Classes as constraints in Moose

Moose | isa | DateTime

Prev Next

In the previous example of the Moose series we had an attribute called "year" that was accepting a number. What if we really want to have an attribute called birthday, that contains a real date. Instead of handling the dates by ourself we would like to use a module that already handles them nicely. For example, we would like to use the DateTime module.

Download: mp4 ogv webm

Let's see the class itself in the **lib/Person.pm** file:

```
1. package Person;
   use Moose;

   has 'name'     => (is => 'rw');
5. has 'birthday' => (isa => 'DateTime', is => 'rw');

   1;
```

It has an attribute called **birthday**, but instead of it being declared as `isa => 'Int'`, we declare it as `isa => 'DateTime'`.

This expression defines that the 'birthday' attribute must be an instance of the 'DateTime' Perl class. So the constrains are not only the various internally declared types such as `Int`, but they can be any Perl class that we can load.

So when we call the setter of the 'birthday' attribute we have to pass it a DateTime object. We can do that by creating the object right in the setter:

```
1. $student->birthday( DateTime->new( year => 1988, month => 4, day => 17) );
```

Here we call the constructor of the DateTime class providing year, month, and day.

When we call the getter, it will return the DateTime object, and with `say` DateTime will stringify to some human-readable format of the date.

```
1. say $student->birthday;
```

`perl -Ilib script/person.pl`

```
1988-04-17T00:00:00
```

In the last expression of the script we called the setter again, but this time we just passed a number.

```
1. $student->birthday(1988);
```

This will throw an exception like this one:

```
Attribute (birthday) does not pass the type constraint because:
    Validation failed for 'DateTime' with value 1988
       at accessor Person::birthday (defined at lib/Person.pm line 5) line 4
    Person::birthday('Person=HASH(0x2143928)', 1988)
       called at script/person.pl line 14
```

This happens because the value passed to the `birthday` setter now needs to be a DateTime object and not just any number.

The full **script/person.pl** file:

```
 1. use strict;
    use warnings;
    use v5.10;

 5. use Person;
    use DateTime;

    my $student = Person->new( name => 'Foo' );

10. $student->birthday( DateTime->new( year => 1988, month => 4, day => 17) );
11.
    say $student->birthday;

    $student->birthday(1988);
```