Lecture - Math class

Acknolwedgement

It has been transferred from static slides to Ed Stem by Lachlan Andrew, with minor changes.

review

Static methods: Can be called without an object.

Static variables: Shared by all objects of this class.

Math class

(Spelled "Math", not "Maths" because Java is American...)

Math class provides many static methods related to (you guessed it) mathematics.

It is in `java.lang` which is imported by default, and so you do not need an `import` statement.

All of its elements are `static`, and so you will never need to create an object of class `Math`.

It has two constants

- Math.E (the base of the natural logarithms, = Math.exp(1.0))
- Math.PI (π=3.14159265...)

```
area = Math.PI * radius * radius;
```

Some methods

```
public static double  pow(double base, double exponent)
```

Returns base to the power of exponent, e.g., `Math.pow(2.0, 3.0)` is 8.0

```
public static double   sqrt (double argument)
```

Returns the square root of its argument, e.g., `Math.sqrt(4.0)` is 2.0

Overloaded methods

Many of the methods are available with different argument types. They typically produce an answer of the same type as the argument. For example

```
public static double abs(double argument)
public static float  abs(float  argument)
public static long   abs(long   argument)
public static int    abs(int    argument)
```

These return the absolute value of the argument (i.e., if argument < 0, then it returns -argument).

We can summarize these four by letting "T" denote a type, and writing

```
public static T abs(T argument)
```

for `T` in `double`, `float`, `long`, `int`

(You'll see this sort of thing more when we discuss "generics".)

Other overloaded methods are:

```
public static T min (T n1, T n2)
public static T max (T n1, T n2)
```

for T in `double`, `float`, `long`, `int`

These return the minimum of n1 and n2, and the maximum of n1 and n2.

Converting to integers

There are three methods for converting floating point values (`double` and `float`) to integers. (There are four counting overloading.)

Round to nearest

```
public static long round (double argument)
public static int  round (float  argument)
```

These return the integer nearest to the argument.

Positive values with a fractional part 0.5 or higher round up, and values with a fractional part less than 0.5 round down. (For negative values, this is reversed. Have a think about why.)

Note that these return **integer** types (`long` and `int`).

Round down

```
public static double floor (double argument)
```

This rounds down; it returns the largest whole number that isn't bigger than the argument, e.g., `Math.floor(3.5)` = 3.0.

(For negative values, it rounds away from zero. `Math.floor(-3.5)` = -4)

Note that these return `double`. They are not useful for changing the type of data; just for mathematical operations.

Round Up

```
public static double ceil (double argument)
```

This rounds up; it returns the smallest whole number that isn't smaller than the argument. e.g., `Math.ceil(3.5)` = 4.0

The name is short for "ceiling".

Again, for negative values this rounds away from zero. Again, it returns a `double`.

Additional methods

The following are some of the other methods in Math. You are not expected to remember these. This list is here just in case it is useful.

random

sin, cos, tan, asin, acos, atan, sinh, cosh, tanh

toDegrees, toRadians

atan2, hypot

exp, log, log10, expm1, logp1