Recitation 2

Exception handling

Exception handling

What could happen without exceptions?

```java
public static double getAverage(double[] b) {
    double sum = 0;
    for (int i = 0; i < b.length; i++) {
        sum += b[i];
    }
    return sum / b.length;
}
```
If `b.length` is 0, what should be returned?
- Infinity
- "special" int - Integer.MAX_VALUE? 2110? 0?

Superclass of exceptions: Throwable

Two subclasses of Throwable exist:
Error: For errors from which one can’t recover –don’t “catch” them
Exception: For errors from which a program could potentially recover –it’s ok to “catch” them

A Throwable instance: ArithmeticException

There are so many different kinds of exceptions we need to organize them.
Throwing an exception

When an exception is thrown, it is thrown to the place of call, which throws it out further to where that method was called. The code that called main will "catch" the exception and print the error message.

Method call: main(new String[] {});

Console:
java.lang.AE: / by zero
at Ex.third(Ex.java:11)
at Ex.second(Ex.java:7)
at Ex.main(Ex.java:3)

AE = ArithmeticException

Decoding the output from an exception

Exception in thread "main" java.lang.ArithmeticException: / by zero
at Animal.main(Animal.java:2)

throw keyword: Forcing a crash

Why might I want to crash the application?

Demo 1: Read an Integer

- Ask the user to input an int
- Try to convert user input to an int
- If an exception is thrown, catch it and ask for more input

Exercise 3: Illegal Arguments

Create class Person with two fields, name and age. Throw an IllegalArgumentException instead of having preconditions when given a null name or a non-positive age.
How to write an exception class

```java
/** An instance is an exception */
public class OurException extends Exception {

    /** Constructor: an instance with message m */
    public OurException(String m) {
        super(m);
    }

    /** Constructor: an instance with no message */
    public OurException() {
        super();
    }
}
```

**throws clause for checked exceptions**

```java
/** Class to illustrate exception handling */
public class Ex {

    public static void main() {
        try {
            second();
        } catch (OurException e) {
            // handle exception
        }
    }

    public static void second() throws OurException {
        third();
    }

    public static void third() throws OurException {
        throw new OurException("mine");
    }
}
```

Key takeaways

Thrown exceptions bubble up the call stack until they are handled by a try-catch block. In the system, the call of method main is in a try-catch statement, and its catch block prints out information about the thrown exception.

Demo 2: Pythagorean Solver

- Given a and b: solve for c in $a^2 + b^2 = c^2$
- Reads input from keyboard
- Handles any exceptions

```
// Given a and b: solve for c in a^2 + b^2 = c^2
// Reads input from keyboard
// Handles any exceptions
```