Appendix

B

Source Code for Chapter 24

This appendix shows the code for the above examples in full. To save space I've done some minor things such as only show the class names and the instance variable names in the class definition.

Superclasses

```
ApplicationModel subclass: #MyInput
	instanceVariableNames: 'person '

MyInput class>>open: aModel
	^self openOn: (self new initialize: aModel)

ApplicationModel subclass: #MyView
	instanceVariableNames: 'person '

MyView class>>open: aModel
	^self openOn: (self new initialize: aModel)

Model subclass: #MyPerson
	instanceVariableNames: 'name age '
```

Example One

```
MyInput subclass: #MyInput1
	instanceVariableNames: 'name age '

MyInput1>>initialize: aPerson
	person := aPerson.
	name := String new asValue.
	name onChangeSend: #nameChanged to: self.
	age := 0 asValue.
	age onChangeSend: #ageChanged to: self

MyInput1>>age
	^age

MyInput1>>name
	^name

MyInput1>>ageChanged
	person age: self age value
```

```
MyInput1>>nameChanged
	person name: self name value

MyView subclass: #MyView1
	instanceVariableNames: 'name age '

MyView1>>initialize: aPerson
	person := aPerson.
	person addDependent: self.
	name := String new asValue.
	age := 0 asValue.

MyView1>>age
	^age

MyView1>>name
	^name

MyView1>>update: aSymbol with: aValue from: anObject
	aSymbol == #name ifTrue: [self name value: aValue].
	aSymbol == #age ifTrue: [self age value: aValue]

MyPerson subclass: #MyPerson1
	instanceVariableNames: ' '

MyPerson1>>age: aValue
		age := aValue.
	self changed: #age with: age

MyPerson1>>name: aValue
	name := aValue.
	self changed: #name with: name
```

Example Two

```
MyInput subclass: #MyInput2
	instanceVariableNames: 'name age '

MyInput2>>initialize: aPerson
	person := aPerson.
	name := String new asValue.
	name onChangeSend: #nameChanged to: self.
	age := 0 asValue.
	age onChangeSend: #ageChanged to: self.

MyInput2>>age
	^age

MyInput2>>name
	^name

MyInput2>>ageChanged
	person age value: self age value

MyInput2>>nameChanged
	person name value: self name value

MyView subclass: #MyView2
	instanceVariableNames: 'name age '

MyView2>>initialize: aPerson
	person := aPerson.
	person name onChangeSend: #nameChanged to: self.
	person age onChangeSend: #ageChanged to: self.
	name := String new asValue.
	age := 0 asValue.

MyView2>>age
	^age

MyView2>>name
	^name

MyView2>>ageChanged
	self age value: person age value.
```

```
MyView2>>nameChanged
	self name value: person name value.

MyPerson subclass: #MyPerson2
	instanceVariableNames: ' '

MyPerson2>>age
	^age isNil
		ifTrue: [age := 0 asValue]
		ifFalse: [age]

MyPerson2>>name
	^name isNil
		ifTrue: [name := String new asValue]
		ifFalse: [name]
```

Example Three

```
MyInput subclass: #MyInput3
	instanceVariableNames: ''

MyInput3>>initialize: aPerson
	person := aPerson.

MyInput3>>age
	^(AspectAdaptor subject: person sendsUpdates: true)
		forAspect: #age.

MyInput3>>name
	^(AspectAdaptor subject: person sendsUpdates: true)
		forAspect: #name.

MyView subclass: #MyView3
	instanceVariableNames: ''

MyView3>>initialize: aPerson
	person := aPerson

MyView3>>age
	^(AspectAdaptor subject: person sendsUpdates: true)
		forAspect: #age.

MyView3>>name
	^(AspectAdaptor subject: person sendsUpdates: true)
		forAspect: #name.

MyPerson subclass: #MyPerson3
	instanceVariableNames: ' '

MyPerson3>>age
	^age

MyPerson3>>age: aValue
	age := aValue.
	self changed: #age

MyPerson3>>name
	^name

MyPerson3>>name: aValue
	name := aValue.
	self changed: #name
```

Example Four

```
MyInput subclass: #MyInput4
	instanceVariableNames: ''

MyInput4>>initialize: aPerson
	person := aPerson asValue

MyInput4>>age
	| adaptor |
```

```
	adaptor := AspectAdaptor subjectChannel: person sendsUpdates: true.
	adaptor
		accessWith: #yearsOld
		assignWith: # yearsOld:
		aspect: #age.
	^adaptor

MyInput4>>name
	| adaptor |
	adaptor := AspectAdaptor subjectChannel: person sendsUpdates: true.
	adaptor
		accessWith: # called
		assignWith: # called:
		aspect: #name.
	^adaptor

MyView subclass: #MyView4
	instanceVariableNames: ''

MyView4>>initialize: aPerson
	person := aPerson asValue

MyView4>>age
	| adaptor |
	adaptor := AspectAdaptor subjectChannel: person sendsUpdates: true.
	adaptor
		accessWith: # yearsOld
		assignWith: # yearsOld:
		aspect: #age.
	^adaptor

MyView4>>name
	| adaptor |
	adaptor := AspectAdaptor subjectChannel: person sendsUpdates: true.
	adaptor
		accessWith: # called
		assignWith: # called:
		aspect: #name.
	^adaptor

MyPerson subclass: #MyPerson4
	instanceVariableNames: ' '

MyPerson4>> yearsOld
	^age

MyPerson4>> yearsOld: aValue
	age := aValue.
	self changed: #age

MyPerson4>>called
	^name

MyPerson4>> called: aValue
	name := aValue.
	self changed: #name
```

Example Five

```
MyInput subclass: #MyInput5
	instanceVariableNames: 'trigger '

MyInput5>>initialize: aPerson
	person := aPerson.
	trigger := false asValue

MyInput5>>accept
	trigger value: true

MyInput5>>age
	| adaptor |
	adaptor := AspectAdaptor subject: person sendsUpdates: true.
	adaptor forAspect: #age.
	^BufferedValueHolder subject: adaptor triggerChannel: trigger.
```

```
MyInput5>>name
	| adaptor |
	adaptor := AspectAdaptor subject: person sendsUpdates: true.
	adaptor forAspect: #name.
	^BufferedValueHolder subject: adaptor triggerChannel: trigger.

MyView subclass: #MyView5
	instanceVariableNames: ''

MyView5>>initialize: aPerson
	person := aPerson

MyView5>>age
	^(AspectAdaptor subject: person sendsUpdates: true)
		forAspect: #age.

MyView5>>name
	^(AspectAdaptor subject: person sendsUpdates: true)
		forAspect: #name.

Model subclass: #MyPerson5
	instanceVariableNames: ''

MyPerson5>>age
	^age

MyPerson5>>age: aValue
	age := aValue.
	self changed: #age

MyPerson5>>name
	^name

MyPerson5>>name: aValue
	name := aValue.
	self changed: #name
```

Example Six

```
MyInput subclass: #MyInput6
	instanceVariableNames: ''

MyInput6>>initialize: aPerson
	person := aPerson asValue.

MyInput6>>person
	^person

MyView subclass: #MyView6
	instanceVariableNames: ''

MyView6>>initialize: aPerson
	person := aPerson asValue.

MyView6>>person
	^person

Model subclass: #MyPerson6
	instanceVariableNames: ' '

MyPerson6>>age
	^age

MyPerson6>>age: aValue
	age := aValue.
	self changed: #age

MyPerson6>>name
	^name

MyPerson6>>name: aValue
	name := aValue.
	self changed: #name
```

Example Seven

```
MyInput subclass: #MyInput7
	instanceVariableNames: 'trigger '

MyInput7>>initialize: aPerson
	person := aPerson asValue.
	trigger := false asValue.

MyInput7>>person
	^person

MyInput7>>trigger
	^trigger

MyInput7>>accept
	trigger value: true.

MyView subclass: #MyView7
	instanceVariableNames: ''

MyView7>>initialize: aPerson
	person := aPerson asValue.

MyView7>>person
	^person

MyPerson subclass: #MyPerson7
	instanceVariableNames: ''

MyPerson7>>age
	^age

MyPerson7>>age: aValue
	age := aValue.
	Transcript cr; show: 'Age changed'.
	self changed: #age

MyPerson7>>name
	^name

MyPerson7>>name: aValue
	name := aValue.
	Transcript cr; show: 'Name changed'.
	self changed: #name
```

Example Eight

```
MyInput subclass: #MyInput8
	instanceVariableNames: ''

MyInput8>>initialize: aPerson
	person := aPerson.

MyInput8>>age
	^(PluggableAdaptor on: person)
		getBlock: [:model | model age]
		putBlock: [:model :aValue | model age: aValue * 3]
		updateBlock: [:model :aspect :parameter | aspect == #age]

MyInput8>>name
	^(PluggableAdaptor on: person)
		getBlock: [:model | model name]
		putBlock: [:model :aValue | model name: aValue asLowercase]
		updateBlock: [:model :aspect :parameter | aspect == #name]

MyView subclass: #MyView8
	instanceVariableNames: ''

MyView8>>initialize: aPerson
	person := aPerson.

MyView8>>age
	^(PluggableAdaptor on: person)
		getBlock: [:model | model age * 10]
```

```
            putBlock: [:model :aValue | ]
            updateBlock: [:model :aspect :parameter | aspect == #age]

MyView8>>name
      ^(PluggableAdaptor on: person)
            getBlock: [:model | model name asUppercase]
            putBlock: [:model :aValue | ]
            updateBlock: [:model :aspect :parameter | aspect == #name ]

Model subclass: #MyPerson8
      instanceVariableNames: ' '

MyPerson8>>age
      ^age isNil
            ifTrue: [age := 0]
            ifFalse: [age]

MyPerson8>>age: aNumber
      age := aNumber.
      self changed: #age

MyPerson8>>name
      ^name isNil
            ifTrue: [name := String new]
            ifFalse: [name]

MyPerson8>>name: aString
      name := aString.
      self changed: #name
```