CS367 Announcements
Monday, June 24, 2013

- H1 due Today, Monday, June 24th 6pm (make sure h1.txt file is in 'in' dir by 6)

Last Time

- Iterators
  - review
  - implementing
- ArrayBagIterator
- using the command line

Today

- Topics so far
- Exceptions
- Complexity

Topics So Far

- ADTs
- Interfaces
- Object, casting, autoboxing
- Generics
- (array-based) Lists
- Iterators

Throwing an Exception

Java Syntax

```
throw exceptionObject;
```

Example

Handling an Exception: `try-catch`

Java Syntax

```
try {
    // try block

    ... // code that might cause an exception to be thrown

} catch (ExceptionType1 identifier1) {
    // catch block

    ... // code to handle exception type 1

} catch (ExceptionType2 identifier2) {
    // catch block

    ... // code to handle exception type 2

} ...
finally {
    // finally block
    ...  // code executed no matter what happens in try block

}
```

Example

Handling an Exception: `throws` clause

Java Syntax

```
... methodName( parameter list)
        throws ExceptionType1, ExceptionType2, ... {
      ...
}
```

Example

```
public static void main(String[] args) throws IOException {
   ...
```

Defining an Exception

Define a new class that is a subclass of either:
`Exception` **(checked) or** `RuntimeException` **(unchecked)**

Example

```
public class EmptyBoxException extends Exception {

    public EmptyBoxException() {
        super();
    }

    public EmptyBoxException(String msg) {
        super(msg);
    }

 }
```

Example

```
public static void main(String[] args) throws IOException {
   ...
```

Defining Checked vs. Unchecked Exceptions

Checked:

```
public class MyException extends ____________________________ {
    ...
}
```

Unchecked:

```
public class MyException extends ____________________________ {
    ...
}
```

What really happens when an exception is thrown?

ExceptionTester Example

```
public class ExceptionTester {
    public static void main(String[] args) {
        System.out.print("main[");
        try {

            methodA( ); System.out.print("after A,");
            methodE( ); System.out.print("after E,");

        } catch (RedException exc) {
            System.out.print("red,");
        } catch (GreenException exc) {
            System.out.print("green,");
        }
        System.out.println("]main");
    }
    private static void methodA( ) {
        System.out.print("\nA[");
        try {

            methodB( ); System.out.print("after B,");

        } catch (BlueException exc) {
            System.out.print("blue,");
        }
        System.out.println("]A");
   }
   private static void methodB( ) {
        System.out.print("\nB[");
        try {

            methodC( ); System.out.print("after C,");

        } catch (YellowException exc) {
            System.out.print("yellow,");
            throw new GreenException();
        } catch (RedException exc) {
            System.out.print("red,");
        }
        methodD( );
        System.out.print("after D,");
        System.out.println("]B");
   }
}
```

What Gets Printed When:

1. `methodC` throws a red exception?

```
main[
A[
B[
```

2. `methodD` throws a red exception?

```
main[
A[
B[
```

What Gets Printed When:

3. `methodE` throws a red exception?

```
main[
A[
B[
```

4. `methodC` throws a blue exception?

```
main[
A[
B[
```

What Gets Printed When:

5. `methodC` throws a green exception?

```
main[
A[
B[
```

6. `methodD` throws a green exception?

```
main[
A[
B[
```

What Gets Printed When:

7. `methodC` throws a yellow exception?

```
main[
A[
B[
```

8. `methodD` throws a yellow exception?

```
main[
A[
B[
```

What Gets Printed When:

9. `methodE` throws a yellow exception?

```
main[
A[
B[
```

10. `methodD` throws an orange exception?

```
main[
A[
B[
```

Analyzing Algorithm Efficiency

complexity =