Основные изменения Perl с последней YAPC::Russia v5.26–v5.41

perl-conf.ru/24

```
class Employee v3.14.1 :isa(Person::Base) {
    field $name   :param :reader //= "Unknown";
    field $office :reader = true;
    field $uuid;

    ADJUST {
        $uuid = UUID::generate();
    }

    method welcome_to($dptm //= "Sales") {
        say "$name ($uuid), welcome to $dptm";
    }

    method cache($data) {
        my $cache = Cache::Remote->open(...);
        defer { $cache->close; }
        $cache->add($data);
    }
    ...
}
```

```
use v5.40;
use experimental qw/class defer/;

class Employee v3.14.1 :isa(Person::Base) {
    field $name   :param :reader //= "Unknown";
    field $office :reader = true;
    field $uuid;

    ADJUST {
        $uuid = UUID::generate();
    }

    method welcome_to($dptm //= "Sales") {
        say "$name ($uuid), welcome to $dptm";
    }

    method cache($data) {
        my $cache = Cache::Remote->open(...);
        defer { $cache->close; }
        $cache->add($data);
    }
    ...
}
```

Perl в 2024 году

```
class Employee v3.14.1 :isa(Person::Base) {
    field $name   :param :reader //= "Unknown";
    field $office :reader = true;
    field $uuid;

    ADJUST {
        $uuid = UUID::g
    }

    method welcome_to(
        say "$name ($uu
    }

    method cache($data)
        my $cache = Ca
        defer { $cache->close; }
        $cache->add($data);
    }
    ...
}
```

Разработка Perl

Релизы новых версий Perl:

perl-5.40.0 (2024-06-09)

perl-5.38.0 (2023-07-02)

perl-5.36.0 (2022-05-28)

perl-5.34.0 (2021-05-20)

perl-5.32.0 (2020-06-20)

perl-5.30.0 (2019-05-22)

perl-5.28.0 (2018-06-23)

Разработка Perl

- С 2019 года разработка Perl ведется на GitHub

  https://github.com/Perl

Разработка Perl

- С 2019 года разработка Perl ведется на GitHub

  https://github.com/Perl
- PPCs (Proposed Perl Changes)

  https://github.com/Perl/PPCs

use VERSION

use v5.xx

- Выбор новых возможностей (feature)
- Обратная совместимость
- Обновление perl без нарушения работы кода
- Разработка языка без нарушения работы кода

use VERSION

Использование в блоках

```
{
    use v5.40;

    sub move($x //= 2, $y //= 10) {
        say "Moving to $x $y";
        ...;
    }
}
```

use VERSION

Раскрытие

```
use v5.12;
```

```
use strict;
use feature ':5.12';
```

use VERSION

Раскрытие

```
use v5.12;
```

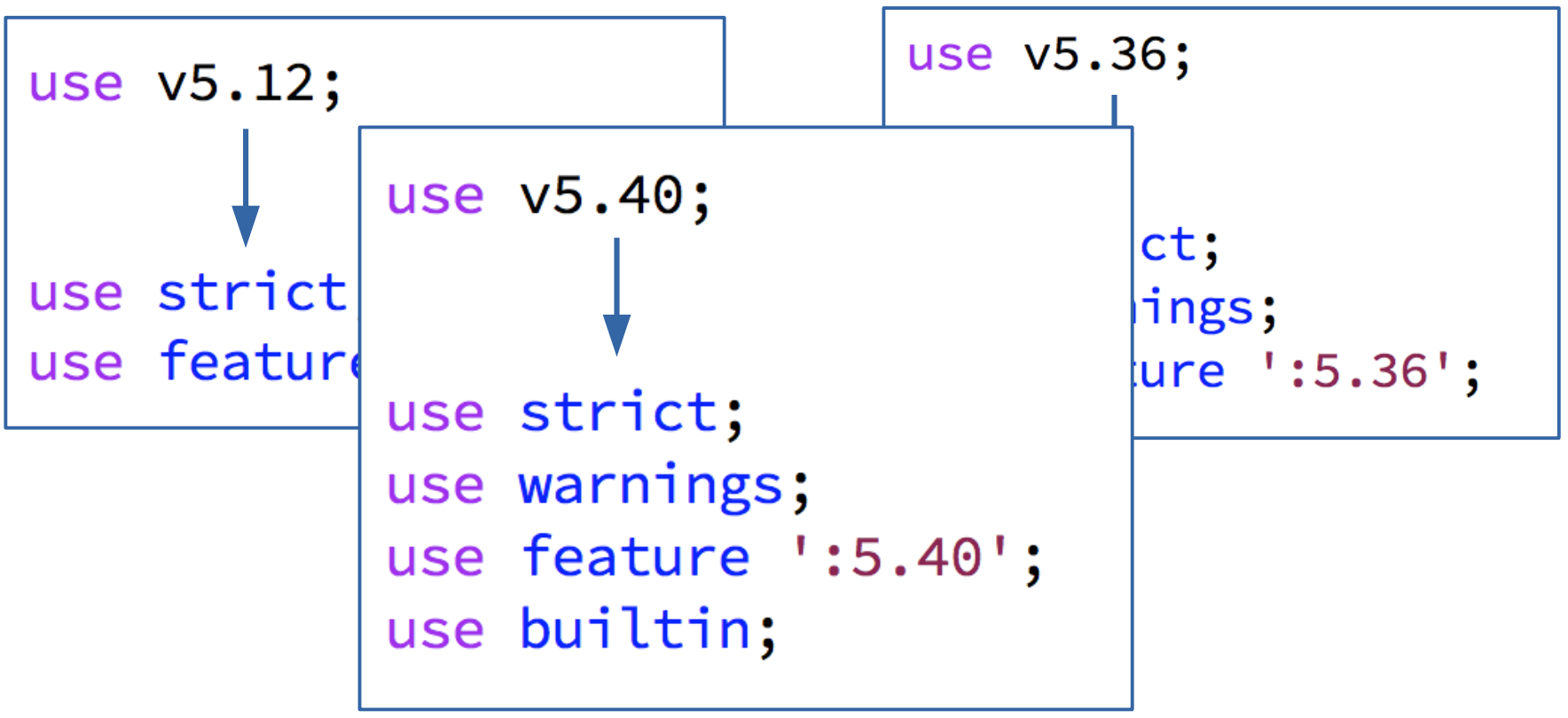
```
use strict;
use feature ':5.12';
```

```
use v5.36;
```

```
use strict;
use warnings;
use feature ':5.36';
```

use VERSION

Раскрытие



use feature

```
use feature ':5.40';

use feature ':all';

use feature qw/say state isa/;
```

isa

Perl v5.32 (experimental), **Perl v5.36** (stable)

```
use feature 'isa';
# use v5.36;

if ($obj isa Some::Class) { ... }
if ($obj isa $name_of_class) { ... }
```

try/catch/finally

Perl v5.34 (experimental), **Perl v5.40** (stable)
finally (experimental)

```
use v5.40;

try {
    ...;
}
catch ($e) {
    warn "... is failed - $e\n";
}
```

try/catch/finally

Perl v5.34 (experimental), **Perl v5.40** (stable)
finally (experimental)

```
use v5.40;

try {
    ...;
}
catch ($e) {
    warn "... is failed - $e\n";
}
finally {
    cleanup();
}
```

module_true

Perl v5.38

```
package Some::Thing;

sub new {
    ...;
}

1;
```

signatures

Perl v5.36 (stable)

```
use v5.36;

sub add($x, $y) {
    return $x + $y;
}
```

signatures

Perl v5.38

Операторы //= и ||= для аргументов сигнатур

```
use v5.38;

sub add($x, $y //= 1) {
    return $x + $y;
}
```

signatures

```
Mojo::Base -signatures;
```

defer

Perl v5.36 (experimental)

```
use feature 'defer';

sub send_data ($data) {
    my $srv = Service->connect(...);
    defer { $srv->close; }

    $srv->send($data);
    ...;
}
```

defer

Perl v5.36 (experimental)

```
use feature 'defer';

sub send_data ($data) {
    my $srv = Service->connect(...);
    defer { $srv->close; }

    $srv->send($data);
    ...;
}
```

Альтернатива: use Guard 'scope_guard';

builtin

Perl v5.36 (experimental)**, Perl v5.40** (stable)

```
use builtin qw(true false is_bool inf nan weaken
               unweaken is_weak blessed refaddr
               reftype created_as_string
               created_as_number stringify
               ceil floor indexed trim is_tainted
               export_lexically load_module);
```

builtin

Perl v5.36 (experimental)**, Perl v5.40** (stable)

```
use builtin qw(true false is_bool inf nan weaken
               unweaken is_weak blessed refaddr
               reftype created_as_string
               created_as_number stringify
               ceil floor indexed trim is_tainted
               export_lexically load_module);
```

Это уже было в ~~Симпсонах~~ Scalar::Util

builtin

Perl v5.36 (experimental)**, Perl v5.40** (stable)

```
use builtin qw(true false is_bool inf nan weaken
               unweaken is_weak blessed refaddr
                                          tring
                                          ringify
                                          rim is_tainted
                                          d_module);
```

```
use v5.40;

my $is_enabled = true;
ceil(10.45); # 11
trim(' Ivan'); # 'Ivan'
```

multivariable foreach

Perl v5.36 (experimental)**, Perl v5.40** (stable)

```
my @list = qw/John 500 Ivan 3000/;

foreach my ($name, $income) (@list) {
    say "Hey $name, your income is $income";
}
```

multivariable foreach

Perl v5.36 (experimental)**, Perl v5.40** (stable)

```
my %hash = (John => 500, Ivan => 3000);

foreach my ($name, $income) (%hash) {
    say "Hey $name, your income is $income";
}
```

infix operators (pluggable operators)

Perl v5.38

Подключаемые ключевые слова (pluggable keywords) с v5.14

Подключаемые операторы (infix pluggable operators) с v5.38

XS::Parse::Infix

infix operators (pluggable operators)

Perl v5.38

```
use v5.38;
use Syntax::Operator::Equ;

if ($x equ $y) {
    say "x и y undef или являются одинаковыми строками";
}

if ($i === $j) {
    say "i и j undef или являются одинаковыми числами";
}
```

class

Perl v5.40

```
use feature 'class';
```

Corinna
https://github.com/Perl-Apollo/Corinna/tree/master/rfc

Object::Pad

https://perldoc.perl.org/perlclass

class

```
use feature 'class';

class Employee v3.14.1 :isa(Person::Base) {
    field $name :param //= "Unknown";
    field $office :reader = true;
    field $uuid;

    ADJUST {
        $uuid = UUID::generate();
    }

    method welcome_to($department //= "Sales") {
        say "Dear $name, welcome to $department";
        say "Your unique user ID: $uuid";
    }

}
```

class

```
use feature 'class';

class Employee v3.14.1 :isa(Per
    field $name :param //= "Unk
    field $office :reader = tru
    field $uuid;

    ADJUST {
        $uuid = UUID::generate();
    }

    method welcome_to($department //= "Sales") {
        say "Dear $name, welcome to $department";
        say "Your unique user ID: $uuid";
    }

}
```

```
my $empl = Employee->new(name => 'Kenny');
$empl->office;     # true
$empl->welcome_to("Support Team");

# Dear Kenny, welcome to Support Team
# Your unique user ID: 3556-7777
```

class

```
use feature 'class';

class Employee v3.14.1 :isa(Per
    field $name :param //= "Unk
    field $office :reader = tru
    field $uuid;

    ADJUST {
        $uuid = UUID::generate();
    }

    method welcome_to($departme
        say "Dear $name, welcom
        say "Your unique user ID: $uuid";
    }

}
```

```
my $empl = Employee->new(name => 'Kenny');
$empl->office;     # true
$empl->welcome_to("Support Team");

# Dear Kenny, welcome to Support Team
# Your unique user ID: 3556-7777
```

```
say reftype $empl; # OBJECT
```

Связанные сравнения (chained comparisons)

Perl v5.32

```
if ($x < $y <= $z) { ... }
```

```
if ($x < $y && $y <= $z) { ... }
```

bitwise, logical xor,..

- bitwise: &. |. ^. ~. (для строк) **v5.28** (stable)

bitwise, logical xor,..

- bitwise: &. |. ^. ~. (для строк) **v5.28** (stable)
- logical xor: ^^ **v5.40**

bitwise, logical xor,..

- bitwise: &. |. ^. ~. (для строк) **v5.28** (stable)
- logical xor: ^^ **v5.40**
- %{^HOOK} хук для require (require__before и require__after) **v5.40**

Улучшение производительности

Улучшение производительности

Perl Version:	5.12	5.16	5.20	5.24	5.28	5.32	5.36
Astro	100%	122%	126%	157%	160%	160%	167%
BioPerl Codons	100%	102%	166%	170%	167%	182%	171%
BioPerl Mono	100%	95%	124%	121%	119%	127%	126%
CSS::Inliner	100%	105%	110%	120%	128%	126%	124%
DateTime	100%	102%	104%	116%	119%	118%	118%
Digest	100%	100%	100%	100%	100%	99%	100%
HTML::FormatT	100%	102%	106%	119%	124%	123%	122%
Math::DCT	100%	148%	149%	148%	245%	233%	243%
Moose	100%	98%	106%	126%	129%	127%	130%
Primes	100%	105%	106%	118%	118%	116%	116%
Regex/Subst	100%	111%	122%	131%	138%	178%	166%
Reg/Subst utf8	100%	104%	134%	149%	132%	136%	138%
Test Moose	100%	95%	96%	99%	99%	93%	95%
Total	100%	105%	118%	127%	130%	131%	132%

Удаленные возможности

Perl v5.30

- array_base: $[= 1

```
my @list = qw/first second/;
$[ = 1;
say $list[1]; # prints "first"
```

Удаленные возможности

Perl v5.42

- Smartmatch: (~~ и given/when)

```
given ($operation) {
    when ($_ ~~ 'send') { ...; }
    when ($_ ~~ 'recv') { ...; }
    defaut {
        ...;
    }
}
```

Удаленные возможности

Perl v5.42

- Smartmatch: (~~ и given/when)

  Альтернативы:
  Syntax::Infix::Smartmatch
  Switch::Back
  Syntax::Operator::Matches
  Syntax::Keyword::Match

Удаленные возможности

Perl v5.42

- Smartmatch: (~~ и given/when)

  Альтернативы:
  Syntax::Infix::Smartmatch
  Switch::Back
  Syntax::Operator::Matches
  Syntax::Keyword::Match

- ' как разделитель в имени модуля

```
use Foo'Bar;
```

Что в будущем?

- Именованные параметры сигнатур

```
sub move_to($smth, :$ltt, :$lng) {
    say "Moving $smth to $ltt, $lng";
    ...;
}

move_to("box", ltt => 30.11, lng => 55.95);
```

Что в будущем?

- Именованные параметры сигнатур
- Роли для классов

```
use feature 'class';

role Role::Serializable::JSON {
    ...;
}

class Cache::LRU :does(Role::Serializable::JSON) {
    ...;
}
```

Что в будущем?

- Именованные параметры сигнатур
- Роли для классов
- :writer для field

```
class Cat {
    field $cleaned :writer;
    ...
}

my $kitty = Cat->new(...);
$kitty->cleaned(true);
```

Что в будущем?

- Именованные параметры сигнатур
- Роли для классов
- :writer для field
- Поддержка meta-программирования

```
my $ver = do {
    no strict 'refs';
    ${"${pkg}::VERSION"};
}
```

```
my $ver = meta::package->get($pkg)
    ->get_symbol('$VERSION')
    ->value;
```

Что в будущем?

- Именованные параметры сигнатур
- Роли для классов
- :writer для field
- Поддержка meta-пр
- match/case

```
use Syntax::Keyword::Match;

match($n : ==) {
    case(1) { say "It's one" }
    case(2) { say "It's two" }
    default { say "It's smth else" }
}
```

Что в будущем?

- Именованные параметры сигнатур
- Роли для классов
- :writer для field
- Поддержка meta-программирования
- match/case
- equ и ===

Что в будущем?

- Именованные параметры сигнатур
- Роли для классов
- :writer для field
- Поддержка meta-программирования
- match/case
- equ и ===
- qt()

```
qt"Greetings, {$user->title} {$user->name}";
```

Что в будущем?

- Именованные параметры сигнатур
- Роли для классов
- :writer для field
- Поддержка meta-программирования
- match/case
- equ и ===
- qt()
- Perl 5 → Perl 7

```
use v7;
```

Спасибо!

perl-conf.ru/24