blocks
- Verification
  - Black-box tests must include cases to test the correct throwing of exceptions

Homework

KATHOLIEKE UNIVERSITEIT LEUVEN

- Add the following methods to the class of rational numbers
  - A constructor initializing a new rational number to 0
  - A constructor initializing a new rational number with given denominator
    - The numerator of the new rational number is set to 1 by this constructor
  - A method returning a textual representation of a rational number
  - A method returning a copy of a rational number
  - A method for adding two given rational number
    - Restrict yourself to a simple version in which the resulting number is not simplified