Introduction to Object-Oriented Programming in MATLAB®

Jos Martin
Principle Software Engineer
jos.martin@mathworks.co.uk
Goals

- Object-oriented programming
- Basic syntax in MATLAB®
- The MATLAB class system
What is a program?

<table>
<thead>
<tr>
<th>Code</th>
<th>Data</th>
<th>Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>x = 12</code></td>
<td><code>x = 12</code></td>
<td>Assignment</td>
</tr>
<tr>
<td><code>while (x &lt; 100)</code></td>
<td><code>while (x &lt; 100)</code></td>
<td>Looping Test</td>
</tr>
<tr>
<td><code>x = x+1</code></td>
<td><code>x = x+1</code></td>
<td>Increment</td>
</tr>
<tr>
<td><code>if ( x == 23)</code></td>
<td><code>if ( x == 23)</code></td>
<td>Test to Act</td>
</tr>
<tr>
<td><code>disp('Hello')</code></td>
<td><code>disp('Hello')</code></td>
<td>Take Action</td>
</tr>
<tr>
<td><code>end</code></td>
<td><code>end</code></td>
<td>End</td>
</tr>
<tr>
<td><code>end</code></td>
<td><code>end</code></td>
<td></td>
</tr>
</tbody>
</table>
Progression of Programming Techniques

Data

- literal
- variable
- structure

Level of Abstraction / Sophistication

Algorithm

- function
- script
- command line
Example: Sensor Array

- Transmitting a signal from a weather balloon
- Locating the signal with a sensor array
- Computing the angle of arrival for the signal (AoA)
Procedural Programming

- Easy to learn
- Minimal planning
- No formal relationship between data and functions
- Every detail is exposed
Data and Actions to Implement

Data
- Wavelength
- Number
- Spacing
- Location
- Reading
- Frequency

Actions
- Compute FFT
- Determine peaks
- Plot results
- Synthesize measurements

MathWorks
Aerospace and Defence Conference '08
Related Data and Actions

Data

- Wavelength
- Location
- Spacing
- Reading
- Frequency
- Wavelength
- Location
- Spacing
- Reading

Actions

- Compute FFT
- Determine peaks
- Plot results
- Synthesize measurements
- Compute FFT
- Determine peaks
- Plot results
- Synthesize measurements
Grouping Related Items

Data
- Target
  - Location
- Signal
  - Frequency
  - Wavelength

Class
- Sensor
  - Synthesize measurements
  - Determine peaks
- FFT Results

Actions
- Sensor
  - Reading
  - Spacing
  - Number
- FFT
  - Results
Progression of Programming Techniques

Data
- literal
- variable
- structure
- class

Algorithm
- function
- script
- command line
Object-Oriented Terminology

- **Class**
  - Blueprint of an idea
  - *Properties* (data)
  - *Methods* (algorithms)

- **Object**
  - Specific example of a *class*
  - *Instance*
Goals

- Object-oriented programming
- Basic syntax in MATLAB®
- The MATLAB class system
Demonstration: Building a Simple Class

- Define a target class
- Create the weather balloon object
- Use the object in place of the structure
Objects

- Easy to create
- Manage their own data
- Interchangeable with a structure
  - No other code changes required
  - Properties behave similar to field names
  - Can’t add fields arbitrarily
Demonstration: Adding Methods to a Class

- Start from a sensor class with existing properties
- Add a method to compute angle of arrival (AoA)
- Integrate a sensor object into the existing code
Objects with Methods

- Have immediate access to their own data (properties)
- Allow you to overload existing functions
- Allow you to perform custom actions at creation and deletion
Goals

- Object-oriented programming
- Basic syntax in MATLAB®
- The MATLAB class system
The MATLAB Class System

- Designed to ‘feel’ like MATLAB
  - Incorporates matrix indexing
    \[
    \gg x = 2*\text{obj.data}(1:\text{end});
    \]
  - Inherent overloading
    \[
    \text{varargout} = \text{obj.function}(...);
    \]
- Works like an object-oriented language
  - Encapsulation, inheritance, polymorphism, etc.
Taking Methods and Properties Further

- Control access
- Create constants
- Make values interdependent
- Execute methods when properties change

**External Methods**
- Plot results
- Compute AoA

**Internal Methods**
- Synthesize measurements
- Determine peaks
- Compute FFT

**External Data**
- Reading
- Spacing
- Number

**Internal Data**
- Speed of light
- Noise ratio
- etc.
Demonstration: Applying Attributes

- Control access
  - Access = public
  - Access = protected

- Restrict modification
  - Constant
  - Dependent
Encapsulation

- Sensor
- Plot results
- Compute AoA
- Sensor Reading
- Number of Towers
- Tower Spacing
Encapsulation

- Separates the interface from the implementation
- Simplifies object use
- Becomes a building block

Sensor

- Plot results
- Synthesize measurements
- Compute FFT
- Determine Peaks
- Noise Ratio
- Sensor Reading
- Number of Towers
- Tower Spacing
- Speed of Light
- Compute AoA
- Compute AoA
- Tower Spacing

MathWorks
Aerospace and Defence Conference ’08
Using an Object as a Building Block

Assignment
Looping Test
  Increment
  Test to Act
    Take Action
  End
End
Using a Class as a Building Block

The Balloon

All Moving Targets

The Red Baron

All Targets
Demonstration: Creating a Moving Target

- Define a new `class` moving target
  - *Inherit* from the existing `class` target
- Add a `method`
- Use the moving target
Inheritance

- **Subclass** substitutes for the **superclass**
- Allows re-envisioning and re-implementing the **superclass**
- Builds on proven code
- Allows inheriting from the base MATLAB classes
How does ‘=’ work in MATLAB?

Round 1

```matlab
>> a = 10000;
>> b = a;
>> b = 20000;
>> disp(a)
```

a) 10,000
b) 20,000
c) Something else
d) No idea
How does ‘=’ work in MATLAB?

Round 2

```matlab
>> a = analoginput('winsound'); addchannel(a,1);
>> a.SampleRate = 10000;
>> b = a;
>> b.SampleRate = 20000;
>> disp(a.SampleRate)
```

a) 10,000
b) 20,000
c) Something else
d) No idea
>> B = A;

A \rightarrow \text{Data}

A \rightarrow \text{Data}

A \rightarrow \text{Data}

A \rightarrow \text{Data}
Value Class

MATLAB default

'=' copies data

data in workspace

Handle Class

Use: < handle

'=' references data

handle in workspace
Optional Demonstration: Using Events

- Events
  - Created in a handle object
  - `events block in classdef`
  - `notify(...)` triggers event

- Listeners
  - Triggers callback function
  - `addlistener(...)`
  - Useable anywhere
Events and Listeners

- Uses technology related to:
  - preSet
  - postSet
  - preGet
  - postGet

- Gives the ability to trigger action

- Anything can listen to an observable object
The MATLAB Class System

- Class definition file describes object behavior
- Objects can substitute for structures
- Apply attributes for a clean interface
- Build on existing classes with inheritance

Extends the matrix-based language to objects
Questions and Answers