

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ имени Н.Э.БАУМАНА  
(национальный исследовательский университет)»

Факультет: Информатика и системы управления

Кафедра: Теоретическая информатика и компьютерные технологии

**Рубежный контроль №3**  
Конспект по языку **Raku**

Выполнил  
студент группы ИУ9-11Б  
Старовойтов Александр

Москва, 2021

# 1 Типы

Последовательная типизация:

- `my $var`; — динамическая, неявная
- `my Str $var`; — статическая, явная

`$var.WHAT`; — вернет тип переменной

Слабая типизация:

```
my $var = "2";
say $var.WHAT; # (Str)
$var *= 2;
say $var; # 4
say $var.WHAT; # (Int)
```

## 2 Управляющие конструкции

- `if <condition> { ... }` — выполняет блок в {}, если `<condition>.Bool == True`
- `else / elsif` — альтернативные ветки для `if`
- `unless <condition> { ... }` — выполняет блок в {}, если `<condition>.Bool == False`
- `with <condition> { ... }` — выполняет блок в {}, если `<condition>` определено
- `for @array -> $item { ... }` — выполняет блок {} для каждого элемента @array
- `given $var {`  
    `when 0..50 { say '<= 50' }`  
    `default { say 'default' }`  
}

аналог `switch`

- `loop (my $i = 0; $i < 5; $i++) { ... }` — `for` в `си-стиле`

## 3 Функциональный стиль

Иммутабельность — префикс `constant`:  
`my constant $var = 5`;

Функции — объекты первого класса:

```
my $f = -> $n { $n + 3 };
sub g($f, $n) {
    $f($n);
}
say g($f, 7) # 10
```

Встроенные функции высших порядков для работы с последовательностями:

- `map`  

```
my @array = <1 2 3>;
sub squared($x) {
    $x ** 2
}
say map(&squared, @array); # (1 4 9)
```
- `reduce`  

```
sub f($a, $b) {
    $a + $b;
}
say reduce(&f, (1, 2, 3)); # 6
```

## 4 IO, строки, регулярные выражения

### 4.1 IO

`$*IN`, `$*OUT`, `$*ERR` — хендлеры для `stdin`, `stdout`, `stderr`

- `say` — вывести в `stdout` с переносом строки
- `print` — вывести в `stdout` без переноса строки
- `note` — вывод в `stderr`

Пример перенаправления `stdout`:

```
my $output-file = open "file.txt", :w;
$*OUT = $output-file;
```

### 4.2 строки

- `indices` — поиск подстроки в строке
- `substr` — взятие подстроки
- `contain` — содержит ли подстроку строка

### 4.3 регулярные выражения

Оператор `~~` — матчит регулярное выражение с строкой:

```
'enlightenment' ~~ m/ light /
```