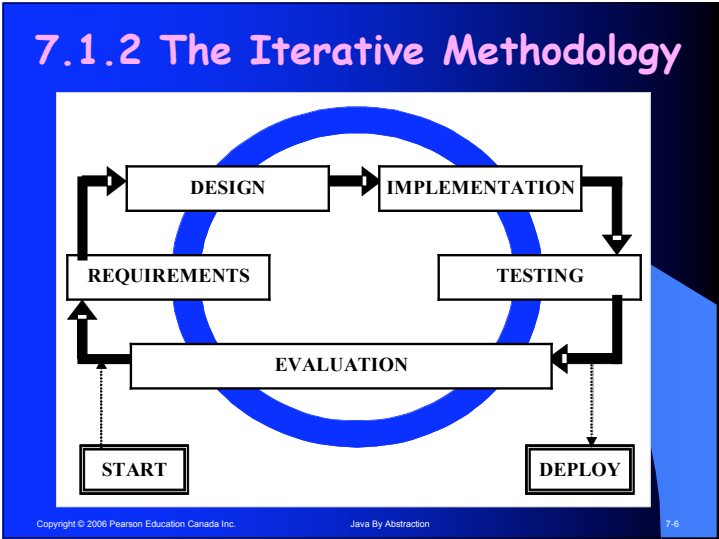


RISKS

- Architectural Risks
- Requirement Change Risks
- Assumptions Risks

What's is needed is:
Risk mitigation through early exposure.

Copyright © 2006 Pearson Education Canada Inc. Java By Abstraction 7-5



Iterative + Incremental

↓ ↓

as a process in use-cases

Copyright © 2006 Pearson Education Canada Inc. Java By Abstraction 7-7

7.1.3 Elements of UML

```

classDiagram
    class Fraction {
    }
    
```

type::lib::Fraction

This is a simple Class Diagram. It is appropriate for early iterations

Copyright © 2006 Pearson Education Canada Inc. Java By Abstraction 7-8

UML

```

type::lib::Fraction
+ isQuoted: boolean
+ separator: char
    
```

A Class Diagram with more details.

Copyright © 2006 Pearson Education Canada Inc. Java By Abstraction 7-9

UML

```

type::lib::Fraction
+ isQuoted: boolean
+ separator: char

+ getNumerator(): long
+ setFraction(Fraction)
+ toString(): String
    
```

A Class Diagram with still more details.

Copyright © 2006 Pearson Education Canada Inc. Java By Abstraction 7-10

UML

A \dashrightarrow **B** Dependency: A uses B
A \diamond **B** Aggregation: A has a B
A \triangle **B** Inheritance: A is a B

Relationship Diagrams

Copyright © 2006 Pearson Education Canada Inc. Java By Abstraction 7-11

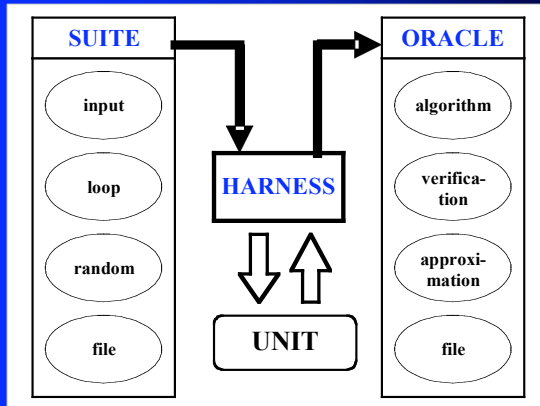
A Multi-class Application

```

classDiagram
    class App
    class GlobalCredit
    class CreditCard
    class RewardCard
    App ..> GlobalCredit
    App ..> CreditCard
    App ..> RewardCard
    GlobalCredit o-- CreditCard
    RewardCard <|-- CreditCard
    
```

Copyright © 2006 Pearson Education Canada Inc. Java By Abstraction 7-12

7.2.1 The Essence of Testing



Copyright © 2006 Pearson Education Canada Inc.

Java By Abstraction

7-13

Example

Describe an appropriate oracle for each of the following scenarios:

1. A program that sorts a list of numbers in non-descending order.
2. A program that determines if a given positive integer is prime.
3. A program that finds the median of a set of an odd number of numbers.

Copyright © 2006 Pearson Education Canada Inc.

Java By Abstraction

7-14

7.2.2 The Test Vector

- Domain Coverage
Black-Box Testing
- Execution-Path Coverage
White-Box Testing
- Regression
The Bug-Recycling Phenomenon

Copyright © 2006 Pearson Education Canada Inc.

Java By Abstraction

7-15

7.2.3 Debugging

Consider this digit detector:

```
boolean containsDigits = false;
for (int i = 0; i < str.length(); i++)
{
    char c = str.charAt(i);
    containsDigits = Character.isDigit(c);
}
output.println(containsDigits);
```

Copyright © 2006 Pearson Education Canada Inc.

Java By Abstraction

7-16

Debugging

Testing Digit Detector:

Test Case	Output
"1234"	true
"F304"	true
"TEST"	false
"99MM"	false

Copyright © 2006 Pearson Education Canada Inc. Java By Abstraction 7-17

Debugging

Debugging Digit Detector:

```
boolean containsDigits = false;
output.println("Test case: " + str);
for (int i = 0; i < str.length(); i++)
{
    char c = str.charAt(i);
    containsDigits = Character.isDigit(c);
    output.println(i + " " + containsDigits);
}
output.println(containsDigits);
```

Copyright © 2006 Pearson Education Canada Inc. Java By Abstraction 7-18

Debugging

Debugging Output:

```
99MM
0 true
1 true
2 false
3 false
false
```

Copyright © 2006 Pearson Education Canada Inc. Java By Abstraction 7-19

Debugging

The Fix: update the boolean cumulatively

```
boolean containsDigits = false;
for (int i = 0; i < str.length(); i++)
{
    char c = str.charAt(i);
    containsDigits = containsDigits || Character...
}
output.println(containsDigits);
```

Or exit the loop early:

```
for (int i = 0; i < .. && !containsDigits...
```

Copyright © 2006 Pearson Education Canada Inc. Java By Abstraction 7-20

Debugging Techniques

- **Print**, print, and then print
- **Read** error messages
- **Assert** your convictions
- Use a **debugger**

Copyright © 2006 Pearson Education Canada Inc.

Java By Abstraction

7-21

Example

What does error message say?

```
java.lang.StringIndexOutOfBoundsException:
String index out of range: -1
    at
    java.lang.String.substring(String.java:1438)
    at
    java.lang.String.substring(String.java:1411)
    at Exercise_7.main(Exercise_7.java:15)
Exception in thread "main"
```

Copyright © 2006 Pearson Education Canada Inc.

Java By Abstraction

7-22

7.3 Case Study

The Trajectory case study (Section 7.3) demonstrates the iterative methodology. In addition, it exposes the string tokenizer class.

Here, we will explain string tokenization and outline a different case study: A real-time weather monitor.

Copyright © 2006 Pearson Education Canada Inc.

Java By Abstraction

7-23

The StringTokenizer Class

- Resides in **java.util**
- Useful for **parsing**:
 - Sentences into words
 - Records into fields
 - Whenever delimiters are known
- Used for **interoperability**
Spreadsheet, databases, ...
Tab-delimited, CSV, ...

Copyright © 2006 Pearson Education Canada Inc.

Java By Abstraction

7-24

The StringTokenizer API

- Two Constructors:
 - String to parse
 - String and delimiters
- `nextToken()`*
- `hasMoreTokens()`

* Overloaded: `nextToken(delimiters)`

Copyright © 2006 Pearson Education Canada Inc.

Java By Abstraction

7-25

Example

What is wrong with this fragment?

```
StringTokenizer stk;
stk = new StringTokenizer(str);
for (String token = stk.nextToken();
     stk.hasMoreTokens();
     token = stk.nextToken())
{
    ...
}
```

Copyright © 2006 Pearson Education Canada Inc.

Java By Abstraction

7-26

Case Study

A Real-Time Weather Monitor

Write an app that monitors the weather in a city and alerts you when the temperature exceeds a certain value.

- Find a site that reports the current weather
- Note the site's URL when the city is selected
- View the HTML source of the weather page
- Tokenize the page to isolate the sought temperature

See Lab 7 (7.3) for details

Copyright © 2006 Pearson Education Canada Inc.

Java By Abstraction

7-27

Additional Exercises

- Write an app that monitor the price of oil, or gold, ...
- Write an app that determines the current CAD/USD FX rate.

Note:

In order to have your program run (and poll the site) every few minutes, use the scheduler program of your operating system.

Copyright © 2006 Pearson Education Canada Inc.

Java By Abstraction

7-28