Recitation 3

2D Arrays, Exceptions

2D arrays

Many applications have multidimensional structures:

- Matrix operations
- Collection of lists
- Board games (Chess, Checkers)
- Images (rows and columns of pixels)
- ...

1D Array Review

Animal[] pets = new Animal[3];

pets.length is 3
pets[0] = new Animal();
pets[0].walk();

Why is the following illegal?
pets[1] = new Object();

Java arrays vs Python lists

Java arrays do not change size!

String[] b = {"Cornell", "Ithaca"};
String[] bBig = Arrays.copyOf(b, 4);
b = bBig;

Java array initialization

Instead of

```java
int[] c = new int[5];
```

Use an array initializer:

```java
int[] c = new int[] {5, 4, 7, 6, 5};
```

Note: The length of c is the number of values in the list.

Exercise 1: Looping over an array

```java
/** Return index of occurrence number n of t in b.
   * Precondition: n >= 1.
   * Return -1 if not found. */
public static int get(int[] b, int n, int t) {
   ... }

get(new int[]{2110, 0, 1, 2110, 2110, 2, 2110});
would return 3
2D arrays: An array of 1D arrays.

Java only has 1D arrays, whose elements can also be arrays.

```
int[][] b = new int[2][3];
```

This array has 2 int[] arrays of length 3 each.

```
0 1
0 0 0 2 0
```

How many rows in `b`?
`b.length`

How many columns in row 0?
`b[0].length`

How many columns in row 1?
`b[1].length`

The elements of `b` are of type int[].

```
int[][] b = new int[2][];
```

Exercise 2: Transpose Matrix

```
A     A^T
|1 1|   |1 1|
|2 2|   |2 3|
|3 3|   |3 3|

A^T[0][0] is A[0][0]
```

Exceptions
Exceptions make your code crash

```java
public static void main(String[] args) {
    System.out.println(args[0]);
}
```

```java
public static void main(String[] args) {
    System.out.println(8 / 0);
}
```

```java
public static void main(String[] args) {
    System.out.println(null.toString());
}
```

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

When some sort of exception occurs, an object of class `java.lang.Throwable` (or one of its subclasses) is created and "thrown"—we explain later what "throw" means.

The object has
1. Field to contain an error message
2. Two constructors
3. Function to get the message in the field

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

```java
public static void main(String[] args) {
    int div = 5/0;
}
```

Exception in thread "main" java.lang.ArithmeticException: / by zero

at Animal.main(Animal.java:2)

Try statement: catching a thrown exception

```
try {
    code (this is the try-block)
} catch (MyException ae) {
    code (this is the catch-block)
} else { code following the try statement}
```

ae is like a parameter. When the catch-block catches a thrown object, ae contains the object.

Demo 1: Read an Integer

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

```java
class Integer {
    /** Parse \texttt{s} as a signed decimal integer. 
     * Throw a NumberFormatException if not possible */
    public static int parseInt(String s) {
        ... if (can't convert to int){
            throw new NumberFormatException();
        }
    }
}
```

Exercise 3: Illegal Arguments

Create \texttt{class Person} with two fields, \texttt{name} and \texttt{age}.

Throw an \texttt{IllegalArgumentException} instead of having preconditions when given a \texttt{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 \texttt{m} */
    public OurException(String m) {
        super(m);
    }
    /** Constructor: an instance with no message */
    public OurException() {
        super();
    }
}
```
**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

**Key takeaways**

1. Java arrays do not extend!
2. A 2D array is just a 1D array of 1D arrays.
3. 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.