NAME

MooseX::Emulate::Class::Accessor::Fast – Emulate Class::Accessor::Fast behavior using Moose attributes

SYNOPSYS

```
package MyClass;
use Moose;

with 'MooseX::Emulate::Class::Accessor::Fast';


#fields with readers and writers
__PACKAGE__->mk_accessors(qw/field1 field2/);
#fields with readers only
__PACKAGE__->mk_ro_accessors(qw/field3 field4/);
#fields with writers only
__PACKAGE__->mk_wo_accessors(qw/field5 field6/);
```

DESCRIPTION

This module attempts to emulate the behavior of Class::Accessor::Fast as accurately as possible using the Moose attribute system. The public API of `Class::Accessor::Fast` is wholly supported, but the private methods are not. If you are only using the public methods (as you should) migration should be a matter of switching your `use base` line to a `with` line.

While I have attempted to emulate the behavior of Class::Accessor::Fast as closely as possible bugs may still be lurking in edge-cases.

BEHAVIOR

Simple documentation is provided here for your convenience, but for more thorough documentation please see Class::Accessor::Fast and Class::Accessor.

A note about introspection

Please note that, at this time, the `is` flag attribute is not being set. To determine the `reader` and `writer` methods using introspection in later versions of Class::MOP (> 0.38) please use the `get_read_method` and `get_write_method` methods in Class::MOP::Attribute. Example

```
# with Class::MOP <= 0.38
my $attr = $self->meta->find_attribute_by_name($field_name);
my $reader_method = $attr->reader || $attr->accessor;
my $writer_method = $attr->writer || $attr->accessor;

# with Class::MOP > 0.38
my $attr = $self->meta->find_attribute_by_name($field_name);
my $reader_method = $attr->get_read_method;
my $writer_method = $attr->get_write_method;
```

METHODS

BUILD `$self %args`

Change the default Moose class building to emulate the behavior of C::A::F and store arguments in the instance hashref.

mk_accessors `@field_names`

Create read-write accessors. An attribute named `$field_name` will be created. The name of the c<reader> and `writer` methods will be determined by the return value of `accessor_name_for` and `mutator_name_for`, which by default return the name passed unchanged. If the accessor and mutator names are equal the `accessor` attribute will be passes to Moose, otherwise the `reader` and `writer` attributes will be passed. Please see Class::MOP::Attribute for more information.

mk_ro_accessors `@field_names`

Create read-only accessors.

mk_ro_accessors `@field_names`

Create write-only accessors.

follow_best_practices
> Preface readers with 'get_' and writers with 'set_'. See original Class::Accessor documentation for more information.

mutator_name_for
accessor_name_for
> See original Class::Accessor documentation for more information.

set
> See original Class::Accessor documentation for more information.

get
> See original Class::Accessor documentation for more information.

meta
> See Moose::Meta::Class.

SEE ALSO
> Moose, Moose::Meta::Attribute, Class::Accessor, Class::Accessor::Fast, Class::MOP::Attribute, MooseX::Adopt::Class::Accessor::Fast

AUTHORS
> Guillermo Roditi (groditi) <groditi AT cpan DOT org>
>
> With contributions from:
>
> Tomas Doran (t0m) <bobtfish AT bobtfish DOT net>
> Florian Ragwitz (rafl) <rafl AT debian DOT org>

LICENSE
> You may distribute this code under the same terms as Perl itself.