



ESUG 2012

Extending a Base Product for Multiple Customers

Denis Defreyne

MediaGeniX NG

“*Your product is cool,
but it's missing feature X!*”

— CUSTOMER



*Yes, feature X would be nice...
for you, but nobody else!*

— YOU



*Why did you not
implement feature X? :(*

— CUSTOMER



*Ugh, this product is
getting bloated! :(*

— OTHER CUSTOMER

Contents

1. Customisations
2. Techniques
3. Modules

1. Customisations

Customisation =
extra bits of code
for a single customer

Customisations

- ▶ **New functionality**
e.g. new issue checks (no porn at noon)
- ▶ **Extended functionality**
e.g. extra metadata on films, contracts, ...
- ▶ **Changed functionality**
e.g. parental rating management in UK, Finland

Integrations

- ▶ Playout server
- ▶ Commercial scheduling system
- ▶ Music scheduling system
- ▶ ...

unique for every customer!

Customisations are
only possible if you
have **FEW** customers

Requirements

- ▶ Be customer-specific
- ▶ Be fine-grained
- ▶ Allow merging bugfixes
- ▶ Be decoupled

Why?

- ▶ **Talk to other devices**
You'll have static on your TV otherwise
- ▶ **Cater to a diverse market**
No two broadcasters have the same workflow
- ▶ **Resolve conflicting requirements**
Don't compromise, let them both have it their way
- ▶ **Make customers happy!**
But don't over-do it :)



COMEDY CENTRAL

nickelodeonTM

2. Techniques

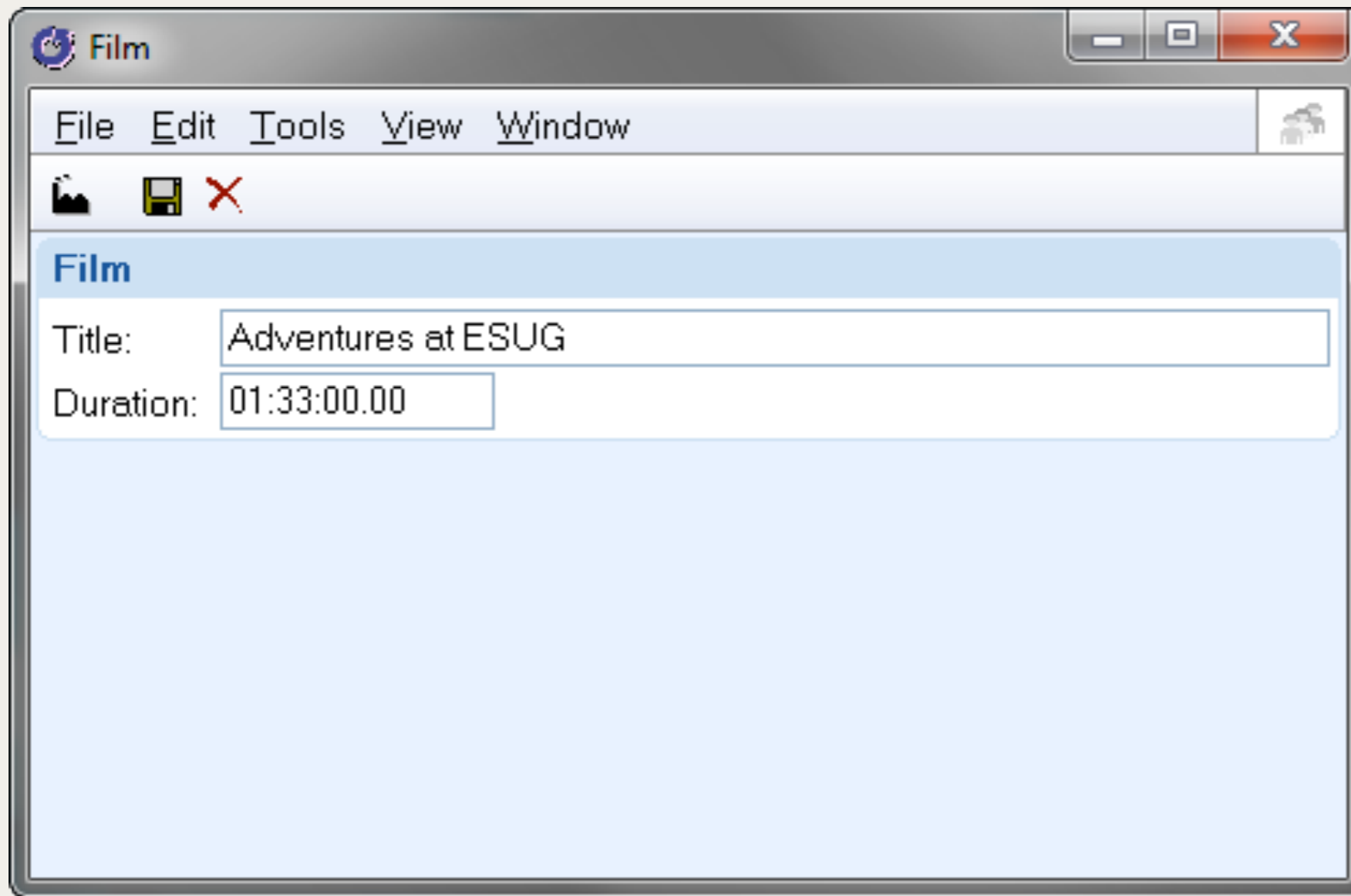
Techniques

1. Site classes
2. Behaviors
3. Convenience methods

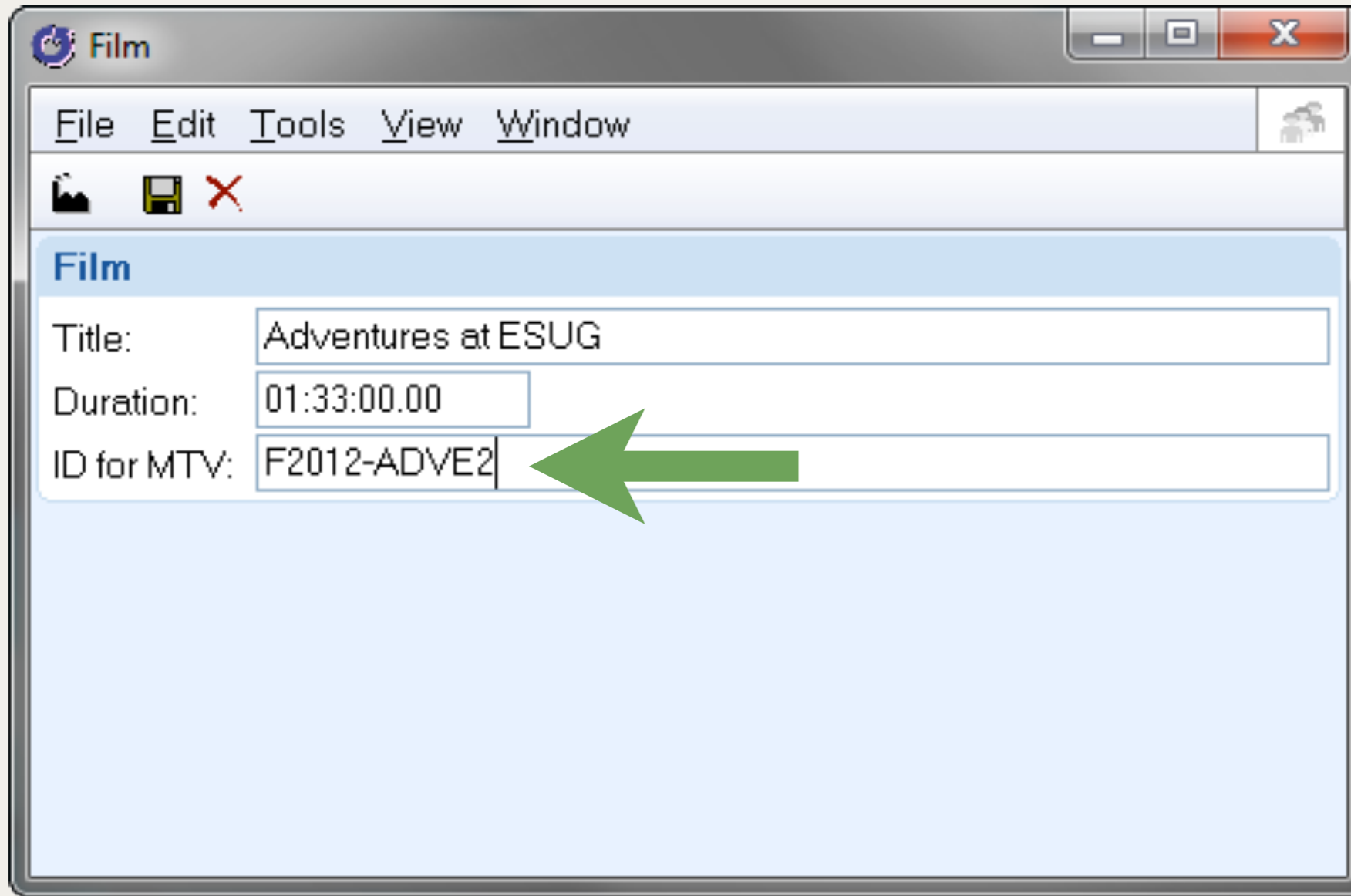
Techniques

- 1. Site classes**
2. Behaviors
3. Convenience methods

Site classes - example



Site classes - example



The screenshot shows a window titled "Film" with a menu bar (File, Edit, Tools, View, Window) and a toolbar with icons for a folder, save, and close. The main area contains a form with the following fields:

Title:	Adventures at ESUG
Duration:	01:33:00.00
ID for MTV:	F2012-ADVE2

A green arrow points to the "ID for MTV" field.

Site class =
subclass for one
specific customer

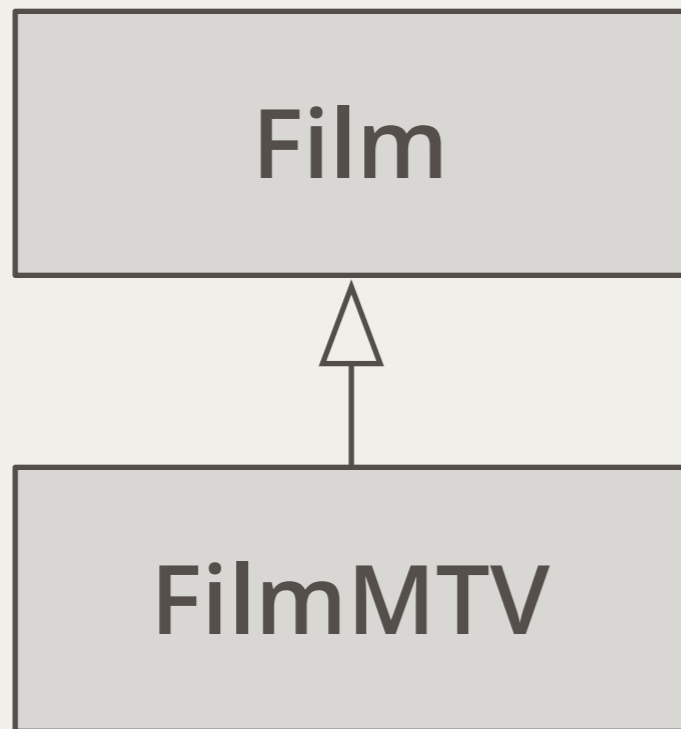
Site = customer

Site classes



Film

Site classes



Site classes

```
Film class >> siteClassForMTV
```

```
^FilmMTV
```

MTV

Site classes

```
newAustinPowersFilm
```

```
^Film siteClass new  
  title: 'Adventures at ESUG';  
  duration: (Duration fromMinutes: 93);  
  yourself
```

BASE

Packages

MTV

WHATS'On code for MTV only

BASE

WHATS'On code shared between customers

FRAMEWORK

MediaGeniX framework code

Packages

NRK

TV2

PRO7

VRT

MTV

VPRO

VTM

SBS FI

NPO

FOX

FTO

...

Extensions in packages

PACKAGE A

MusicClip
MusicClip>>title
MusicClip>>videoFile
MusicClip>>audioFile
MusicClip>>is16Plus

PACKAGE B

Contract
Agreement

MusicClip>>contract
MusicClip>>agency

Site classes

```
Film class >> siteClassForMTV
```

```
^FilmMTV
```

MTV

Site classes

BASE

Film

MusicClip

Commercial

MTV

FilmMTV

Film>>siteClassForMTV

Film >> siteClassForMTV

Film siteClass new

Site classes

```
Class >> siteClass  
  ^Site current siteClassOf: self
```

FRAMEWORK

Site classes

```
SiteMTV >> siteClassOf: aClass  
^aClass siteClassForMTV
```

MTV

Site classes

```
Site >> siteClassOf: aClass  
      ^aClass
```

BASE

Site classes – example

```
Film class >> buildSuperModelWith: aBuilder  
  
super buildSuperModelWith: aBuilder.  
  
aBuilder  
  addString: #title name: 'Title';  
  addDuration: #duration name: 'Duration'
```

BASE



Site classes – example

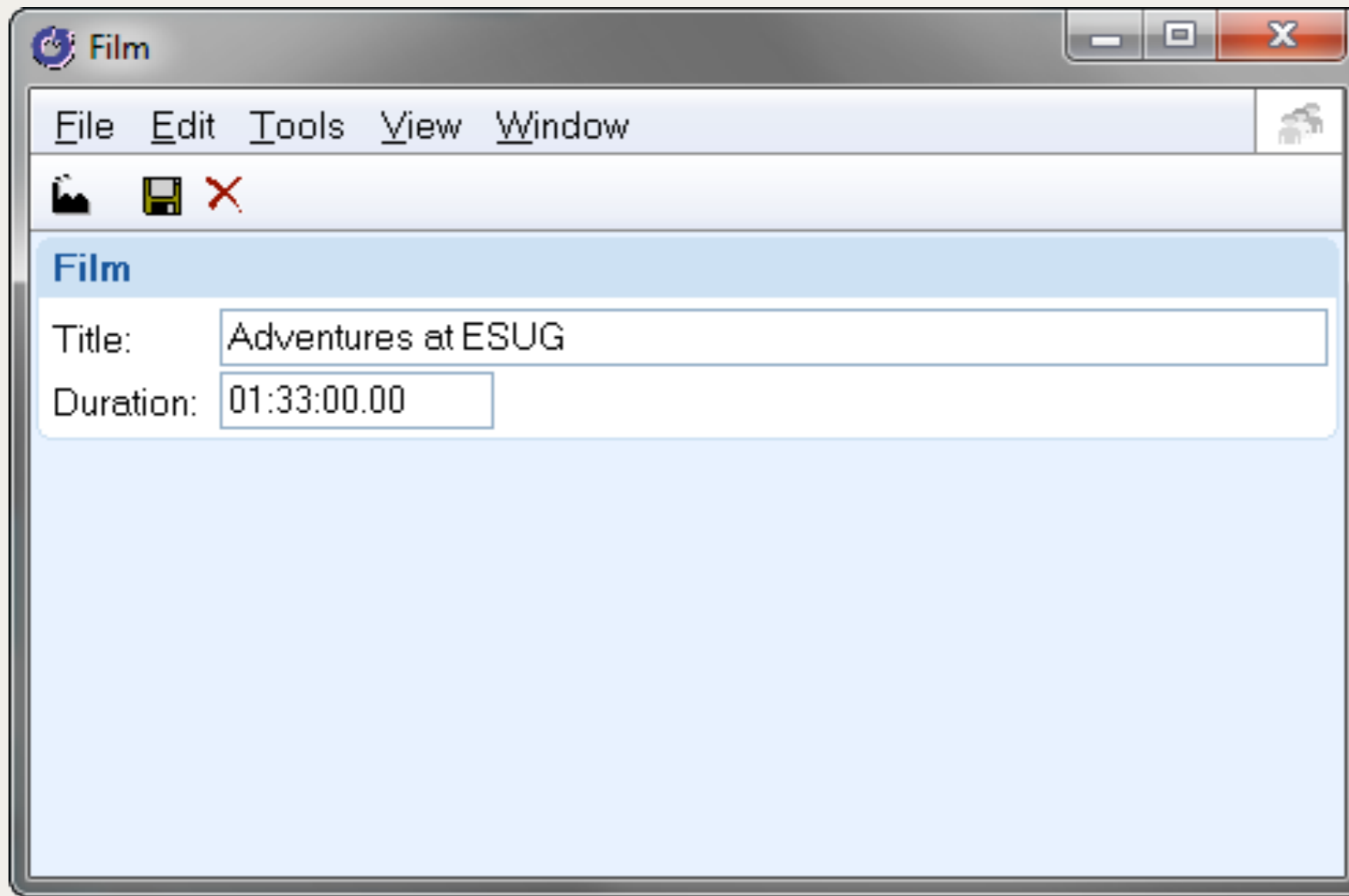
```
Film class >> buildSuperModelWith: aBuilder  
  
super buildSuperModelWith: aBuilder.  
  
aBuilder  
  addString: #title name: 'Title';  
  addDuration: #duration name: 'Duration'
```

BASE

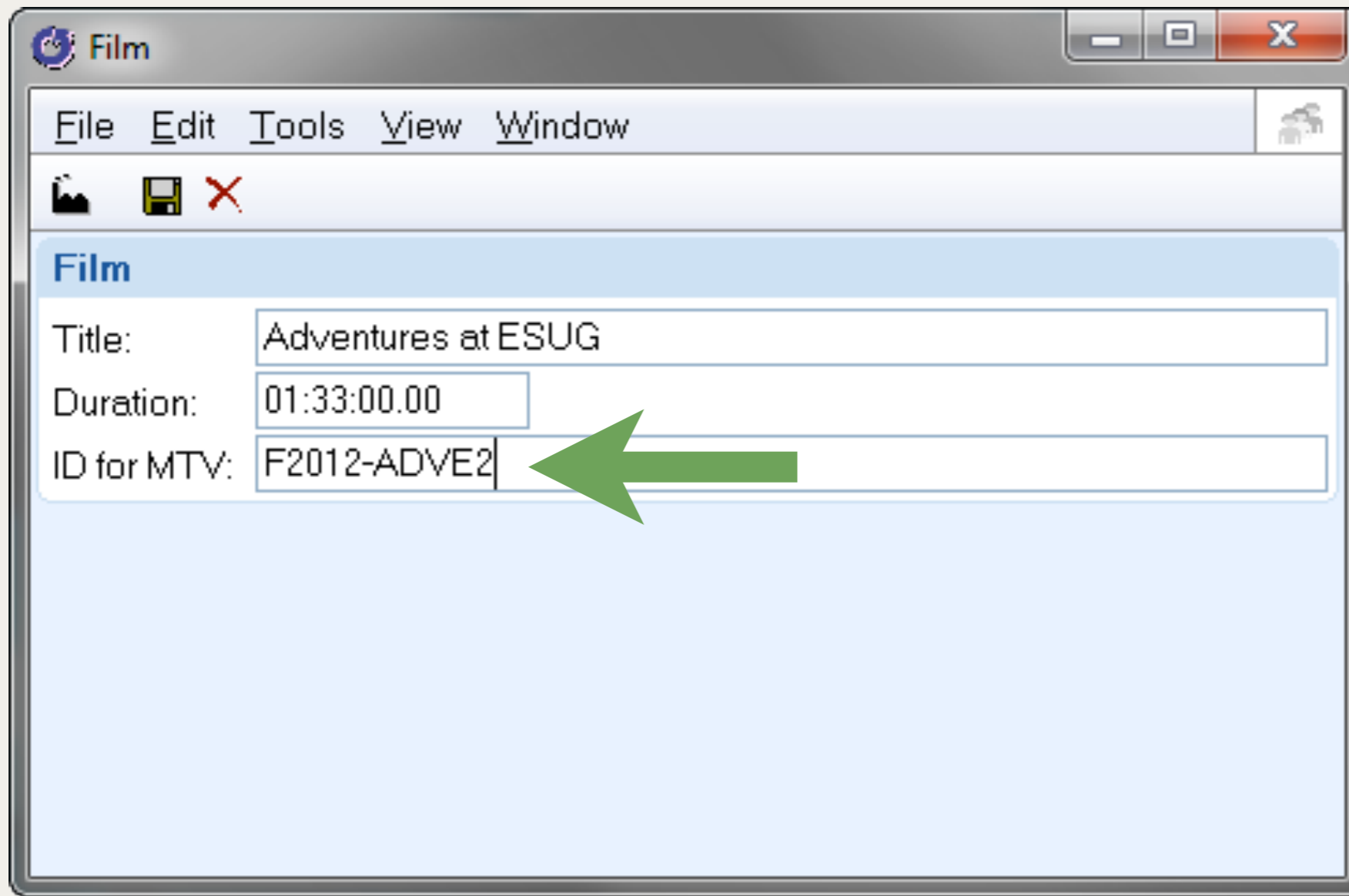
Site classes – example

```
FilmMTV class >> buildSuperModelWith: aBuilder  
  
super buildSuperModelWith: aBuilder.  
  
aBuilder  
  addString: #idForMTV name: 'ID for MTV'
```

Site classes - example



Site classes - example



The screenshot shows a window titled 'Film' with a menu bar (File, Edit, Tools, View, Window) and a toolbar with icons for a folder, save, and close. The main area contains a form with the following fields:

Title:	Adventures at ESUG
Duration:	01:33:00.00
ID for MTV:	F2012-ADVE2

A green arrow points to the 'ID for MTV' field.

Site classes – example

```
FilmEditor class >> form
```

```
  ^self newForm
```

```
    addFormField: #title;
```

```
    addFormField: #duration;
```

```
  yourself
```

BASE

Site classes – example

```
FilmEditor >> siteClassForMTV
```

```
^FilmEditorMTV
```

```
FilmEditorMTV class >> form
```

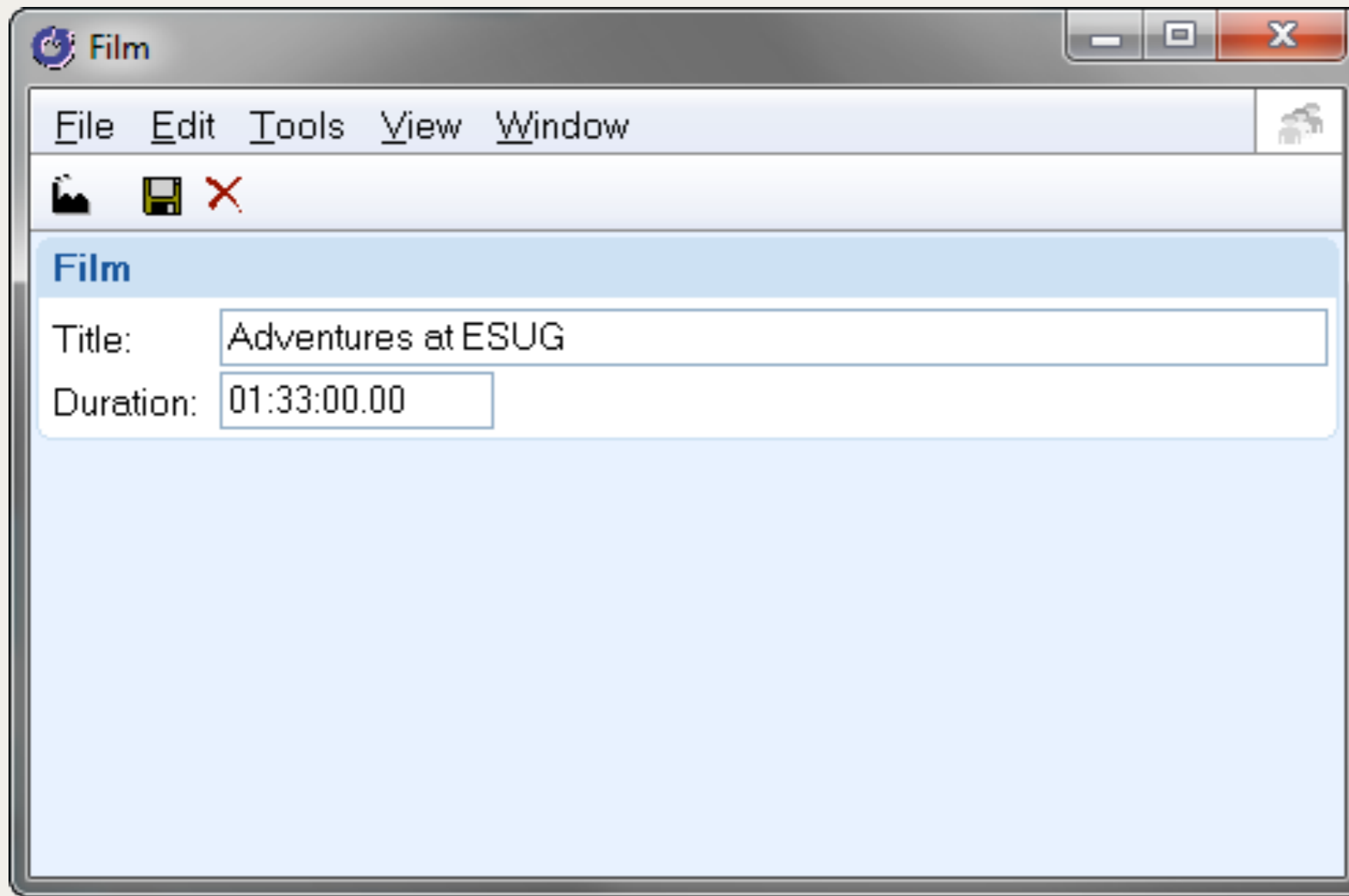
```
^super form
```

```
  addFormField: #idForMTV;
```

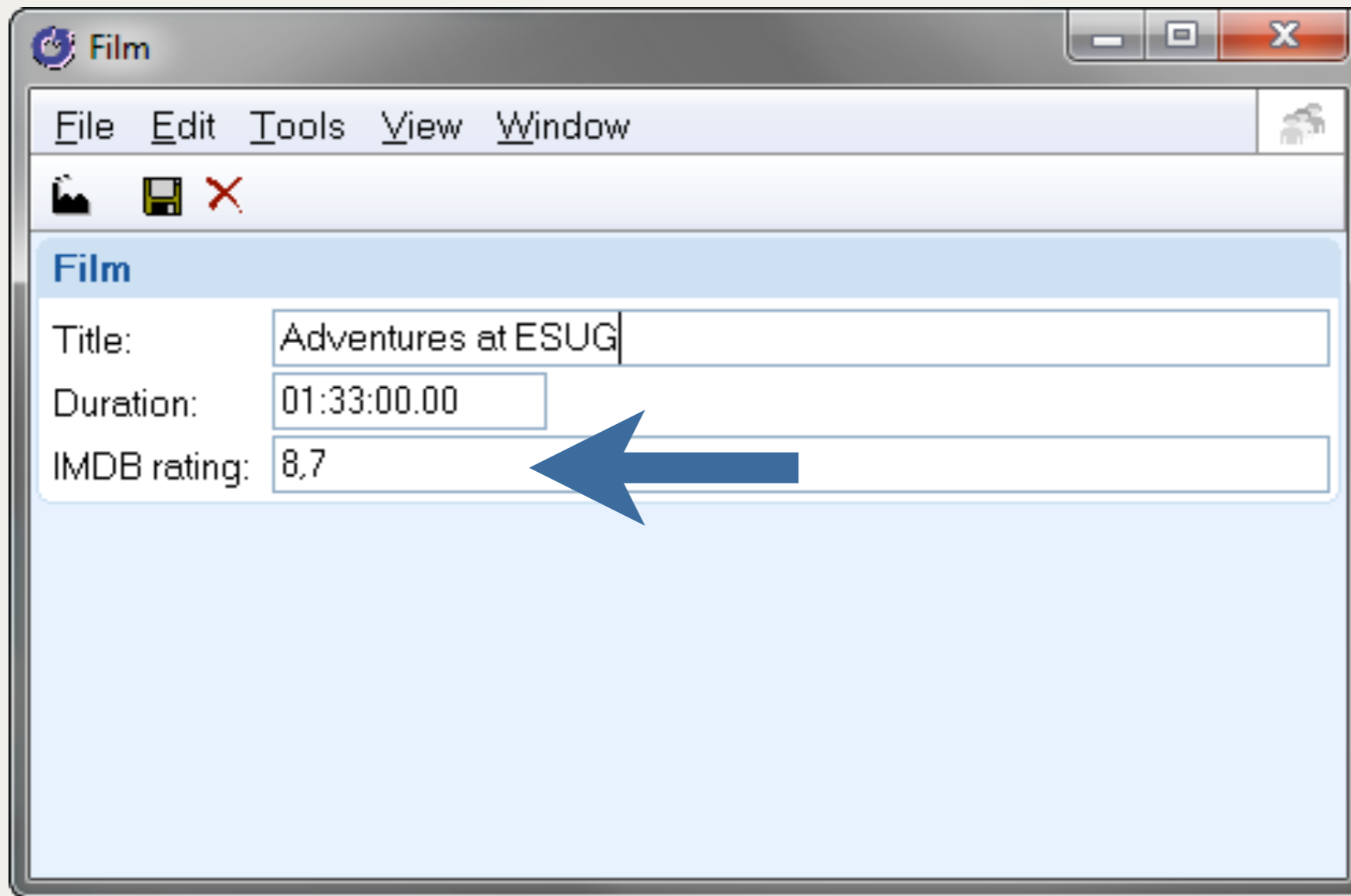
```
  yourself
```

MTV

Site classes - example



Site classes - example



The screenshot shows a window titled "Film" with a menu bar (File, Edit, Tools, View, Window) and a toolbar with icons for a folder, save, and close. The main area contains a form with the following fields:

Title:	Adventures at ESUG
Duration:	01:33:00.00
IMDB rating:	8,7

A blue arrow points to the IMDB rating field.

Site classes - example

```
FilmEditor class >> form
```

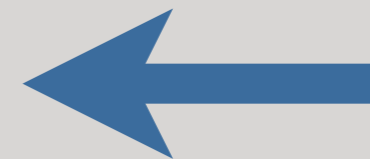
```
^self newForm
```

```
  addFormField: #title;
```

```
  addFormField: #duration;
```

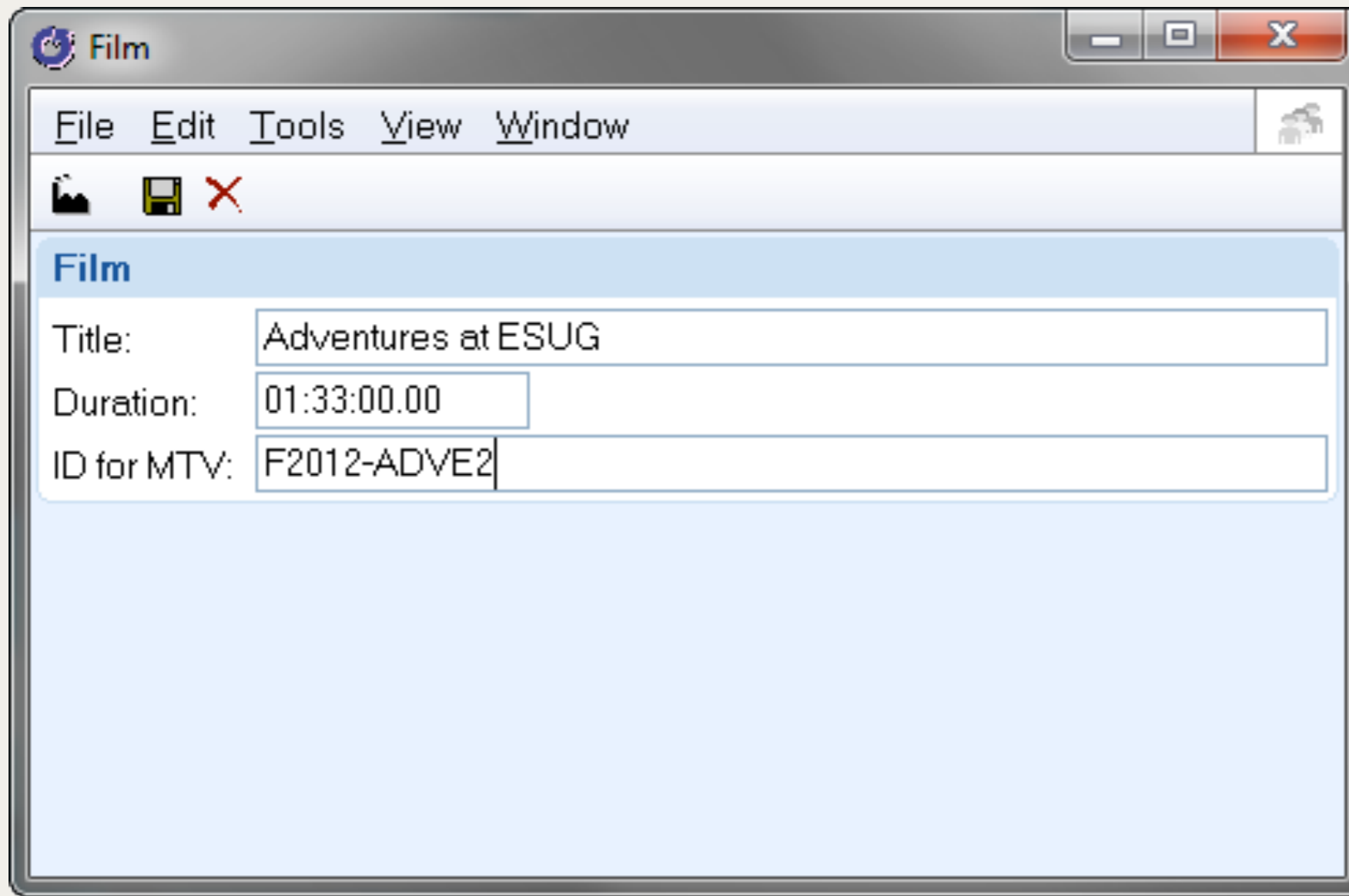
```
  addFormField: #imdbRating;
```

```
  yourself
```



BASE

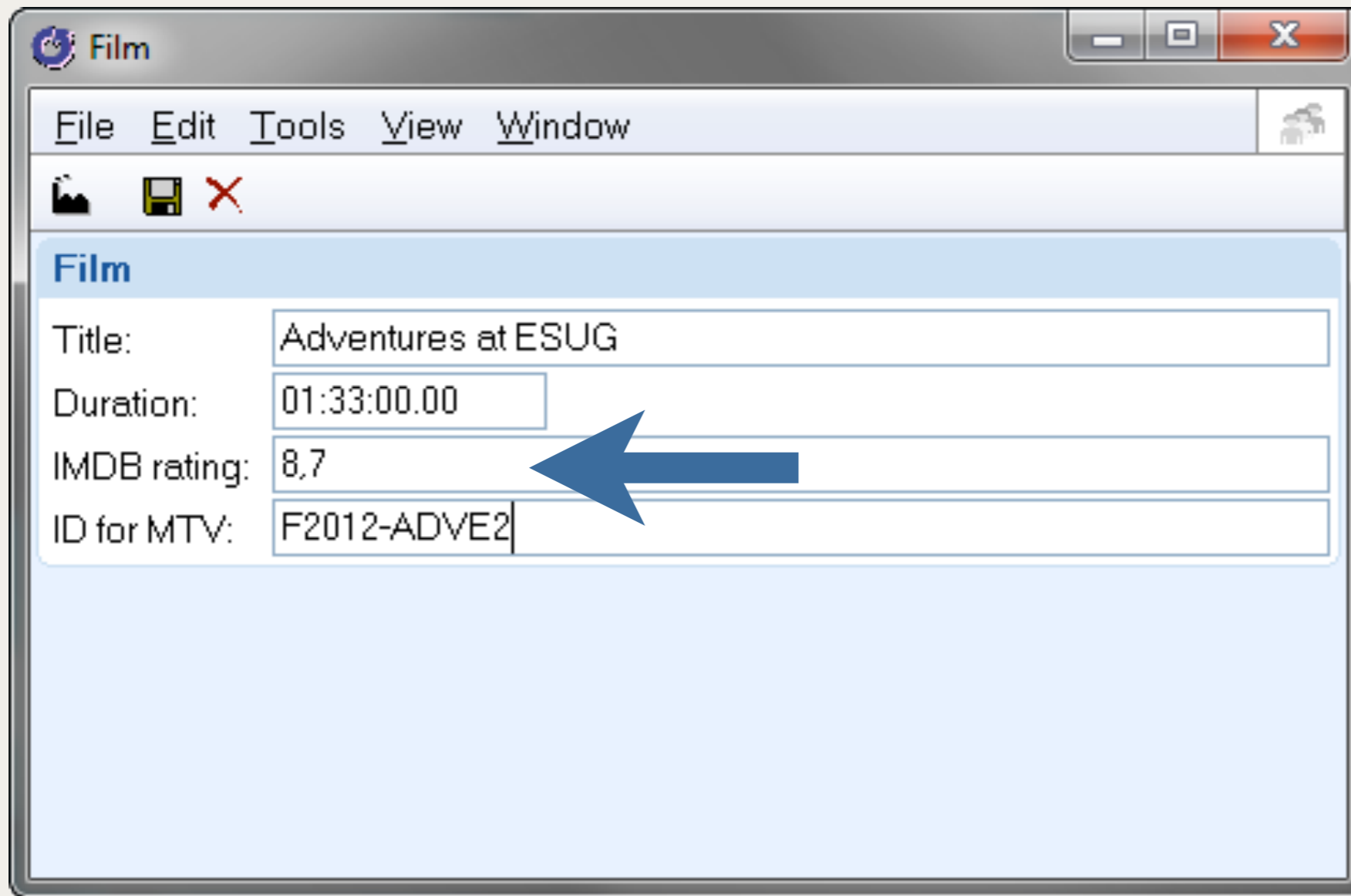
Site classes - example



The image shows a screenshot of a software application window titled "Film". The window has a standard Windows-style title bar with minimize, maximize, and close buttons. Below the title bar is a menu bar with the following items: File, Edit, Tools, View, and Window. Underneath the menu bar is a toolbar containing three icons: a film strip, a floppy disk, and a red 'X'. The main content area of the window is light blue and contains a form with three input fields:

Film	
Title:	Adventures at ESUG
Duration:	01:33:00.00
ID for MTV:	F2012-ADVE2

Site classes - example



The screenshot shows a window titled "Film" with a menu bar (File, Edit, Tools, View, Window) and a toolbar with icons for a folder, save, and close. The main area contains a form with the following fields:

Title:	Adventures at ESUG
Duration:	01:33:00.00
IMDB rating:	8,7
ID for MTV:	F2012-ADVE2

A blue arrow points to the IMDB rating field.

Site classes – example

```
Film >> openInEditor
```

```
^FilmEditor siteClass  
  openWithEditee: self
```

BASE

Call #siteClass everywhere?

YES!

It's not as bad as you may think...

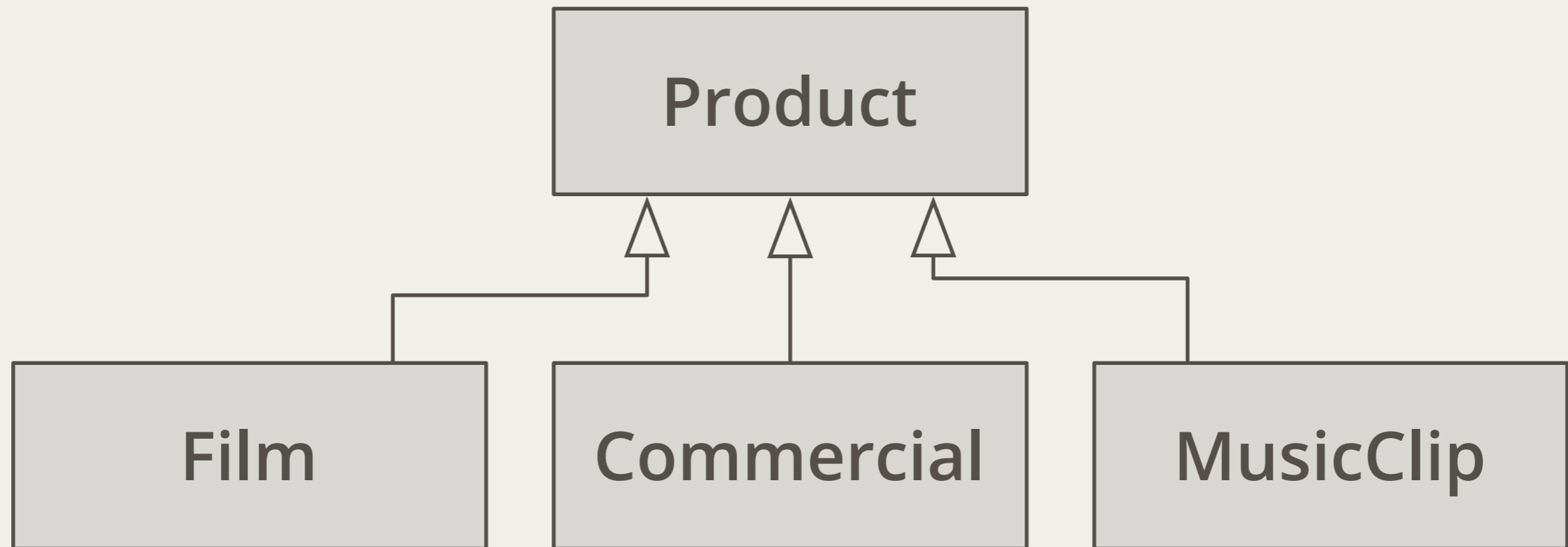
PRO TIP

Avoid long methods

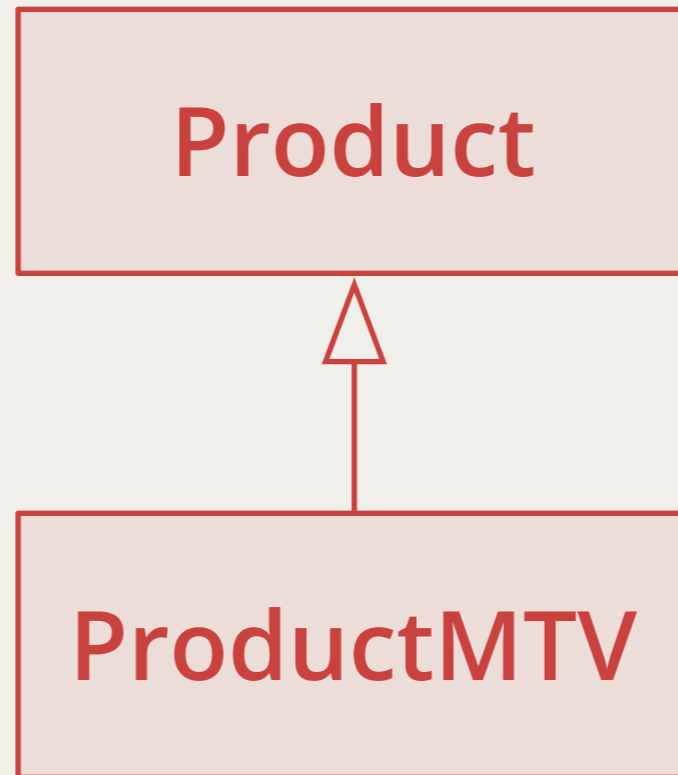
Techniques

1. Site classes
- 2. Behaviors**
3. Convenience methods

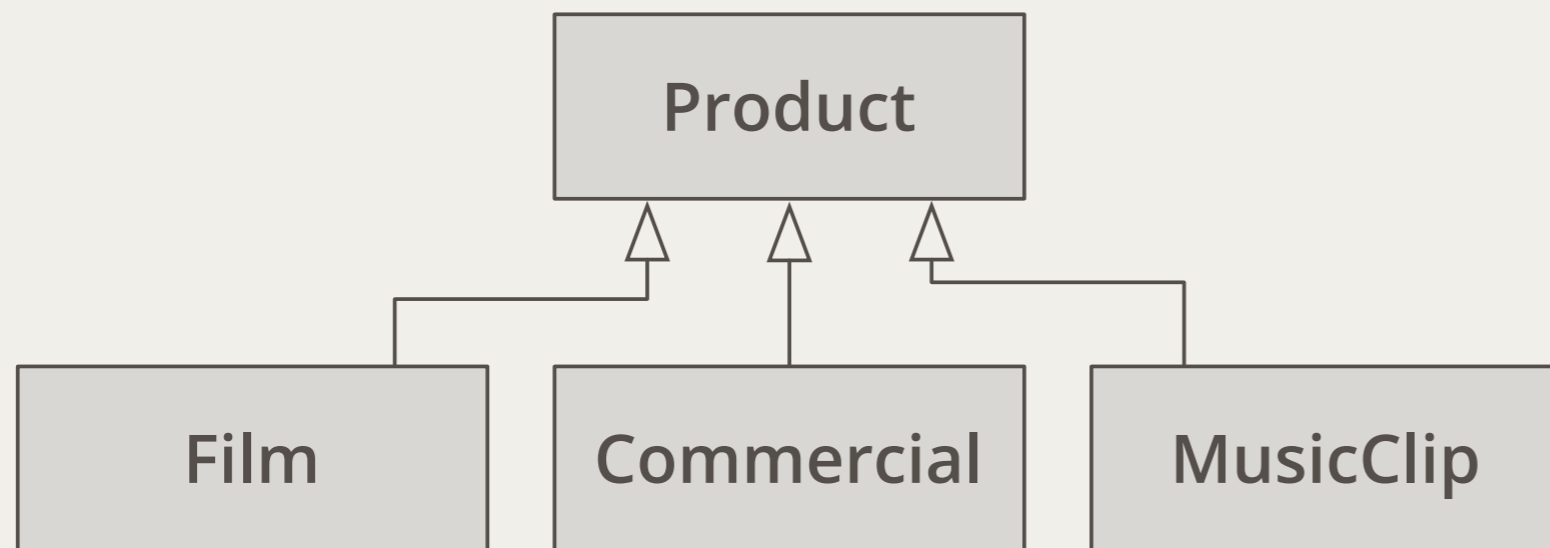
Behaviors



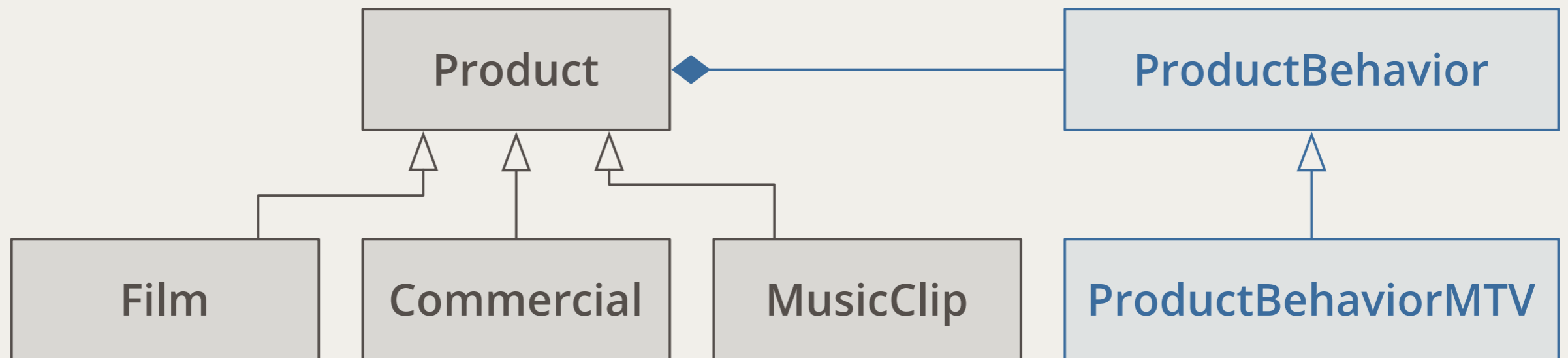
Behaviors



Behaviors



Behaviors



Behavior class =
siteclassable class with
behavior from original class

Behaviors

```
BehaviorObject class >> newOn: anObject
```

```
  ^self new  
    orig: anObject;  
    yourself
```

FRAMEWORK

Behaviors – example

```
Product >> hasValidCertForTx: aTx
```

```
  ^self certification isOkayForTx: aTx
```

BASE

Behaviors – example

```
ProductBehavior >> hasValidCertForTx: aTx
```

```
  ^self orig certification isOkayForTx: aTx
```

BASE

Behaviors – example

```
ProductBehaviorMTV >> hasValidCertForTx: aTx
```

```
^self orig videoFiles allSatisfy:  
  [:f | f certification isOkayForTx: aTx]
```

MTV

Behaviors – example

```
Product >> hasValidCertForTx: aTx
```

```
  ^self behavior hasValidCertForTx: aTx
```

BASE

Behaviors – example

```
Product >> behavior
```

```
behavior ifNil:  
  [behavior := ProductBehavior siteClass  
    newOn: self].  
^behavior
```

BASE

PRO TIP

**Prefer composition
over inheritance**

Techniques

1. Site classes
2. Behaviors
- 3. Convenience methods**

Convenience method =
commonly implemented
or overridden method

Convenience methods

buildSuperModelWith: `aBuilder`

addToolsMenuItemsTo: `aMenu`

addCommandsToToolBar: `aToolBar`

Convenience methods

buildSuperModelWith: **aBuilder**

buildSuperModelForMTVWith: **aBuilder**

addToolsMenuItemsTo: **aMenu**

addToolsMenuItemsForMTVTo: **aMenu**

addCommandsToToolBar: **aToolBar**

addCommandsToToolBarForMTV: **aToolBar**

Convenience methods

```
FilmMTV class >> buildSuperModelWith: aBuilder  
  
super buildSuperModelWith: aBuilder.  
  
aBuilder  
  addString: #idForMTV name: 'ID for MTV'
```

Convenience methods

```
Film class >> buildSuperModelForMTVWith: aBuilder  
  
    super buildSuperModelForMTVWith: aBuilder.  
  
    aBuilder  
        addString: #idForMTV name: 'ID for MTV'
```

Convenience methods

```
Site >> buildSuperModelWith: aBuilder  
        for: aClass
```

```
aClass buildSuperModelWith: aBuilder
```

FRAMEWORK

Convenience methods

```
SiteMTV >> buildSuperModelWith: aBuilder  
           for: aClass
```

```
super buildSuperModelWith: aBuilder for: aClass.
```

```
aClass buildSuperModelForMTVWith: aBuilder.
```


PRO TIP

Use convenience methods
for common methods only

Techniques to avoid

- ▶ ~~Branches~~

Use them, but not for customer-specific code

- ▶ ~~VisualWorks "overrides"~~

They prevent access to original code

Techniques

1. Site classes
2. Behaviors
3. Convenience methods

3. Modules

Module =
pluggable component
that provides a set of
related functionalities

Example modules

- ▶ Contract management
- ▶ Secondary events (logos)
- ▶ Video on demand

Super models in modules

```
Film class >> buildSuperModelWith: aBuilder
```

```
super
```

```
    buildSuperModelWith: aBuilder.
```

```
aBuilder
```

```
    "... other attributes here ..."
```

```
    addReference: #contract
```

```
        to: Contract name: 'Contract'
```

WRONG

BASE

Activating modules

```
SiteMTV >> moduleClasses
```

```
  ^OrderedCollection new  
    add: ContractModule;  
    add: SecondaryEventsModule;  
    yourself
```

MTV

Super models in modules

Super models in modules

```
Film class >>
  buildSuperModelForContractModuleWith: aBuilder

super
  buildSuperModelForContractModuleWith: aBuilder.

aBuilder
  addReference: #contract
    to: Contract name: 'Contract'
```

BASE

Super models in modules

```
Site >> buildSuperModelWith: aBuilder  
        for: aClass
```

```
aClass buildSuperModelWith: aBuilder.
```

FRAMEWORK

Super models in modules

```
Site >> buildSuperModelWith: aBuilder  
        for: aClass
```

```
aClass buildSuperModelWith: aBuilder.
```

```
self modules do:  
    [:m |  
    m buildSuperModelWith: aBuilder for: aClass].
```

FRAMEWORK

Super models in modules

```
Module >> buildSuperModelWith: aBuilder  
          for: aClass
```

"nothing by default"

FRAMEWORK

Super models in modules

```
ContractPresentModule >> buildSuperModelWith: aBuilder  
                           for: aClass
```

```
aClass buildSuperModelForContractModuleWith:  
        aBuilder.
```

BASE

Extensions in modules

Extensions in modules

```
FilmEditor class >> form
```

```
form := self newForm.
```

```
form addFormField: #title.
```

```
form addFormField: #duration.
```

```
^form
```

BASE

Extensions in modules

```
FilmEditor class >> form
```

```
form := self newForm.
```

```
form addFormField: #title.
```

```
form addFormField: #duration.
```

```
form addFormField: #contract.
```

```
^form
```

WRONG

BASE

Extensions in modules

```
FilmEditor class >> form  
  
form := self newForm.  
form addFormField: #title.  
form addFormField: #duration.  
ContractModule current  
    addFieldsTo: form for: self.  
^form
```

BASE

Extensions in modules

```
ContractPresentModule >>
```

```
  addFieldsTo: aForm
```

```
  for: anEditor
```

```
  anEditor addContractFieldsTo: aForm
```

BASE

Extensions in modules

```
FilmEditor >>  
  addContractFieldsTo: aForm  
  
  aForm addFormField: #contract.
```

BASE

Extensions in modules

```
ContractAbsentModule >>
```

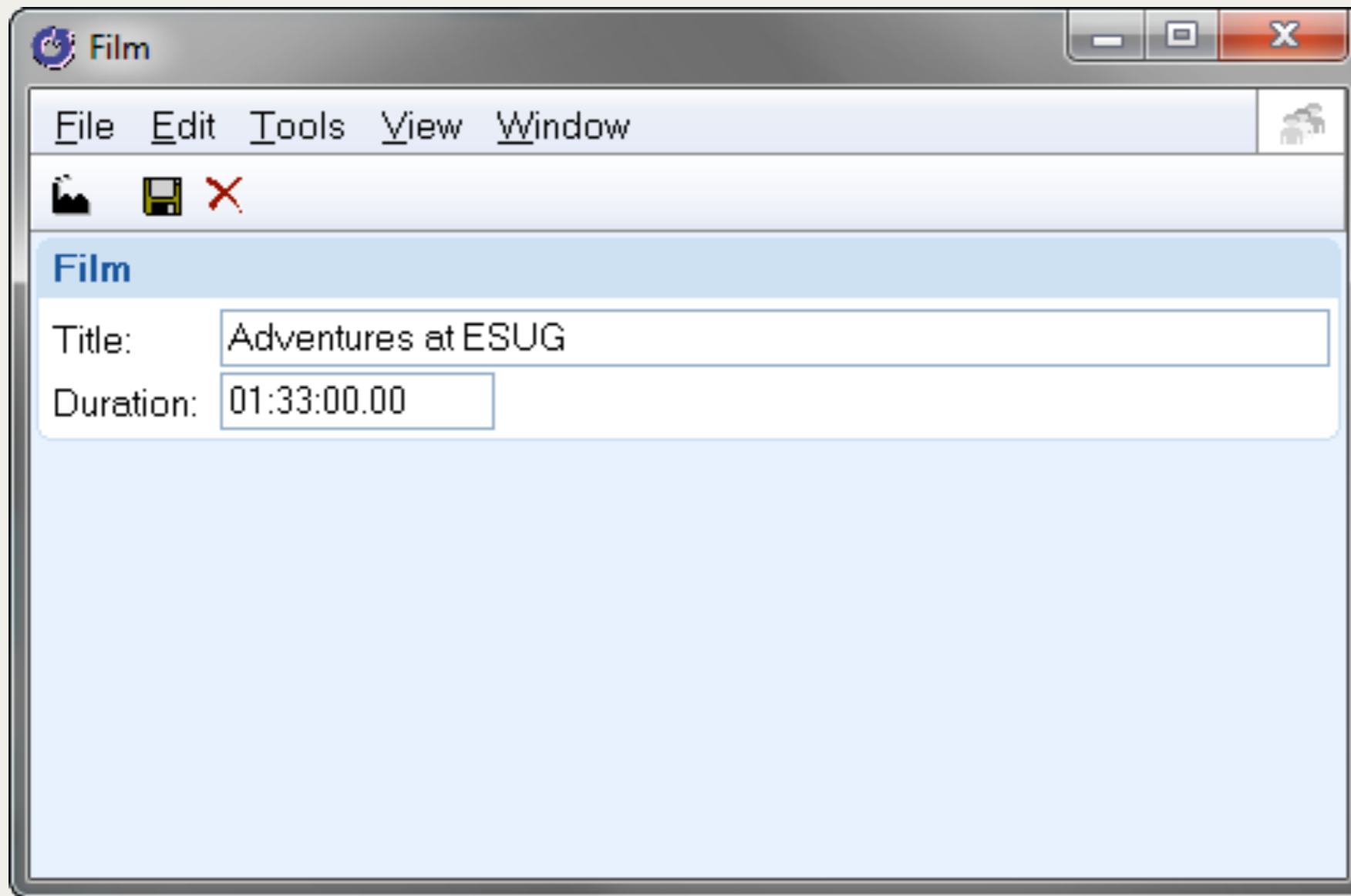
```
  addFieldsTo: aForm
```

```
  for: aSender
```

```
  "nothing by default"
```

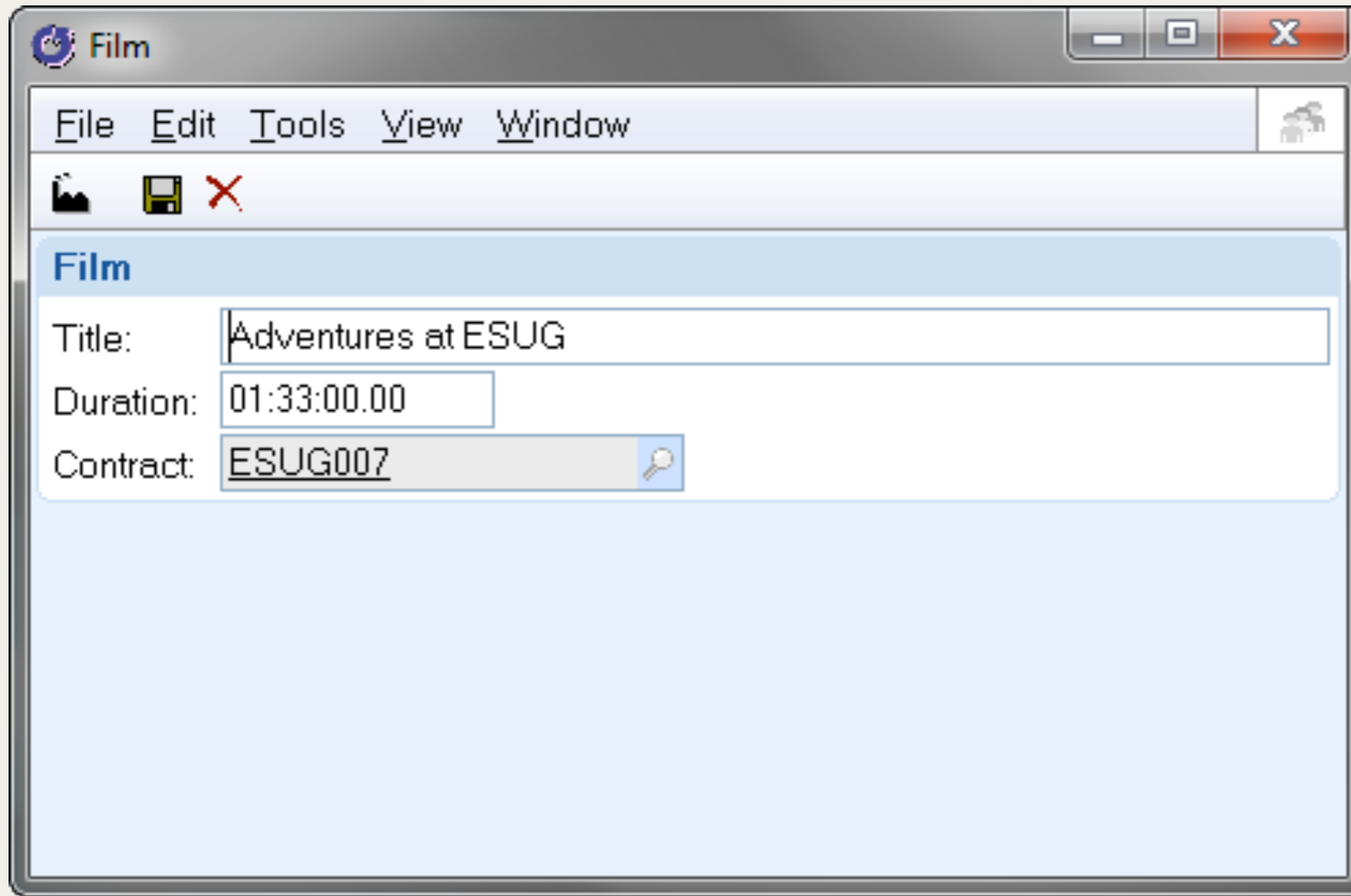
BASE

Extensions in modules



CONTRACT

Extensions in modules



CONTRACT

Site classes vs. modules

- ▶ Modules can be shared across sites
- ▶ Modules require hooks

Recap

Techniques

- ▶ **Use site classes**
whenever possible
- ▶ **Use behaviors**
when site classes fail (inheritance trees)
- ▶ **Use modules**
when you need more separation

Requirements

- ▶ Be customer-specific (with packages)
- ▶ Be fine-grained (with site classes)
- ▶ Allow merging bugfixes (with packages)
- ▶ Be decoupled (with site classes and modules)



*Your product is cool,
but it's missing feature X!*

— CUSTOMER



*Do not worry.
We'll implement it for you!*

— YOU



MAIL denis.defreyne@mediagenix.tv

TWITTER [@ddfreyne](https://twitter.com/ddfreyne)

GITHUB github.com/ddfreyne

Extra Slides

Site classing modules

```
Site >> modules
```

```
^self moduleClasses collect:  
[:c | c siteClass new]
```

FRAMEWORK

Site classing modules

```
SuperModellary >> buildSuperModelFor: aClass
```

```
builder := SuperModelBuilder new.
```

```
Site current buildSuperModelWith: builder  
for: aClass.
```

```
supermodels at: aClass put: builder superModel.
```

FRAMEWORK

Extensions in modules

```
FilmEditor class >> form  
  
form := self newForm.  
form addFormField: #title.  
form addFormField: #duration.  
ContractModule current  
    addFieldsTo: form for: self.  
^form
```

BASE

Extensions in modules

```
FilmEditor class >> form
```

```
form := self newForm.
```

```
form addFormField: #title.
```

```
form addFormField: #duration.
```

```
ContractModule current isAvailable ifTrue:
```

```
    [form addFormField: #contract].
```

```
^form
```

BASE

Extensions in modules

```
Film >> isLegallyBroadcastable
```

```
^ContractModule current isAvailable  
  ifFalse: [true]  
  ifTrue: [self contract isSigned].
```

BASE

Extensions in modules

```
Film >> isLegallyBroadcastable
```

```
^ContractModule current isAvailable  
  ifFalse: [true]  
  ifTrue: [self contract isSigned].
```

BASE

Extensions in modules

```
Film >> isLegallyBroadcastable
```

```
^ContractModule current
```

```
  isFilmLegallyBroadcastable: self
```

BASE

Extensions in modules

```
ContractPresentModule >>  
  isFilmLegallyBroadcastable: aFilm  
  
  ^aFilm contract isSigned
```

BASE

Extensions in modules

```
ContractAbsentModule >>  
  isFilmLegallyBroadcastable: aFilm  
  
  ^true
```

BASE