CSE1720

Week 13, Class Meeting 33 (Lecture 23)

Winter 2013 ◆ Tuesday, April 2, 2013

YORK UNIVERSITÉ UNIVERSITY

Topics

- preparation for exception handling questions

YORK UNIVERSITÉ UNIVERSITY

Lab Exercises

- Assigned **Exercises** pp. 436-439
- 11.1 – 11.7
- 11.10, 11.11, 11.12
- Labs this week will provide support for these exercises

- Final Exam Labtest:
  - will have a focus on exceptions
  - some questions will be in the style of the Chapter 11 exercises
  - other questions will concern the game code base and exceptions

YORK UNIVERSITÉ UNIVERSITY

Conceptual Questions

- Assigned **Review Questions**, p. 431
- questions 1 - 26

YORK UNIVERSITÉ UNIVERSITY

Sample Questions

- What is the difference between a *checked* and an *unchecked* exception?

Sample Questions

- What does a checked exception encapsulate?
- What does an unchecked exception encapsulate?

YORK UNIVERSITÉ UNIVERSITY

Sample Questions

- What is the acknowledgement rule? How and when is it applied?

Sample Questions

- When an exception is thrown, is this an indication that the contract of a method has been violated?

Sample Questions

- Consider the app shown on the next slide. This app compiles without error.
  - (a) When this app runs, state what will happen. The API for the substring(int, int) method is provided. If an exception is thrown, who threw it and at what line of code? Who, if anyone, handles the exception?
  - (b) Has the contract been violated? Why or why not?

Sample Questions

substring

```
public String substring(int beginIndex,
                        int endIndex)
```

Returns a new string that is a substring of this string. The substring begins at the specified `beginIndex` and extends to the character at index `endIndex - 1`. Thus the length of the substring is `endIndex-beginIndex`.

Examples:

```
"hamburger".substring(4, 8) returns "urge"
"smiles".substring(1, 5) returns "mile"
```

Parameters:
`beginIndex` - the beginning index, inclusive.
`endIndex` - the ending index, exclusive.

Returns:
the specified substring.

Throws:
`IndexOutOfBoundsException` - if the `beginIndex` is negative, or `endIndex` is larger than the length of this `String` object, or `beginIndex` is larger than `endIndex`.

```
1  public class AppA {
2
3          public static void main(String[] args) {
4                  String thePhrase = "Hello";
5                  String someSubstring = thePhrase.substring(10, 11);
6                  System.out.println(someSubstring);
7          }
8  }
```

Sample Questions

- Consider the app shown on the next slide.
- This app **does not** compile. A compiler error occurs on line 6:
  `Unhandled exception type FileNotFoundException.`
- Why is it that `AppB` results in a compiler error but `AppA` does not?

Sample Questions

FileInputStream

```
public FileInputStream(String name)
                throws FileNotFoundException
```

Creates a `FileInputStream` by opening a connection to an actual file, the file named by the path name `name` in the file system. A new `FileDescriptor` object is created to represent this file connection.

First, if there is a security manager, its `checkRead` method is called with the `name` argument as its argument.

If the named file does not exist, is a directory rather than a regular file, or for some other reason cannot be opened for reading then a `FileNotFoundException` is thrown.

Parameters:

`name` - the system-dependent file name.

Throws:

`FileNotFoundException` - if the file does not exist, is a directory rather than a regular file, or for some other reason cannot be opened for reading.

`SecurityException` - if a security manager exists and its `checkRead` method denies read access to the file.

```
1  import java.io.FileInputStream;
2
3  public class AppB {
4
5          public static void main(String[] args) {
6                  FileInputStream theFile = new FileInputStream("Names.txt");
7          }
8  }
```