Delegation & APIs

EXAMPLE OF SERVICES

• Format in String (static)
  Turn something to a string and format it.
  Good for interacting with the UI.

• pow in Math (static)
  Raises a number to an exponent.

• getTime in Date (non-static)
  Number of milliseconds since 1970

String Format API

<table>
<thead>
<tr>
<th>Conversion</th>
<th>%d</th>
<th>Integer (base 10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>%x</td>
<td>Integer (base 16)</td>
<td></td>
</tr>
<tr>
<td>%f</td>
<td>Real, floating point</td>
<td></td>
</tr>
<tr>
<td>%e</td>
<td>Real, exponential</td>
<td></td>
</tr>
<tr>
<td>%c</td>
<td>Character</td>
<td></td>
</tr>
<tr>
<td>%b</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>%A</td>
<td>Day (a for short)</td>
<td></td>
</tr>
<tr>
<td>%B</td>
<td>Month (b for short)</td>
<td></td>
</tr>
<tr>
<td>%Y</td>
<td>Year (y for 2-digit)</td>
<td></td>
</tr>
<tr>
<td>%t</td>
<td>%M, %H, %s</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Escape for %</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String Escapes</th>
<th>%n</th>
<th>New line</th>
</tr>
</thead>
<tbody>
<tr>
<td>%t</td>
<td>tab</td>
<td></td>
</tr>
<tr>
<td>%\</td>
<td>Escape for \</td>
<td></td>
</tr>
</tbody>
</table>

Flag | Thousand separator |
---|-------------------|
>0 | Right-align |
<0 | Left-align |

Flag | Include sign |
---|--------------|
+   | Include sign |
-   | Include sign |

Flag | Leading space if >0 |
---|---------------------|
|    | Leading space if <0 |

<table>
<thead>
<tr>
<th>width</th>
<th>0</th>
<th>Zero fill</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&gt;0</td>
<td>Right-align</td>
</tr>
<tr>
<td></td>
<td>&lt;0</td>
<td>Left-align</td>
</tr>
</tbody>
</table>

APPROACH

• Determine the available services
  The API tells you what a service does, not how it does it.

• Compose your solution accordingly
  Orchestrate what does what you need done...

• Delegate to Method m
  - Static Methods: Invoke on the class
    C.m(...) |
  - Non-Static Methods: Instantiate then invoke on the object.
    C ref = new C(...); ref.m(...);
Math API

Method Summary

- `static double abs(double a)`
  Returns the absolute value of a double value.
- `static int abs(int a)`
  Returns the absolute value of an int value.
- `static double pow(double a, double b)`
  Returns the value of the first argument raised to the power of the second argument.

The Constructor

- A constructor section looks like a method but it has no return and its name is the same as the class.
- There can be more than one (overloaded). The default one creates an object with a default state.
- A class with only a default constructor must have mutators if we are to be able to set a non-default state.
- The constructor is invoked with `new` in order to instantiate the class and create an object of its type.
- A class with no constructor (i.e. a private one) must have all its fields and methods static or else we cannot use it.

Anatomy of an API

The API of a class has three sections:

- A Constructor Section
- A Field Section
- A Method Section

- Each section has a summary and a linked detailed part.
- The job of the constructor is to initialize the state.
- Fields (or public attributes) are used mainly for constants.
- Methods are the ones who "do the work".
- See Links in the course website to basic APIs.

Methods

- Each method has a signature = name and parameter types.
- A class cannot have two methods with the same signature (even if the return is different). Hence, can have two methods with the same name (overloaded).
- To bind `C.m(…)` the compiler locates `C` (or else issues No Class Definition Found) and then locates `m(…)` in `C` (or else issues Cannot Resolve Symbol). If more than one such `m` is found, it binds with the most specific one.
- Parameters are passed by value. Hence values stored in your variables cannot be inadvertently changed by passing the variables to a method.
Invocation syntax is like two-level addressing: country code then number. The dot separates the two.

For static methods, the class is the country. For others, a copy is involved. The object reference is the country here.

Within the object, the address of the object is this.

Instantiation is slow. So if many state changes are needed and when don't need to retain each, a default constructor and a mutator is better than multiple objects.

Controller can thus instantiate in onCreate and mutate when a button is clicked.