Go places with

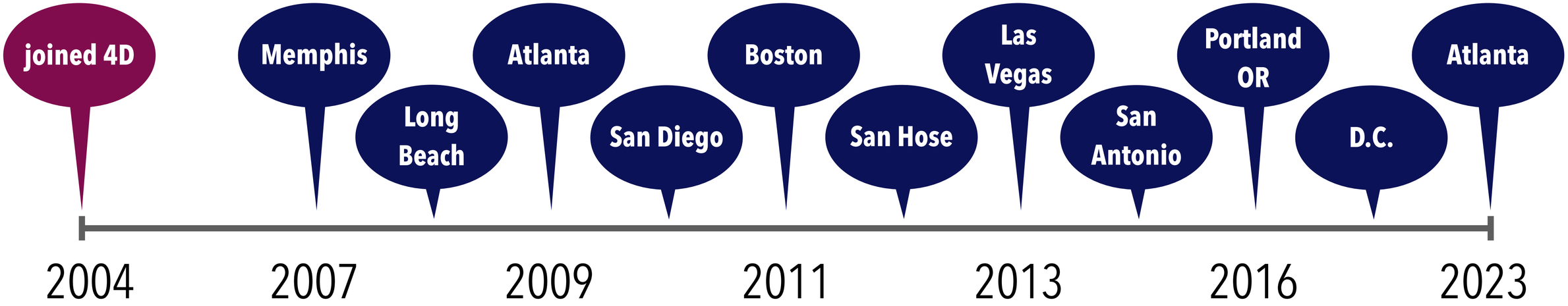
System Workers!

4D Summit 2023 presentation

Keisuke Miyako

Technical Account Manager

Introduction



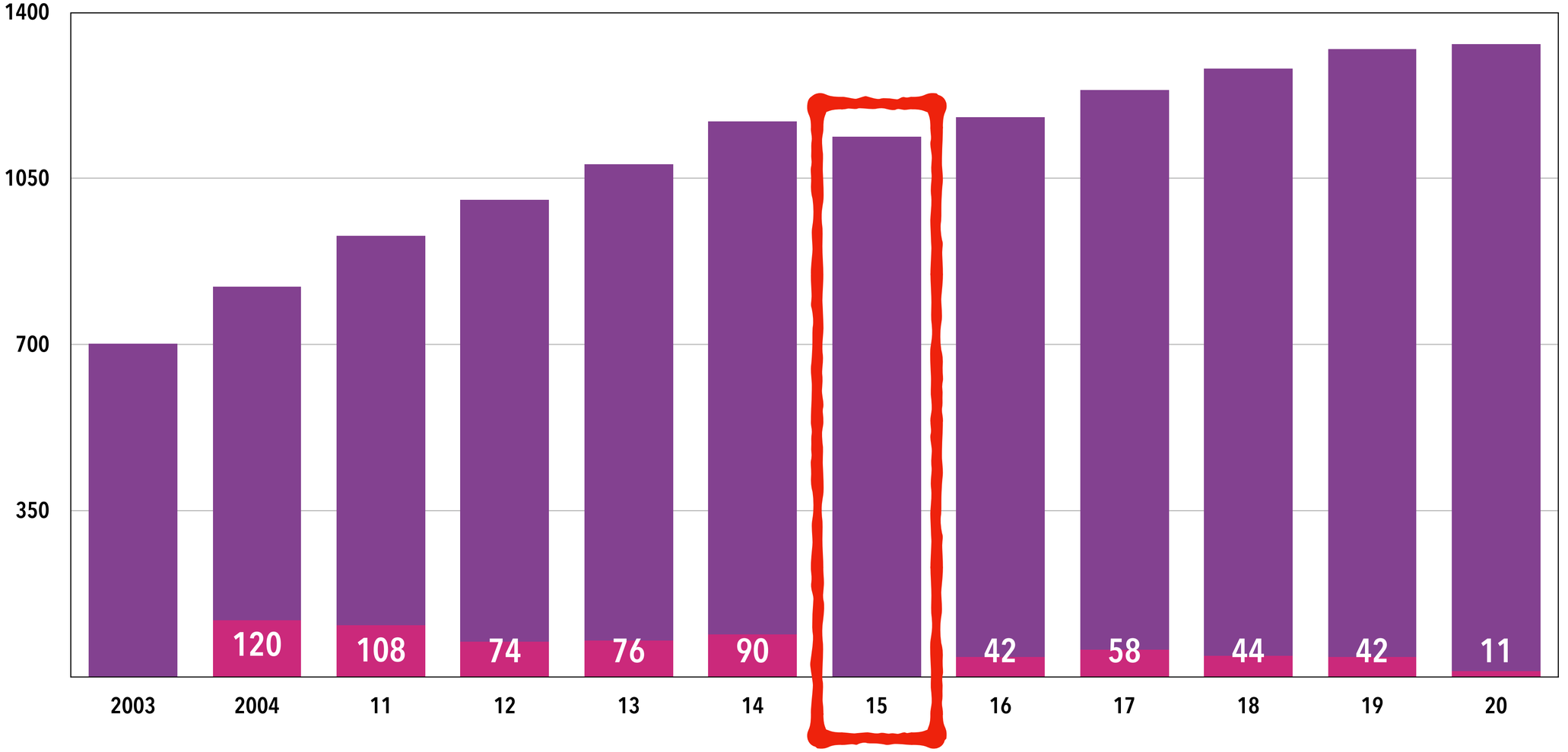
Introduction

first 4D version: 2004

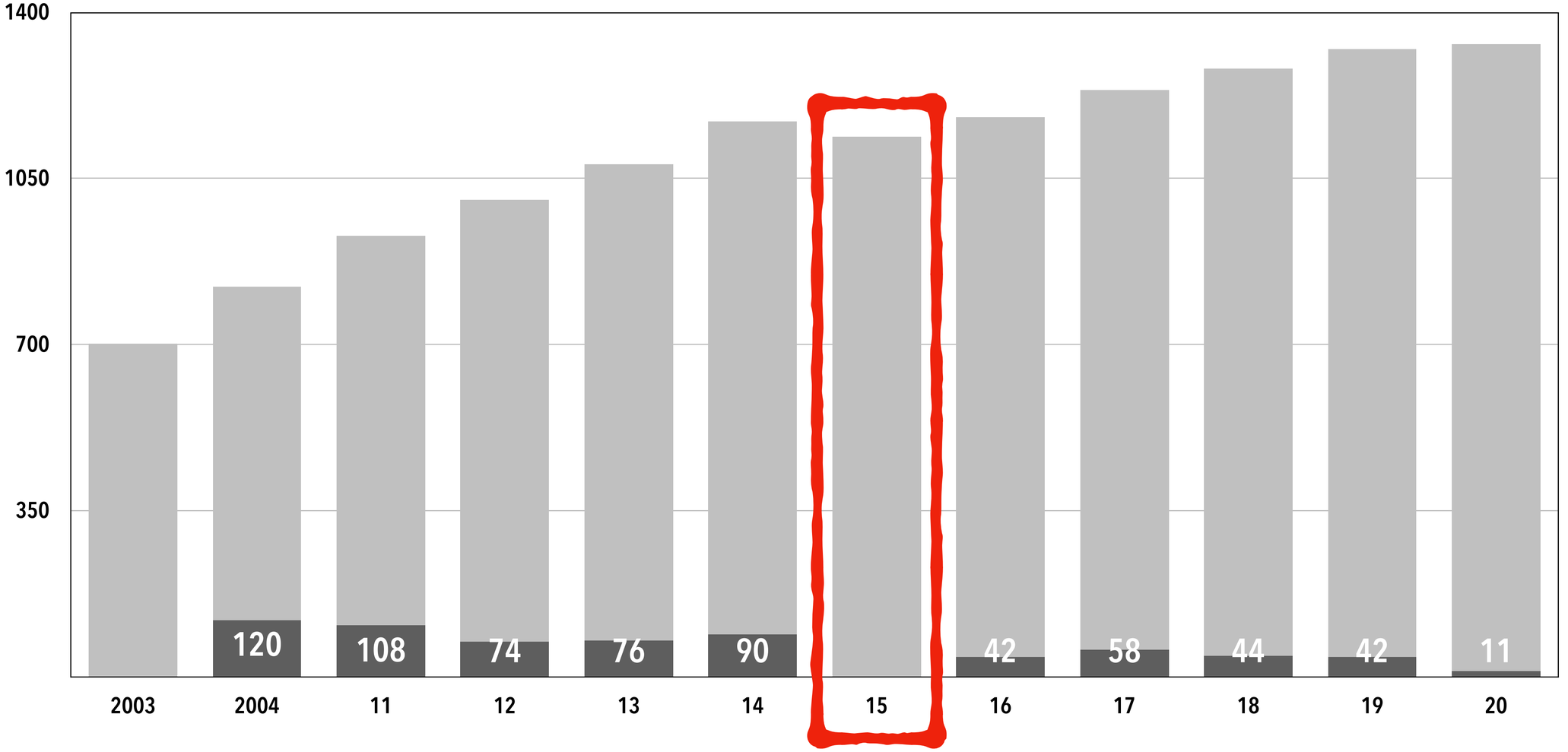
prior programming knowledge: BASIC

4D is easy to learn, fun to code!

4D commands by version



4D commands by version



paradigm shift

Object

paradigm shift

v14: C_OBJECT, ARRAY OBJECT, JSON

v15: **Object Fields**

v16: Commands to access object attributes

v17: Object Notation, ORDA

v18: Project Mode, formula object

v19: Class API for object definition

v20: Blob object, property declaration

Object

paradigm shift

v14: C_OBJECT, ARRAY OBJECT, JSON

v15: **Object Fields**

v16: Commands to access object attributes

v17: Object Notation, ORDA

v18: Project Mode, formula object

v19: Class API for object definition

v20: Blob object, property declaration

Procedural, Functional, Object Oriented

more commands the better!

sequence of imperative commands

state (current record, current selection, process variables) is shared implicitly

Procedural, **Functional**, Object Oriented

fewer process variables the better!

host

arguments

return value

component

dependency purely based on **arguments** and **return value**

state (current record, current selection, process variables) is not shared

Object

v14: C_OBJECT, ARRAY OBJECT, JSON

v15: **Object Fields**

v16: Commands to access object attributes

paradigm shift

v17: Object Notation, ORDA

v18: Project Mode, formula object

v19: Class API for object definition

v20: Blob object, property declaration

Object - Generic Data Structure

record

Student	
ID	2^{32}
firstName	A
lastName	A
englishLevel	2^{16}
finalExam	A
ranking	2^{16}
score	2^{32}

object

```
Student: {
        ID:1,
        firstName: William,
        lastName: Shakespeare,
        englishLevel: 1,
        finalExam: pass,
        ranking: 1,
        score: 1616
      }
```

alternative to...

- key/value pair
- associative arrays
- XML
- BLOB
- list properties

Object - Generic Data Structure

record **object** **alternative to...**

Student	
ID	2^{32}
firstName	A
lastName	A
englishLevel	2^{16}
finalExam	A
ranking	2^{16}
score	2^{32}

Formula creates a formula object based upon the *formulaExp* expression. *formulaExp* can be as simple as a single value or complex, such as a project method with parameters.

Formula **This** **game changer!**

You can specify the object on which the formula is executed. The properties of the object can then be accessed via the **This** command.

Object - Generic Data Structure with Code

record

object

```
Student: {
        ID:1,
        firstName: William,
        lastName: Shakespeare,
        englishLevel: 1,
        finalExam: pass,
        eligibilityForTransfer(school),
        ranking: 1,
        score: 1616
      }
```

alternative to...

- key/value pair
- associative arrays
- XML
- BLOB
- list properties

Procedural, Functional, **Object Oriented**

abstraction

This.capitalise(**This**.trim(**This**.firstName))

Inheritance *kind of...*

Person
.firstName:="William"
.lastName:="Shakespeare"
Student is based on **Person**
.englishLevel:=1
.finalExam:="pass"

encapsulation *kind of...*

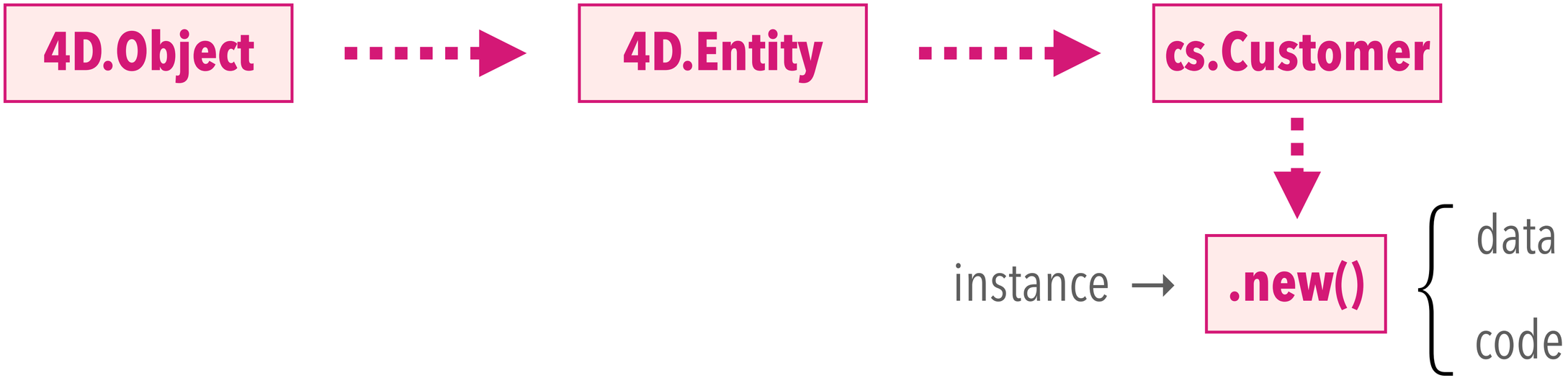
Student
._ranking:=1
._score:=1616

polymorphism

Student
This.register:=**Formula**(**registerStudent**)
Teacher
This.register:=**Formula**(**registerTeacher**)

Procedural, Functional, **Object Oriented**

fewer commands, more classes!



an object contains **data** and **code** (properties and functions)

dynamic state is represented within an **instance** of an object

Procedural, Functional, Object Oriented

4D embraces all 3 paradigms!

procedural	functional	object oriented
user interface	generic code library	asynchronous
global scope	sharable module	event-driven

Procedural, Functional, Object Oriented

4D embraces all 3 paradigms!

procedural

user interface

global scope

11 new commands since v19

1. Create deployment license
2. SET HELP MENU
3. Last errors
4. OBJECT SET SUBFORM CONTAINER VALUE
5. OBJECT Get subform container value
6. Copy parameters
7. Get license usage
8. WP DELETE TEXT BOX
9. WP New text box
10. WP Get data context
11. WP SET DATA CONTEXT

Procedural, Functional, Object Oriented

4D embraces all 3 paradigms!

functional

generic code library

sharable module

4D Progress

4D SVG

4D ViewPro

4D Widgets

4D Go Mobile

Procedural, Functional, Object Oriented

4D embraces all 3 paradigms!

object oriented

asynchronous

event-driven

new built-in classes

FileHandle
HTTPRequest
SystemWorker
WebSocketServer/Connection

classic implementation

Open document, SEND PACKET, etc.
HTTP Get, HTTP Request
LAUNCH EXTERNAL PROCESS
WEB SEND RAW DATA

Go places with

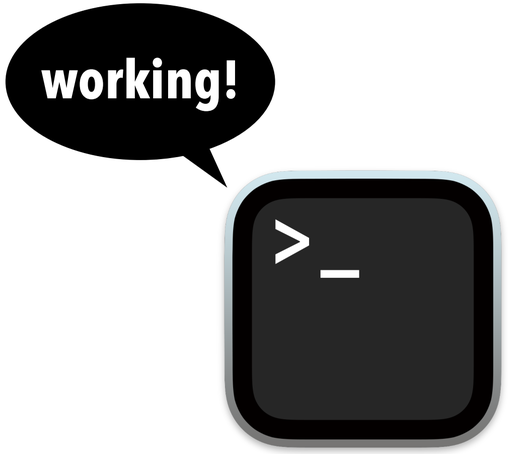
System Workers!

4D Summit 2023 presentation

System Worker

- **object-oriented** implementation of **LAUNCH EXTERNAL PROCESS**
  - **efficient**: communication takes place in the **background**
  - **asynchronous**: process **streams of data**
  - **interactive**: respond to **command line prompts** and make **consecutive calls**

LAUNCH EXTERNAL PROCESS



4D process

external process

Why does it matter?

Progress Indicator - Command Line Interface

standard error stream

```
0% U archive.7z \b\b\b\b\b\b\b\b\b\b\b
```

ASCII backspace character

Progress Indicator - LAUNCH EXTERNAL PROCESS

standard error stream

```
0% U archive.7z \b\b\b\b\b\b\b\b\b\b\b
1% U archive.7z \b\b\b\b\b\b\b\b\b\b\b
2% U archive.7z \b\b\b\b\b\b\b\b\b\b\b
3% U archive.7z \b\b\b\b\b\b\b\b\b\b\b
4% U archive.7z \b\b\b\b\b\b\b\b\b\b\b
5% U archive.7z \b\b\b\b\b\b\b\b\b\b\b
```

System Worker is asynchronous!

- designed for time-consuming operations
  - **display progress information** in real time
  - **process large data** in smaller chunks
  - **abort command** if necessary

Execution Context

Execution Context

New process

```
$worker:=4D.SystemWorker.new()
```

abort

background thread

execution context does not exist anymore!

Execution Context

CALL WORKER

```
$worker:=4D.SystemWorker.new()
```

background thread

Execution Context

DIALOG

```
$worker:=4D.SystemWorker.new()
```

background thread

Demonstration

SystemWorker needs execution context

Recap

Recap

- external process is executed in **background thread**
- background thread invokes **SystemWorker instance**
- SystemWorker instance needs **execution context** (WORKER or DIALOG)
  - implement **callback functions** for asynchronous execution
  - implement **buffer** to process partial responses
  - implement **task queue** to consecutive execution

Inheritance

Inheritance

object oriented programming

generic → **specific**

Base class to implement
any command line interface

Subclass of **_CLI** to execute
the **7z** command line interface

Inheritance

object oriented programming

generic → **specific**

_CLI

properties

controller
currentDirectory
EOL
executableFile
executableName
executablePath
name
platform

methods

.escape()
.quote()

SevenZip extends _CLI

SevenZip

methods

implement the **7z** command line interface

- .terminate()
- .add()
- .extract()

```
Usage: 7z <command> [<switches>...] <archive_name>
[<file_names>...] [@listfile]
```

Inheritance

object oriented programming

generic → **specific**

_CLI

properties

controller
currentDirectory
EOL
executableFile
executableName
executablePath
name
platform

methods

.escape()
.quote()

cURL extends _CLI

cURL

methods

.terminate()
.get()
.post()

implement the **curl** command line interface

```
Usage: curl [options...] <url>
```

Inheritance

object oriented programming

generic	specific
Classes	
	_CLI.4dm
	_CLI_Controller.4dm
	_Worker_Controller.4dm
	_Form_Controller.4dm

Inheritance

object oriented programming

generic	specific
Classes	_CLI.4dm
	SevenZip.4dm
	_CLI_Controller.4dm
	_Worker_Controller.4dm
	_Form_Controller.4dm
	_SevenZip_Form_Controller.4dm

Inheritance

object oriented programming

generic	specific
Classes	_CLI.4dm
	cURL.4dm
	_CLI_Controller.4dm
	_Worker_Controller.4dm
	_Form_Controller.4dm
	_cURL_Form_Controller.4dm

Demonstration

SystemWorker and class inheritance

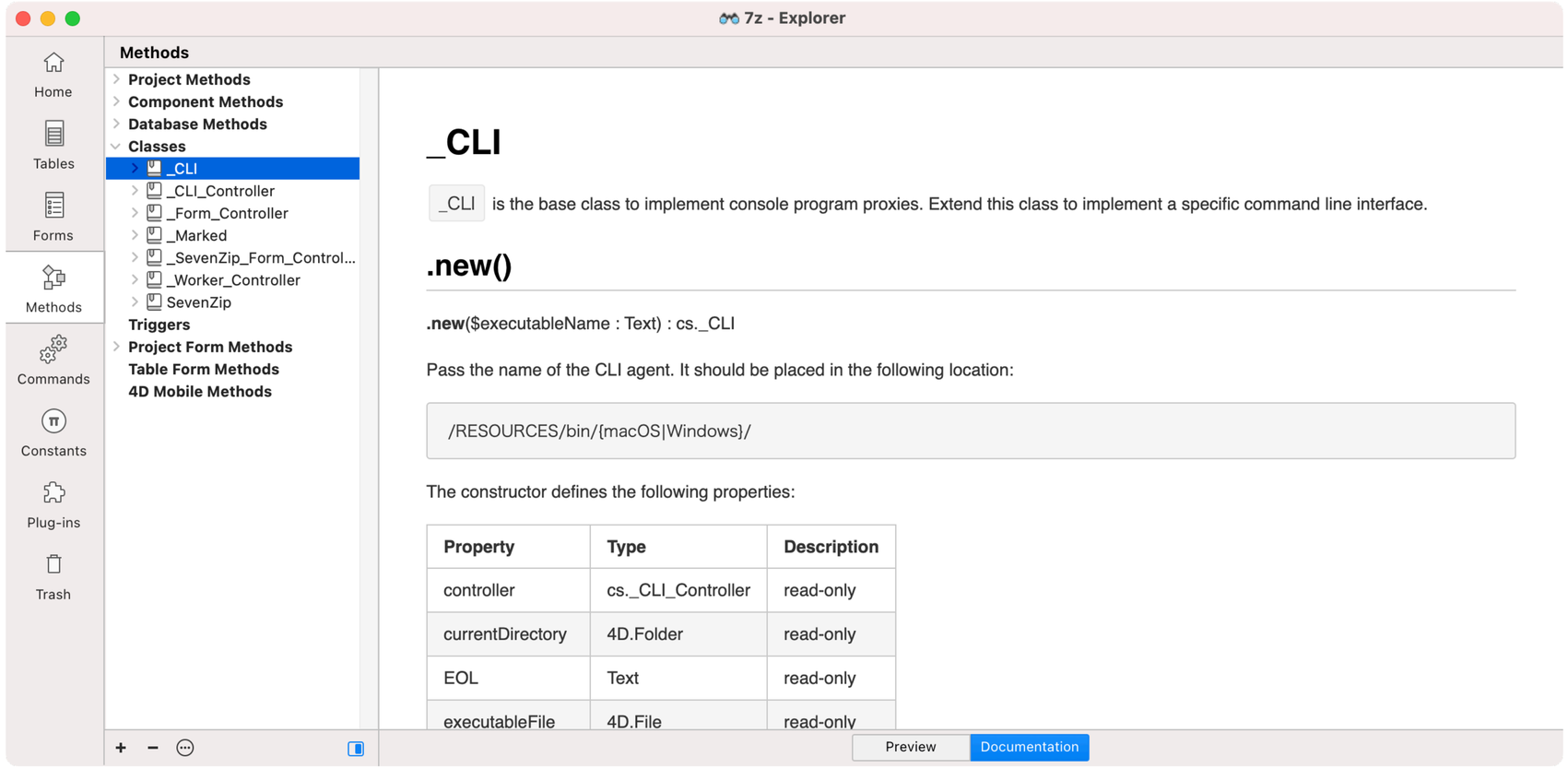
Recap

Recap

- **platform agnostic**
  - convert File/Folder object to and from platform path
  - take care of folder separator, end-of-line, UNIX execute bit, file extension
- flexible API
  - take advantage of **Variant, Copy parameters**
- class documentation

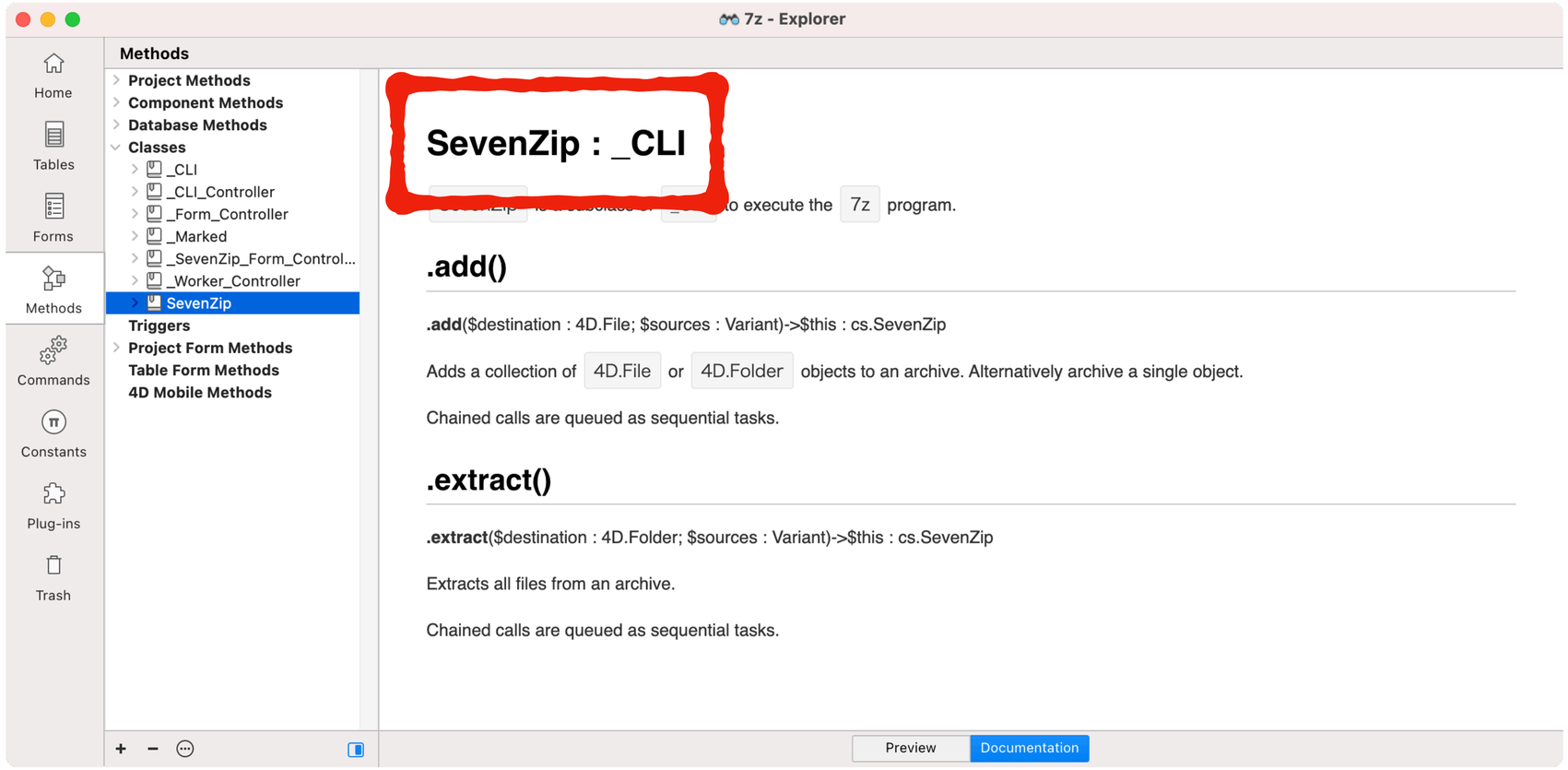
Inheritance

Explain the class hierarchy to other developers (and to your future self)



Inheritance

Explain the class hierarchy to other developers (and to your future self)



Abstraction

Abstraction

```
$worker:=4D.SystemWorker.new($command;$options)
```

what it this?

Abstraction

```
$worker:=4D.SystemWorker.new($command;$options)
```

controller

.*onResponse*
.*onData*
.*onDataError*
.*onError*
.*onTerminate*

.timeout
.dataType
.encoding
.variables
.currentDirectory
.hideWindow

Abstraction

```
$worker:=4D.SystemWorker.new($command;$options)
```

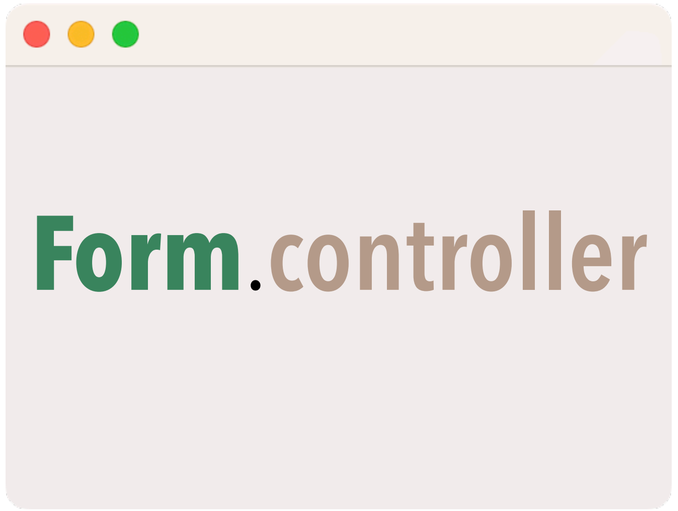
controller

```
Class Constructor

Function onResponse(4D.SystemWorker; Object)
Function onData (4D.SystemWorker; Object)
Function onDataError (4D.SystemWorker; Object)
Function onError (4D.SystemWorker; Object)
Function onTerminate (4D.SystemWorker; Object)
```

Abstraction

```
$worker:=4D.SystemWorker.new($command;$controller)
```

$controller

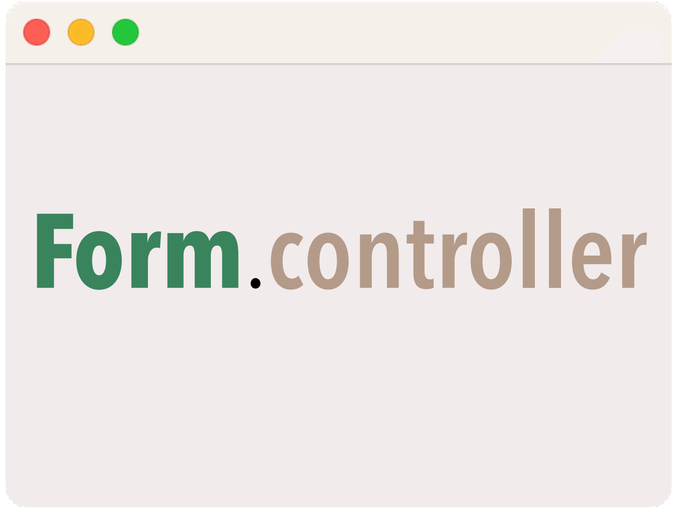


WORKER

DIALOG

Abstraction

```
$worker:=4D.SystemWorker.new($command;$controller)
```



WORKER

DIALOG is like a WORKER with UI layer

Abstraction

```
$worker:=4D.SystemWorker.new($command;$controller)
```

$controller

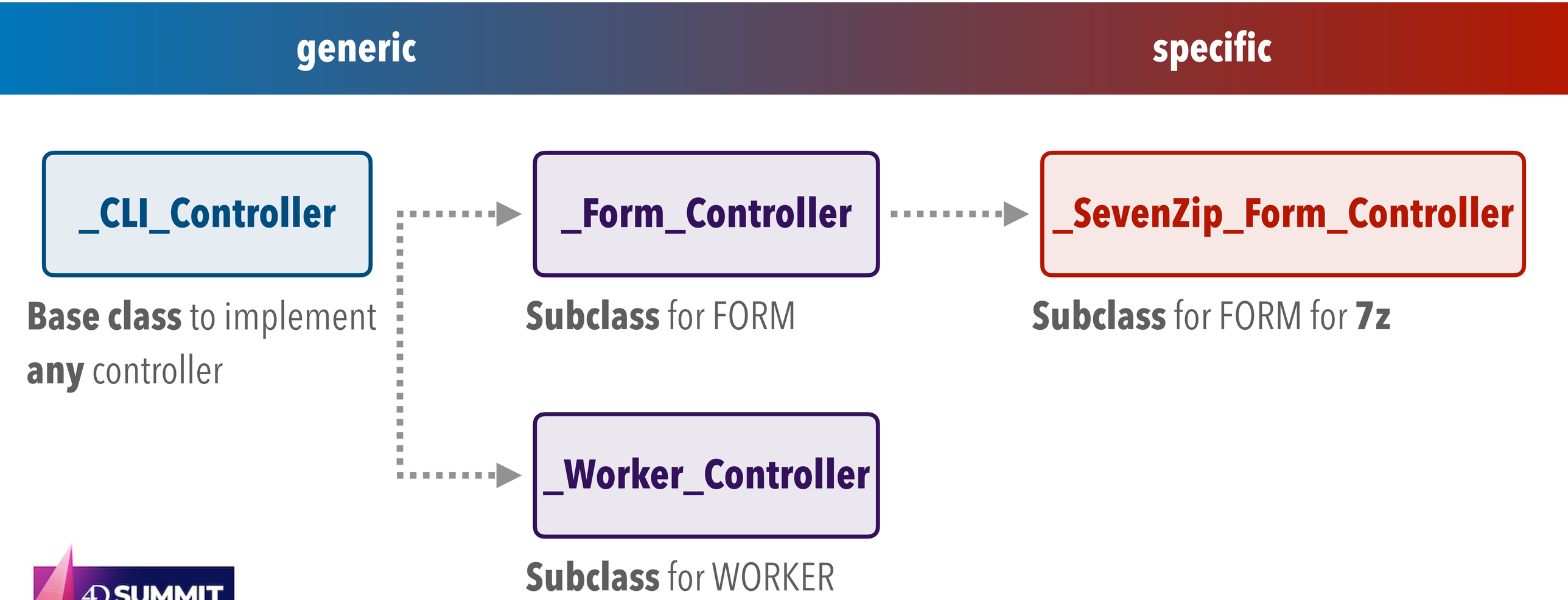
WORKER

$controller + Form

DIALOG is like a WORKER with UI layer

Abstraction

object oriented programming



Demonstration

DIALOG is like a WORKER with UI layer

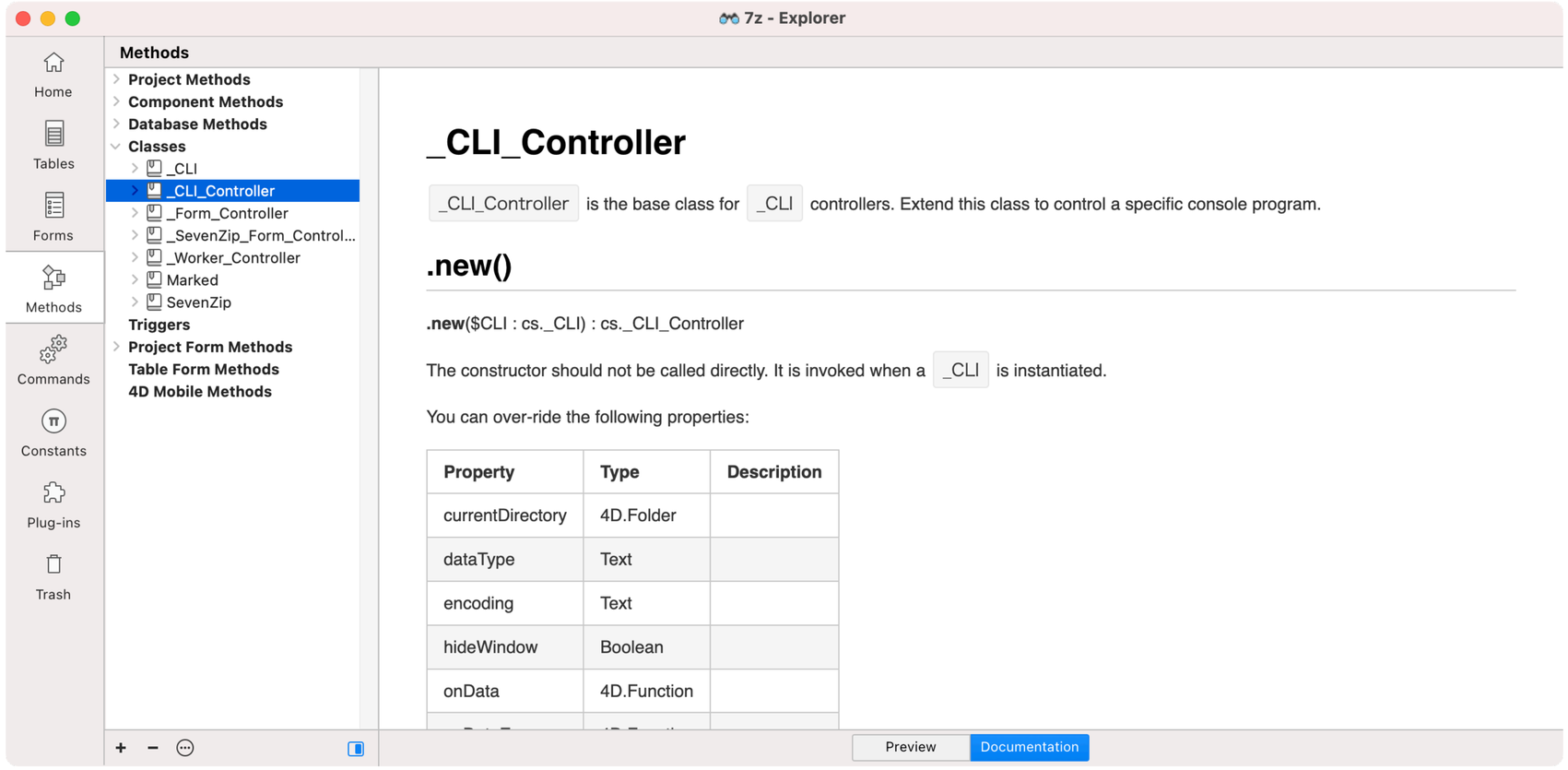
Recap

Recap

- **abstract super-class** for both WORKER and DIALOG
- **common sub-class** suitable for any form
- **specialised sub-sub-class** for specific forms

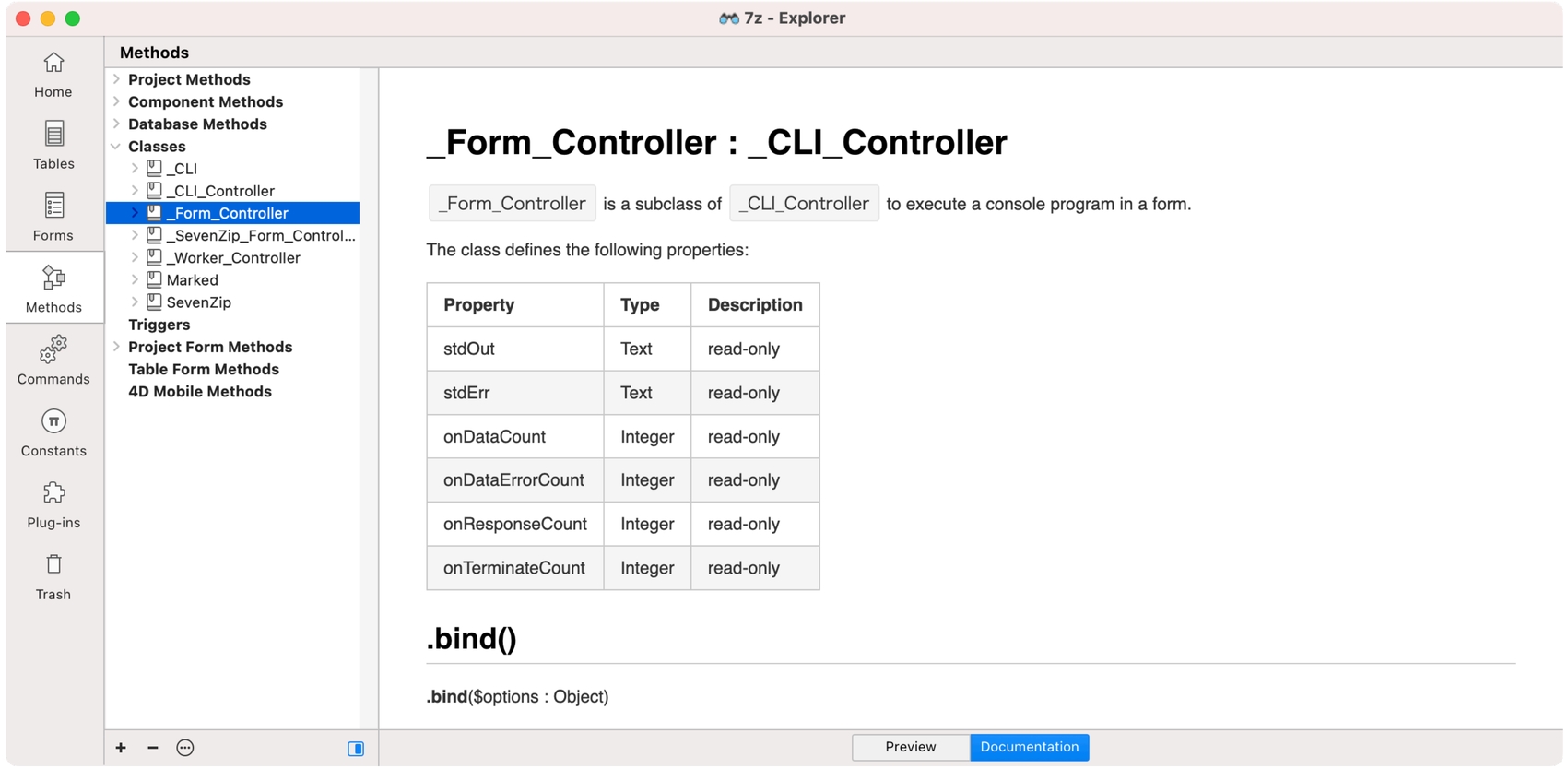
Abstraction

Explain the class hierarchy to other developers (and your future self)



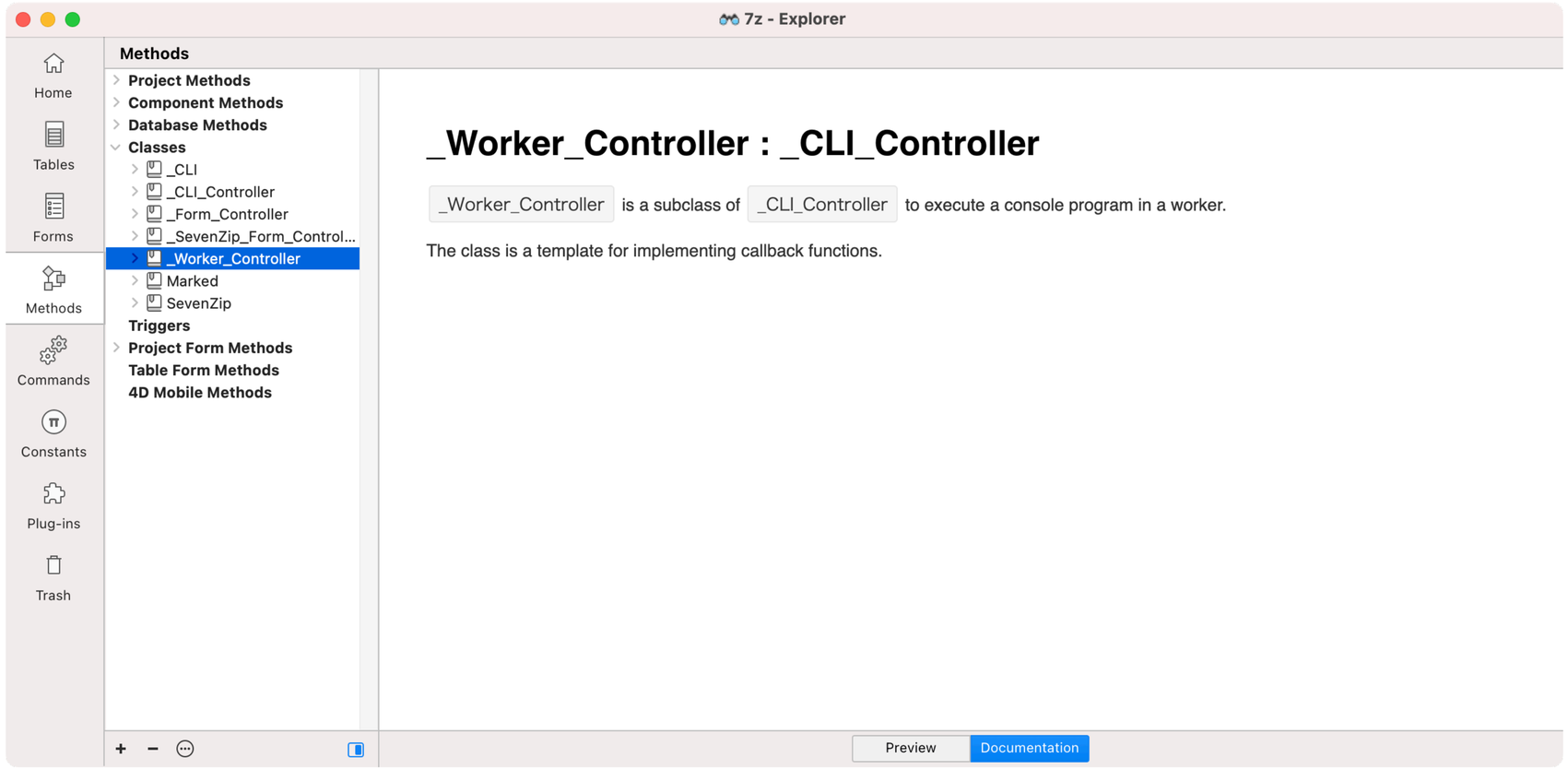
Abstraction

Explain the class hierarchy to other developers (and your future self)



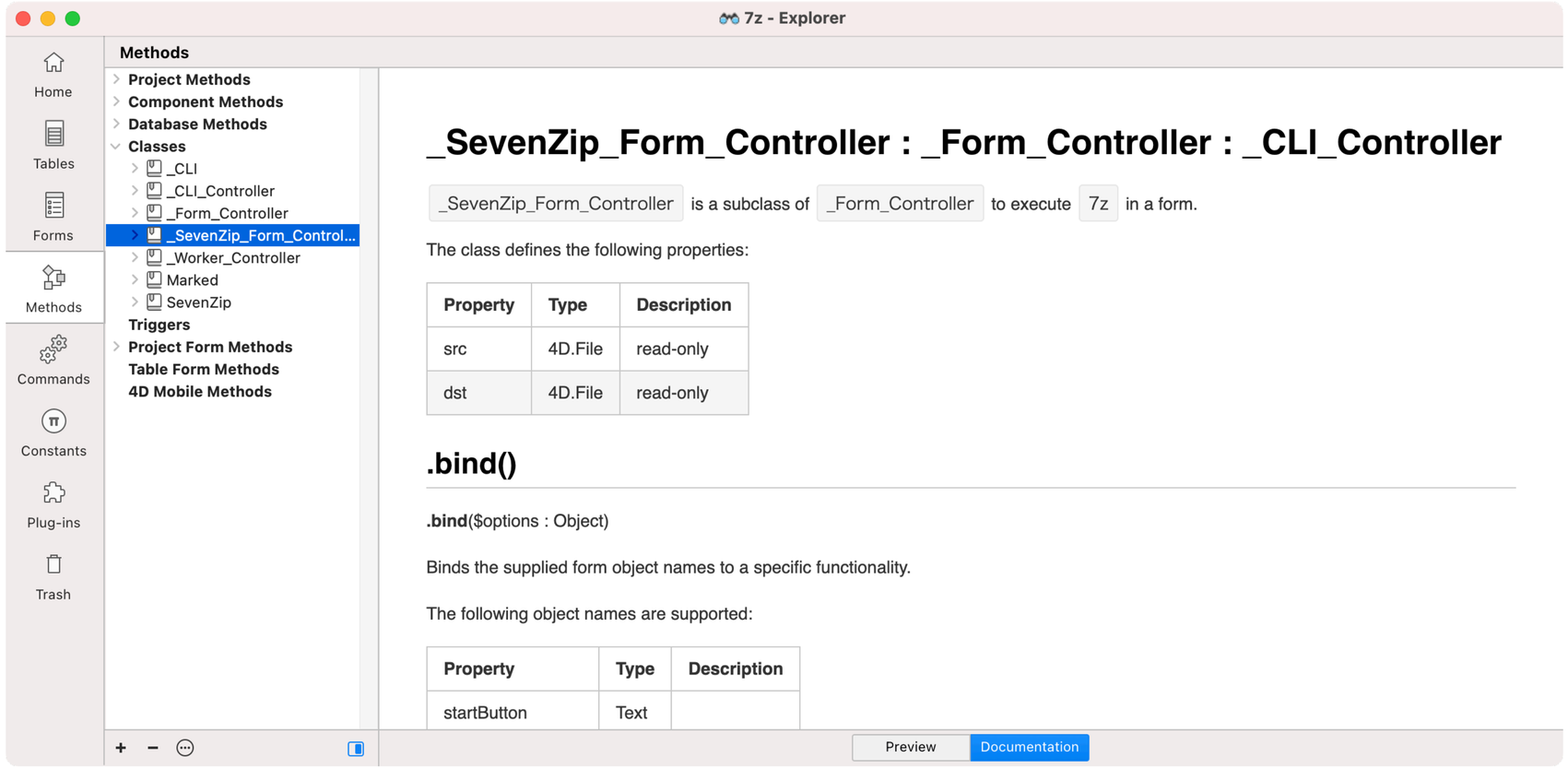
Abstraction

Explain the class hierarchy to other developers (and your future self)



Abstraction

Explain the class hierarchy to other developers (and your future self)



Encapsulation

Encapsulation

```
Class constructor($executableName : Text; $controller : 4D.Class)

  This._name:=OB Class(This).name

  Case of
    : (Is macOS)
      This._platform:="macOS"
      This._executableName:=$executableName
      This._EOL:="\n"
    : (Is Windows)
      This._platform:="Windows"
      This._executableName:=$executableName+".exe"
      This._EOL:="\r\n"
  End case
```

Encapsulation

```
Class constructor($executableName : Text; $controller : 4D.Class)

  This._name



     This._platform
     This._executableName
     This._EOL

     This._platform
     This._executableName
     This._EOL
```

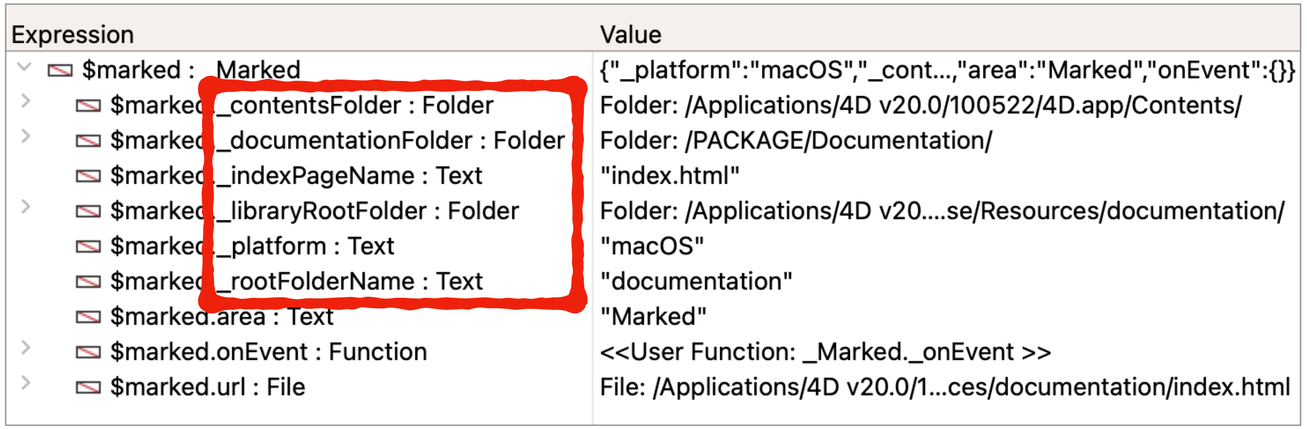
Encapsulation

```
Class constructor($executableName : Text; $controller : 4D.Class)

  This._name:=OB Class(This).name

Function get name()->$name : Text

  $name:=This._name
```

get name
read-only property

Encapsulation

- underscore-prefixed members are **excluded** from **autocompletion**
  - the purpose is to **filter** private or internal properties and methods in **development**
  - available in native method editor and LSP compatible external editor
- hiding doesn't protect data or restrict access
  - values are accessible from **debugger**



Encapsulation

- constructor
- private
- createTempFolder
- getDocumentationFile
- getFormFile
- getIndexPage
- onEvent
- splitPath
- public
- generate
- parse

```
$marked.createTempFolder
        getDocumentationFile
        getFormFile
        getIndexPage
        onEvent
        splitPath
        generate
        parse
        area
        result
        url
```

Encapsulation

- constructor
- private
- _createTempFolder
- _getDocumentationFile
- _getFormFile
- _getIndexPage
- _onEvent
- _splitPath
- public
- generate
- parse

```
$marked.generate
        parse
        area
        result
        url
```

Encapsulation

object oriented programming

_CLI

base abstract class

cannot be used directly

_Worker_Controller

extended generic class

can be used with any CLI

SevenZip

public class

constructor instantiates **default controller**

_CLI_Controller

base generic class

can be used directly (**default controller**)

_Form_Controller

extended generic class

can be used with any CLI

_SevenZip_Form_Controller

extended specific class

designed for any form

Encapsulation

object oriented programming

public class is explicitly used

hidden class is implicitly used

_CLI

base abstract class

cannot be used directly

_Worker_Controller

ric class

any CLI

SevenZip

public class

constructor instantiates **default controller**

_CLI_Controller

base generic class

can be used directly (**default controller**)

_Form_Controller

extended generic class

can be used with any CLI

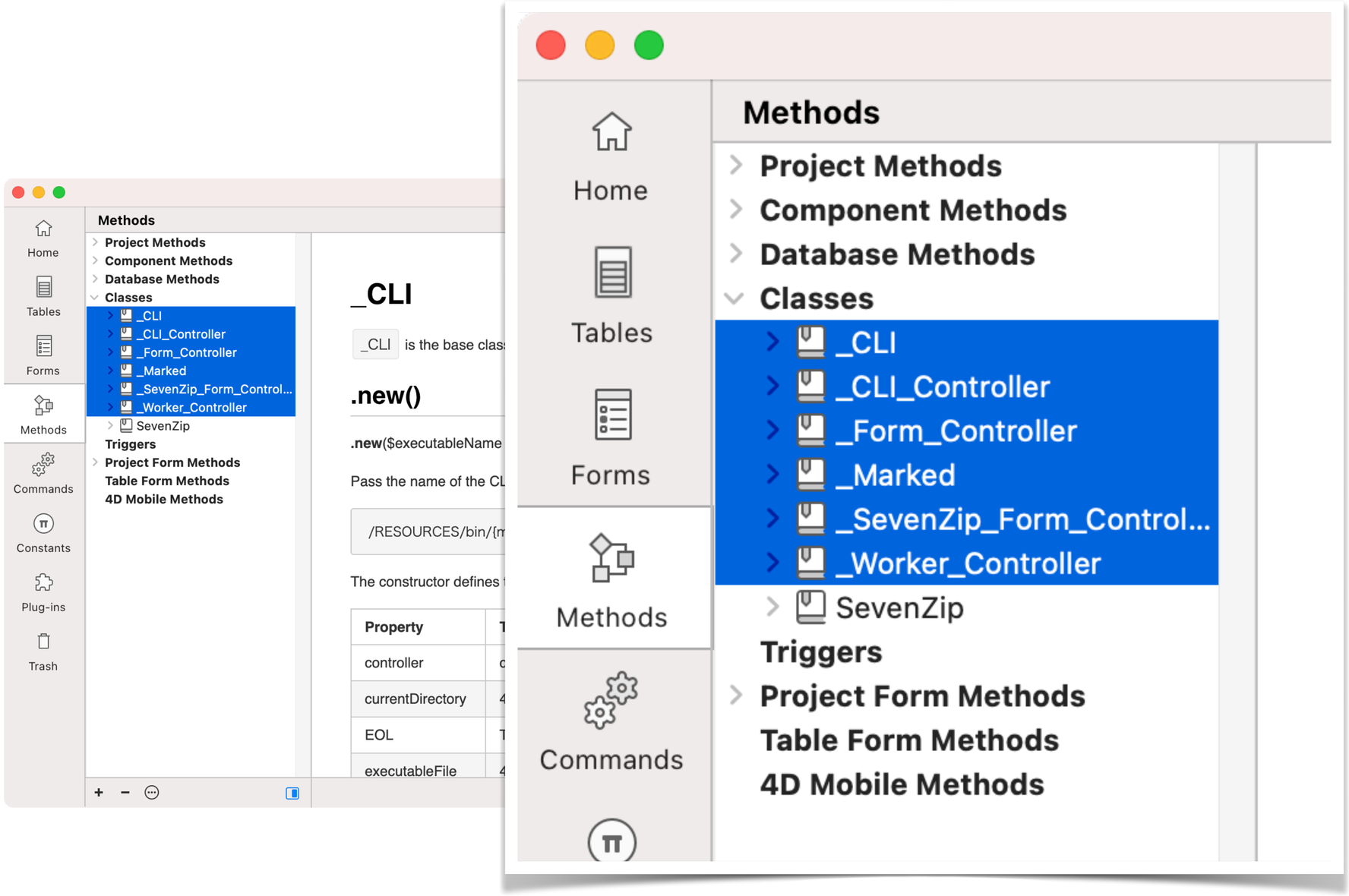
_SevenZip_Form_Controller

extended specific class

designed for any form

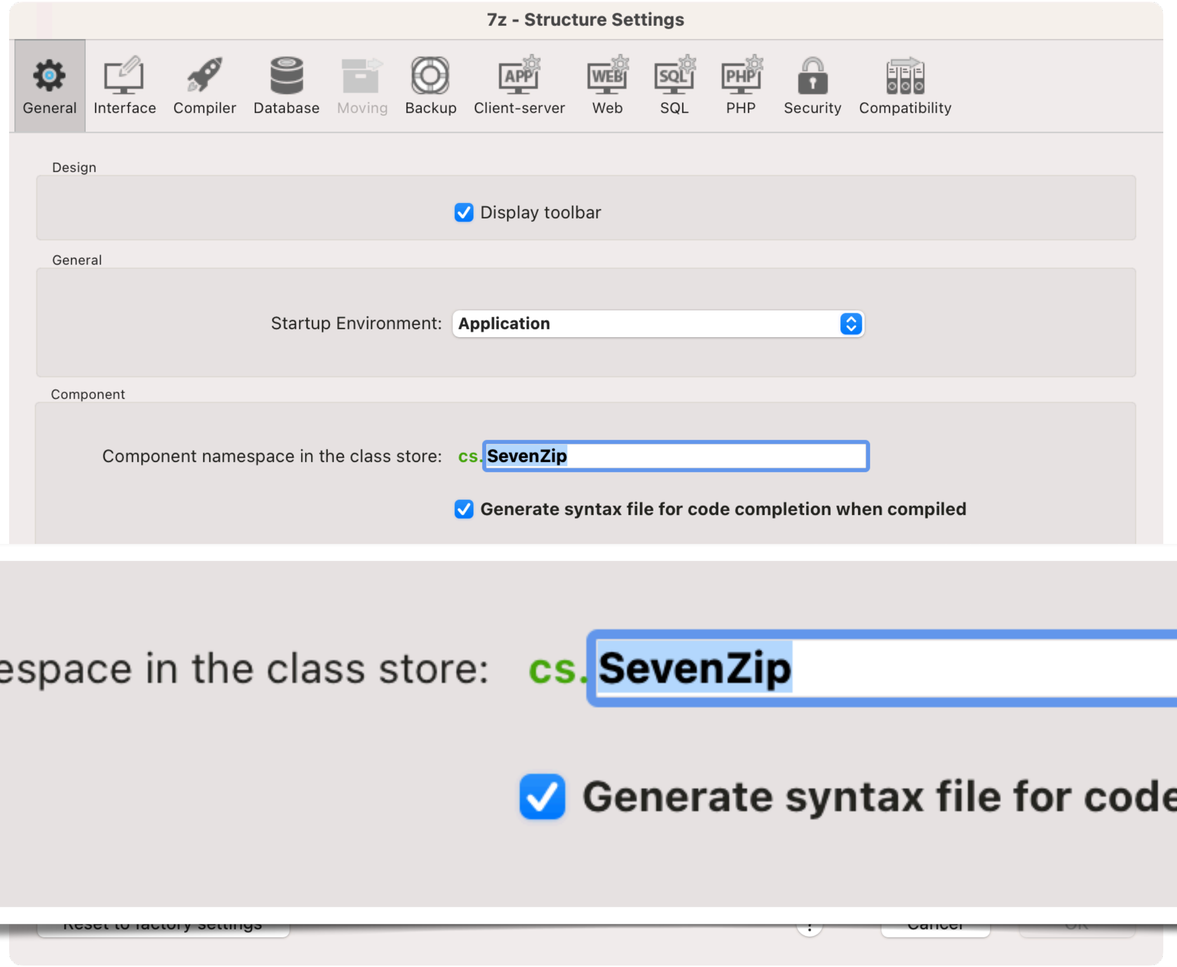
Encapsulation

private classes



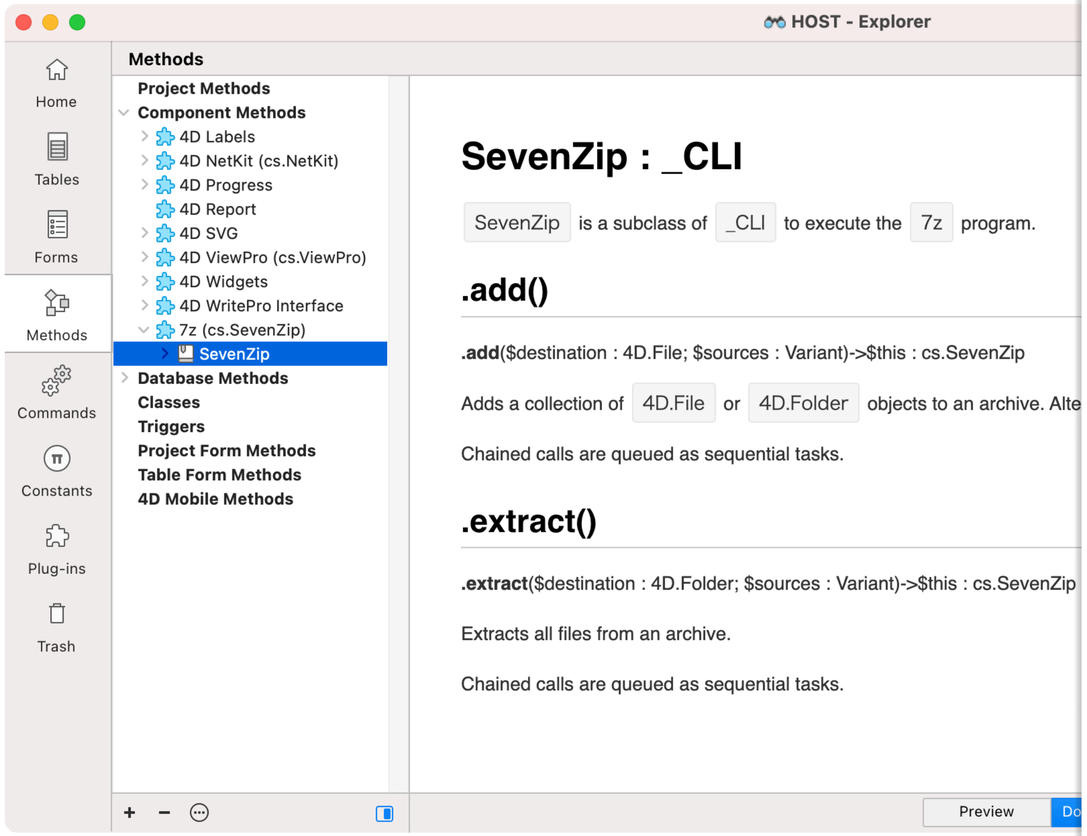
Encapsulation

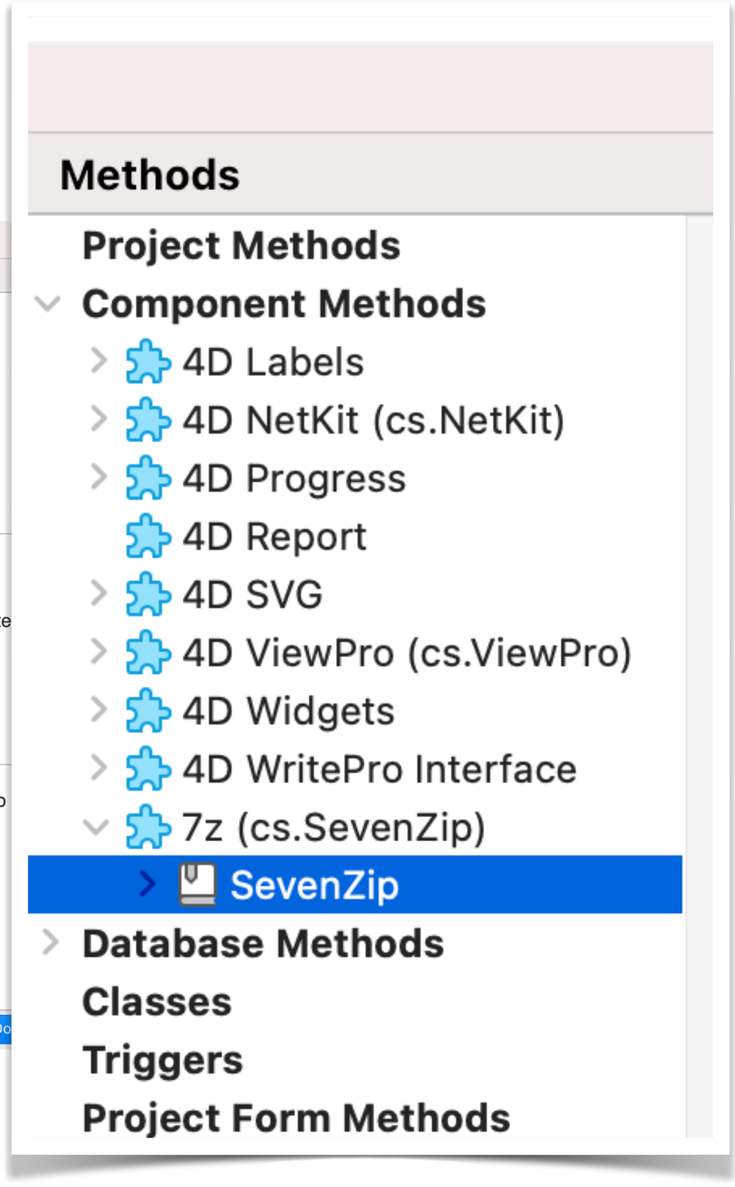
component namespace



Encapsulation

public class





Demonstration

hidden class, property, method + Marked.js

Recap

Recap

- **access control**: 4D does **not** have private, public, protected
  - underscore prefix **hides** information
- **read-only**: a property can have a **get function** only
- **public**: share component classes with **namespace** (also available in binary mode)

Polymorphism

Inheritance

object oriented programming

generic → specific

_CLI_Controller

.execute

push CLI to queue

_Form_Controller

.execute

push CLI to queue

_Worker_Controller

.execute

push CLI to queue

_SevenZip_Form_Controller

.execute

push CLI to queue

methods and properties are

inherited from super-class

Inheritance

object oriented programming

generic → specific

_CLI_Controller

.*complete* *(r/o)*
True if queue is empty

_Form_Controller

.*complete* *(r/o)*
True if queue is empty

_Worker_Controller

.*complete* *(r/o)*
True if queue is empty

_SevenZip_Form_Controller

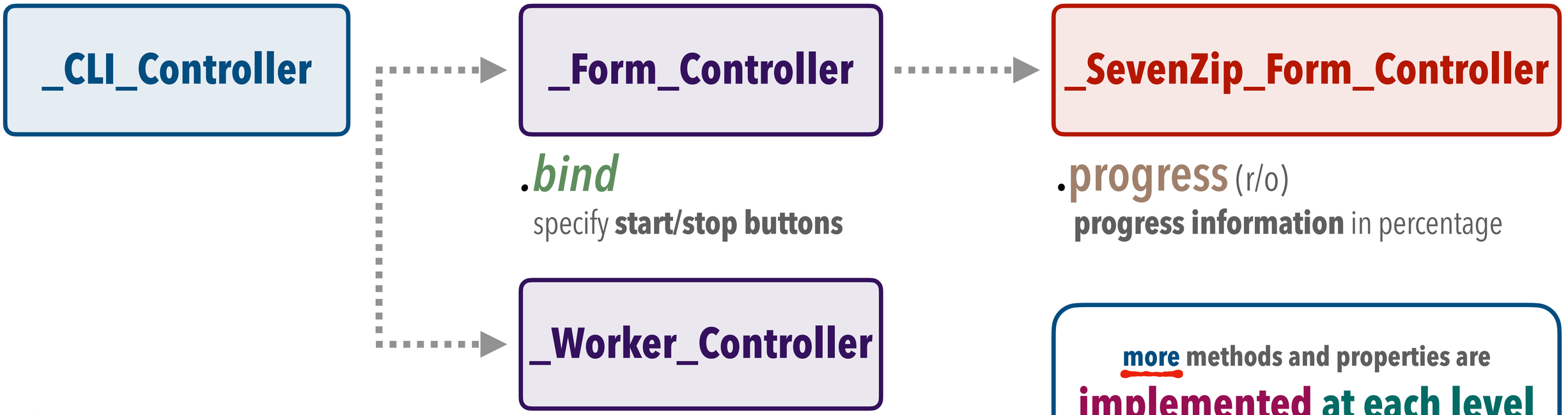
.*complete* *(r/o)*
True if queue is empty

methods and properties are
inherited from super-class

Inheritance

object oriented programming

generic → **specific**



Polymorphism

object oriented programming

generic → **specific**

_CLI_Controller

.*onDataError*

redirect to onEvent

_Form_Controller

.*onDataError*

capture **stdErr** in buffer

_Worker_Controller

.*onDataError*

placeholder function

_SevenZip_Form_Controller

.*onDataError*

parse **progress information** from stdErr

same methods and properties are

implemented at each level

Demonstration

System Worker and polymorphism

Recap

Recap

- **callback**: different needs for WORKER vs DIALOG
  - DIALOG has **form objects**; WORKER does not
  - DIALOG must check for the **Form** object
- **progress**: real-time update for DIALOG; log recording for WORKER
- **interruption**: stop button for DIALOG; timeout option for WORKER

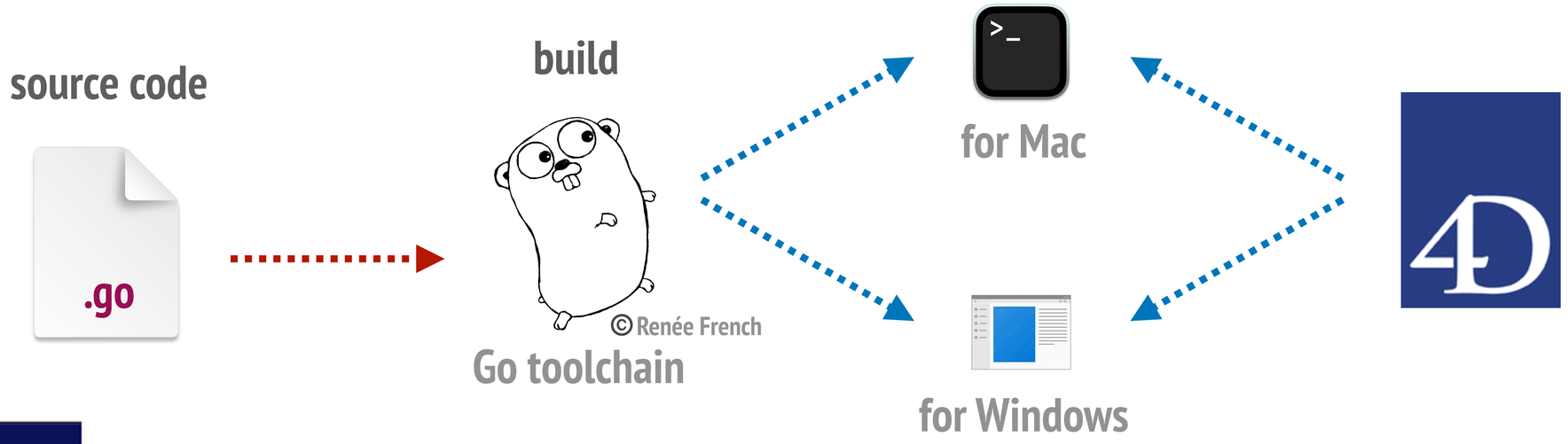
Extra

Go Primer

The Go Programming Language

https://golang.org/

- **modern**: statically typed, compiled high-level language
- **cross-platform**: supports Windows and macOS
- **portable**: generates statically linked binary executable



The Go Programming Language

https://golang.org/

- **popular**: active developer community
  - tutorials, libraries, examples
- **performance**: high-speed execution, efficient memory usage
- **formulaic**: consistent language design, ideal for AI assisted code

write me code in go that converts json to apache paquet

To convert JSON data to Apache Parquet format in Go, you can use the parquet-go library. Unfortunately, as of my knowledge cutoff in September 2021, there is no official Go implementation of Apache Parquet. However, there are third-party libraries like parquet-go available for this purpose.
To get started, you'll need to...

Install Go - Windows

https://go.dev/doc/install

installer option

Download (1.20.7)

package manager option

Install Go - macOS

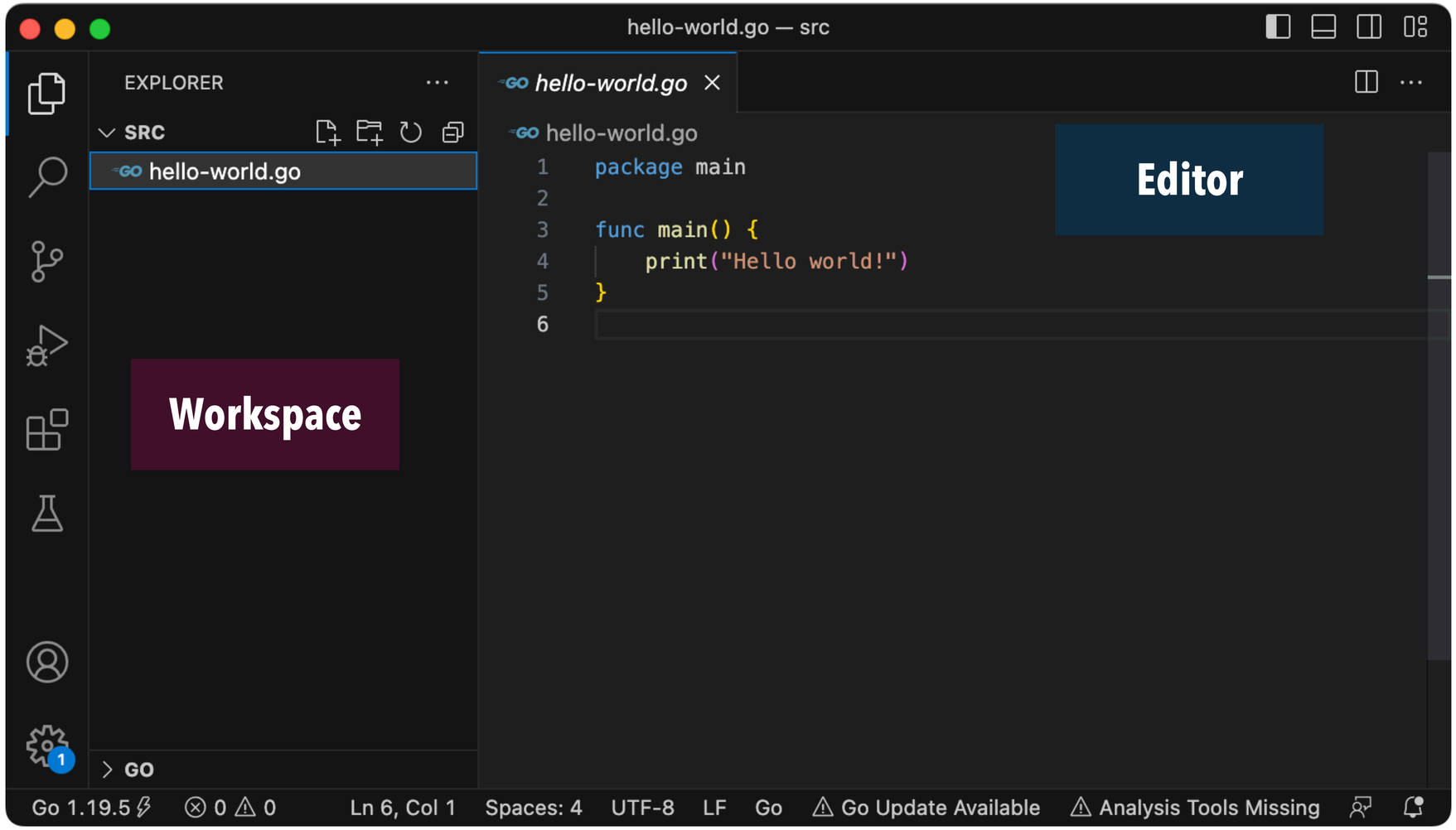
https://go.dev/doc/install

installer option

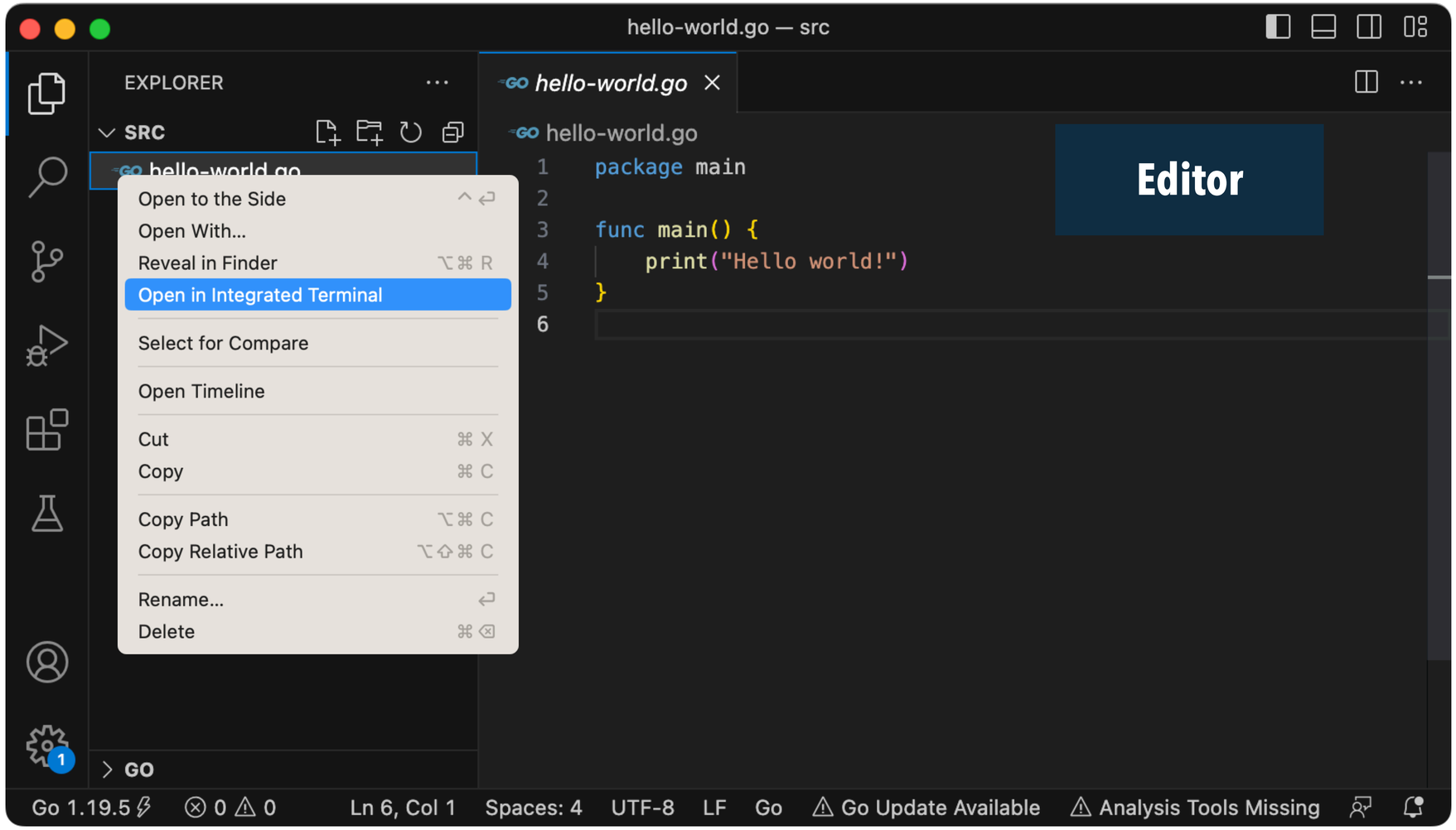
Download (1.20.7)

package manager option

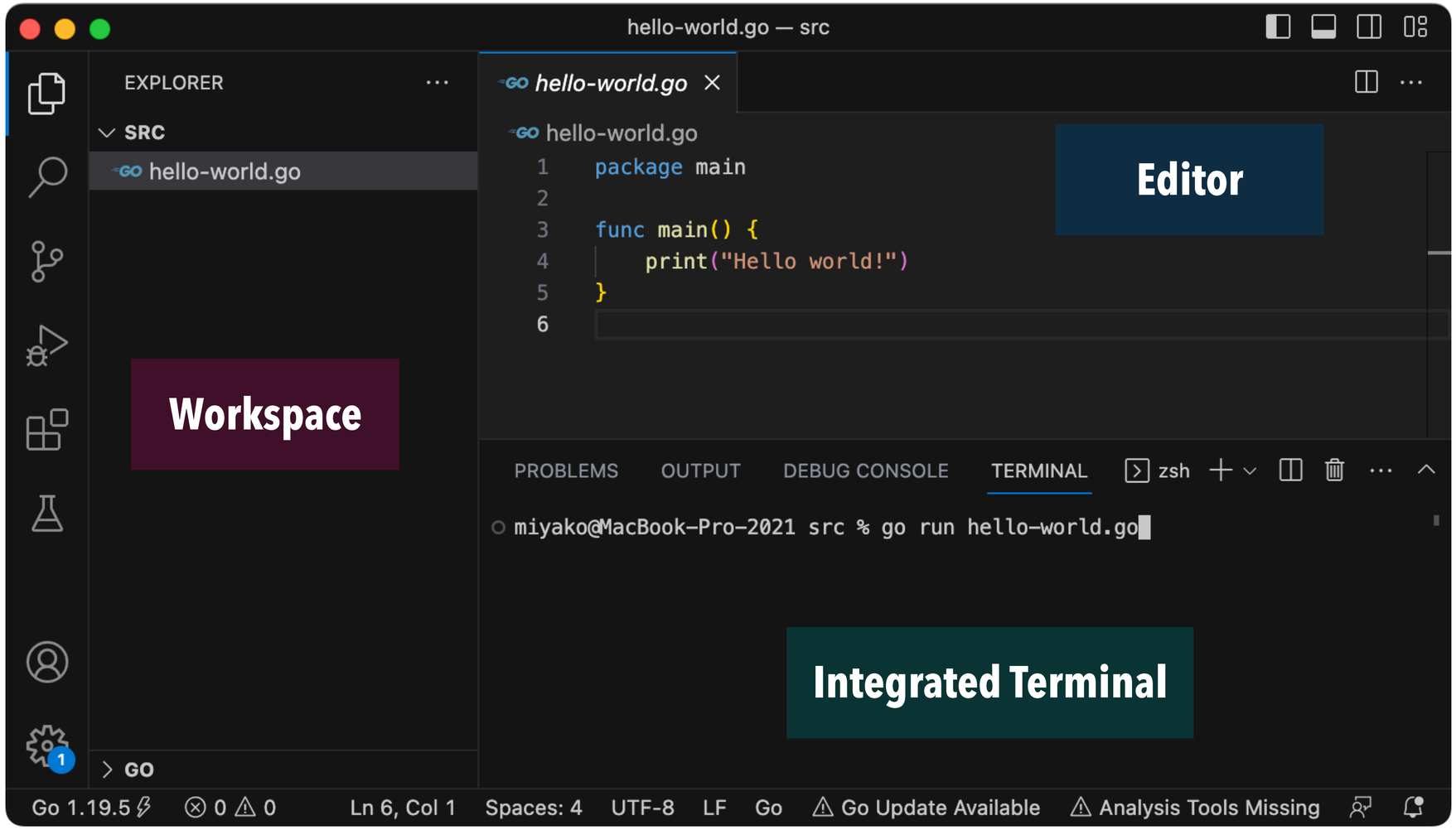
Go for Visual Studio Code



Go for Visual Studio Code



Go for Visual Studio Code



Go for Visual Studio Code

Build Go - Windows

```
go build hello-world.go
```

Build Go - macOS

```
GOOS=darwin GOARCH=amd64 go build -o
hello-world_amd64 hello-world.go

GOOS=darwin GOARCH=arm64 go build -o
hello-world_arm64 hello-world.go

lipo -create -output hello-world
hello-world_amd64 hello-world_arm64
```

Demonstration

System Worker is ready to Go

The Go Programming Language

https://golang.org/

- read/write **Microsoft Excel** spreadsheets

  https://github.com/qax-os/excelize

- generate **responsive HTML e-mail**

  https://github.com/matcornic/hermes

- generate **PDF**

  https://github.com/signintech/gopdf

https://github.com/avelino/awesome-go

Question Time

System Workers, OOP, Go

Go places with

System Workers!

4D Summit 2023 presentation