Programming with MATLAB

Paolo Fabbri
Senior Engineer
MATLAB Application Development Landscape

Prototyping  Programming  Deployment
MATLAB Application Development Landscape

Prototyping  Programming  Deployment
MATLAB Application Development Landscape

Prototyping  Programming  Deployment
Today Focus

Object-Oriented Programming
Unit Test Framework
Source Control Integration
Programming Interfaces
What is a program?
What is a program?

```matlab
x = 12
while (x < 100)
    x = x+1
    if (x == 23)
        disp('Hello')
    end
end
```

**Data**

- `x = 12`
- `while (x < 100)`
- `x = x+1`
- `if (x == 23)`
- `disp('Hello')`
- `end`
- `end`

**Operations**

- Assignment
- Looping Test
- Increment
- Test to Act
- Take Action
- End
- End

**Code**
Range of Programming Techniques

Data
- value
- variable
- structure
- (properties)
- class
- (methods)

Operations
- function
- script
- command line
### Classes and Objects

#### People

<table>
<thead>
<tr>
<th>Properties</th>
<th>Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>• FirstName</td>
<td></td>
</tr>
<tr>
<td>• LastName</td>
<td></td>
</tr>
<tr>
<td>• BirthDate</td>
<td></td>
</tr>
<tr>
<td>• Address</td>
<td></td>
</tr>
<tr>
<td>• Contacts</td>
<td></td>
</tr>
</tbody>
</table>

### Paolo

<table>
<thead>
<tr>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>• FirstName = Paolo</td>
</tr>
<tr>
<td>• LastName = Fabbri</td>
</tr>
<tr>
<td>• BirthDate = ...</td>
</tr>
<tr>
<td>• Address = ...</td>
</tr>
<tr>
<td>• Contacts = ...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>• getFullName</td>
</tr>
<tr>
<td>• getAge</td>
</tr>
</tbody>
</table>

### Davide

<table>
<thead>
<tr>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>• FirstName = Davide</td>
</tr>
<tr>
<td>• LastName = Ferraro</td>
</tr>
<tr>
<td>• BirthDate = ...</td>
</tr>
<tr>
<td>• Address = ...</td>
</tr>
<tr>
<td>• Contacts = ...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>• getFullName</td>
</tr>
<tr>
<td>• getAge</td>
</tr>
</tbody>
</table>
Object-Oriented Programming Basics

**Properties**

encapsulate object data

**Methods**

implement the object behavior

**Events** and **Listeners**

implement objects communication

**Inheritance**

allows composition and reusability

---

**People**

- **Properties**
  - FirstName
  - LastName
  - BirthDate
  - Address
  - Contacts
- **Methods**
  - getFullName
  - getAge

**Employee**

- **Properties**
  - Salary
  - Group
  - Manager
- **Methods**
  - showInOrgChart
Object-Oriented Programming with MATLAB

**properties**
encapsulate object data

**methods**
implement the object behavior

**events** and **listeners**
implement objects communication

**inheritance**
allows composition and reusability

MATLAB EXPO 2014
Case Study: Team Resources Assignment
MATLAB Unit Test Framework
Application Development Process

Requirements

Research & Design
- Explore and discover
- Gain insight into problem
- Evaluate options, trade-offs

Design

Test

Elaborate

Implementation
- Desktop or Web Deployment
  - .dll
  - .exe
  - C, C++

- Embedded software and digital electronics
  - C, C++
  - VHDL, Verilog

Test & Verification
xUnit Patterns

Many testing Best Practices are emerging built around xUnit

- Consistent nomenclature
- Consistent architecture
- Platform for robust testing methodologies
- Setup/Teardown capabilities
MATLAB Unit Test Framework

Write, run and analyze tests for MATLAB programs

Write in function form or xUnit-style class form

Run tests individually or grouped into a test suite

Analyze values using qualification functions

Setup and teardown to pretest and restore a state

Advanced testing capabilities
# Types of Qualifications

<table>
<thead>
<tr>
<th>Type</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verify</td>
<td>Fail &amp; Continue Execution</td>
</tr>
<tr>
<td>Assert</td>
<td>Fail &amp; Halt Current Test, Continue to Next</td>
</tr>
<tr>
<td>Fatal assert</td>
<td>Fail &amp; Halt Framework Execution</td>
</tr>
<tr>
<td>Assume</td>
<td>Filter Current Test</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Assumption</th>
<th>Assertion</th>
<th>Fatal Assertion</th>
</tr>
</thead>
<tbody>
<tr>
<td>assumeTrue</td>
<td>assertTrue</td>
<td>fatalAssertTrue</td>
</tr>
<tr>
<td>assumeFalse</td>
<td>assertFalse</td>
<td>fatalAssertFalse</td>
</tr>
<tr>
<td>assumeEqual</td>
<td>assertEqual</td>
<td>fatalAssertEqual</td>
</tr>
<tr>
<td>assumeNotEqual</td>
<td>assertNotEqual</td>
<td>fatalAssertNotEqual</td>
</tr>
<tr>
<td>assumeSameHandle</td>
<td>assertSameHandle</td>
<td>fatalAssertSameHandle</td>
</tr>
<tr>
<td>assumeNotSameHandle</td>
<td>assertNotSameHandle</td>
<td>fatalAssertNotSameHandle</td>
</tr>
<tr>
<td>assumeReturnsTrue</td>
<td>assertReturnsTrue</td>
<td>fatalAssertReturnsTrue</td>
</tr>
</tbody>
</table>
Source Control Integration
Source Control Integration in MATLAB R2014b

Stay inside MATLAB for development workflow

**GIT** and **Subversion** Integration in Current Folder

Integrated with tools designed for MathWorks file types (i.e. compare and merge)
Programming Interfaces
Calling MATLAB from Other Languages

Engine interface
- C/C++
- Fortran
- Python®

Automation server
- COM
- .NET
Calling Other Languages from MATLAB

MATLAB C and Fortran API
>> mex mycode.c

MATLAB interface to C shared library
>> loadlibrary('foo', 'foo.h')

MATLAB COM client support
>> actxserver('Excel.Application')

MATLAB .NET interface
>> NET.addAssembly('System.Speech')

MATLAB Java interface
>> java.lang.String('boo')

MATLAB Python interface
>> py.textwrap.wrap('Text'); R2014b
Target-Specific Implementation and Large Scale Distribution
MATLAB Application Development Landscape

Prototyping

Programming

Deployment
MATLAB Application Development Landscape

Prototyping  Programming  Deployment
Thank You!