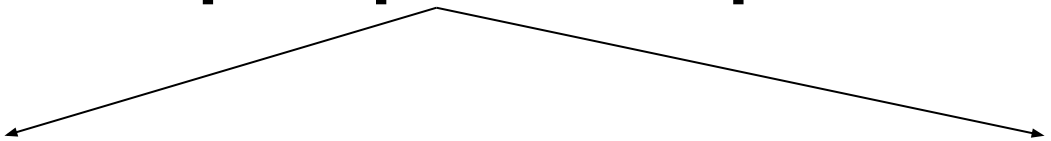
Скриптовые языки на примере Perl

Языки программирования



Скриптовые

- Программа (или ее бай-код) интерпретируется
- Зачастую более высокий уровень абстрагирования от системы
- Минимум риска «завалить» всю систему
- Кроссплатформенность

Компилируемые

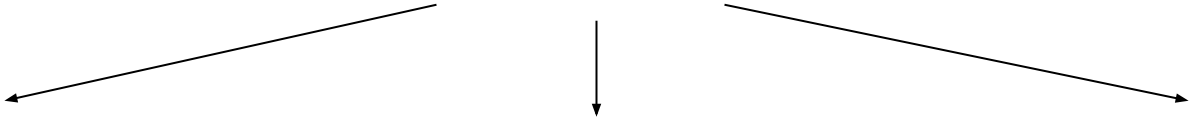
- Программа транслируется в машинные коды
- Быстродействие
- Более полный доступ к ресурсам системы

Perl – предварительно компилируемый (скриптовый) язык программирования высокого уровня. Основной особенностью языка считаются его богатые возможности по работе с текстом.

```
#!/usr/bin/perl -w

print "Hello, world!";
```

Типы данных



Скаляр

$index

- числа, строки и ссылки
- тип определяется динамически

Массив

@array

- упорядоченный список скаляров
- обращение к элементу: $array[10]

Хэш-таблица

%hash

- Ключ => скаляр
- обращение к элементу: $hash{ “key” }
- keys(%hash) – массив ключей

Операции и выражения

Аналогичные используемым в языке C

+, -, *, /, %

++, --

==, !=, <, >, <=, >=

&&, ||, !

&, |, ^, ~, <<, >>

=, +=, -=, *=, /=, %=, &=, |=, ^=, <<=, >>=, &&=, ||=

,

x ? y : z

[]

Уникальные

******

. (конкатенация)

x (повторение)

eq, ne, lt, gt, le, ge, cmp

or, and, not, xor

.=, x=

\ (ссылка), ->

=~, !~, m//, s///, tr///

print

`команда`

<>

..

=>

{}

Операторы

Условные

If и unless (модификатор)

If {…} else {…}

If {…} elsif {…} … else {…}

Цикловые

while и until

foreach

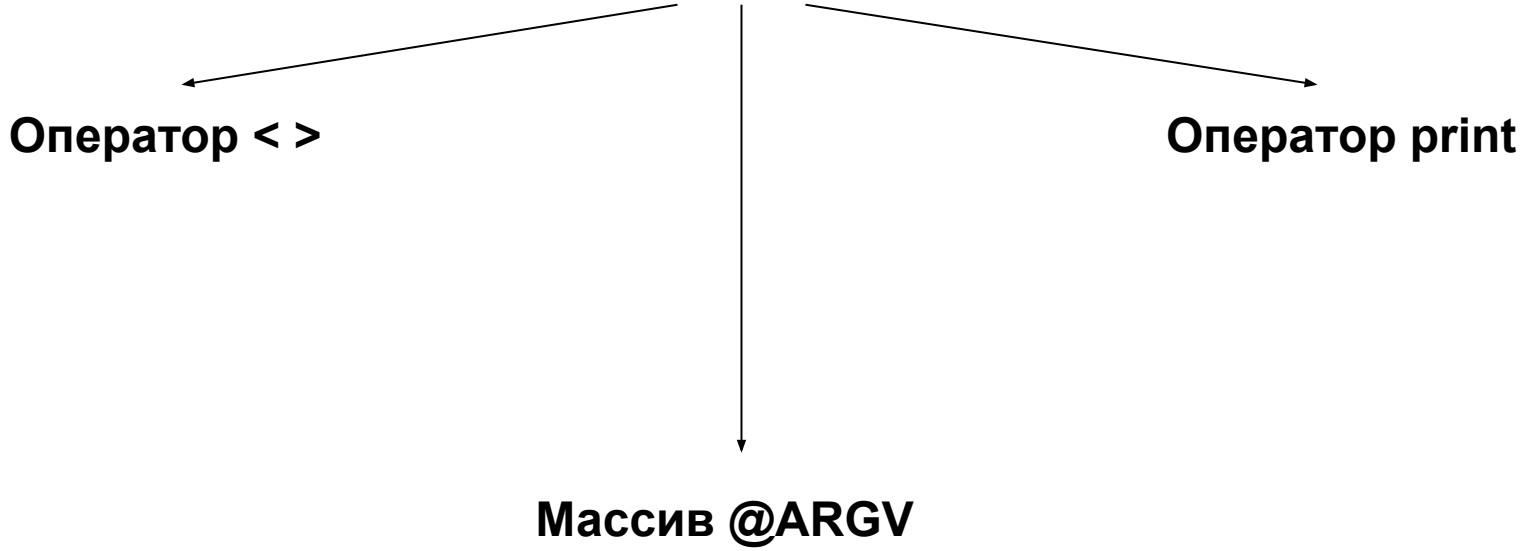
for

last

next

redo

Ввод-вывод



Простейшая работа с файлами

Дескриптор – это символическое имя, которое используется для представления файла

Открытие файлов: **open** дескриптор, имя_файла_с_префиксом

Префиксы: < (чтение, по умолчанию), > (запись), >> (добавление)

Возвращает 0 в случае успешного выполнения.

Закрытие файлов: **close** дескриптор

Чтение из файлов: **<**дескриптор**>**

Запись в файлы: **print** дескриптор

Работа со строками

Регулярные выражения

По сути это строки-шаблоны, состоящие из символов и метасимволов (символы, имеющие специальное значение) и задающие определенные правила поиска строк.

Метасимволы:

**** - экранирование или отрицание

. - одиночный символ

^ - начало строки

$ - конец строки

| - выбор

[ ] - класс символов

( ) - группировка

***** - нуль и более повторений

+ - одно и более повторений

? - нуль или одно повторение

{ } - явное количество повторений

Метапоследовательности:

\n – символ новой строки

\r – символ возврата каретки

\t – символ табуляции

\s – класс пробельных символ

\d – цифровые символы

\w – буквенно-цифровые символы

Подпрограммы и функции

Определение: **sub** имя [(параметры)] [{ тело }];

Вызов: имя (параметры);

имя параметры;

Определение локальных переменных: функции **my** и **local**

Возврат значений: функция **return**

Передача параметров: массив **@_**

Практическое задание

1. Шаблонный парсинг входной последовательности строк:
   - bad -> good
   - выходим, если встретили слово “exit”

2. Объединение данных из двух файлов в один

Файл1:

Anna 17
Pavel 20
Lena 40
Dima 25
Ivan 90

Файл2:

Inna 000-00-00
Ivan 111-11-11
Marina 444-44-33
Dima 345-32-55

Нужный результат:

Name	Age	Phone
Inna	-	000-00-00
Pavel	20	-
Marina	-	444-44-33
Dima	25	345-32-55
Ivan	90	111-11-11
Anna	17	-
Lena	40	-

Решение Задачи 1

```
while( <>)
{
    exit() if /exit/;
    $_ =~ s/good/bad/;
    print $_;
}
```

Решение Задачи 2 (часть 1)

```
open FILE1, "f1";
open FILE2, "f2";

@file1 = <FILE1>;
$file1_size = $#file1;
@file2 = <FILE2>;
$file2_size = $#file2;

for( $i = 0; $i < $file1_size + 1; $i++)
{
    if( $file1[$i] =~ /(\w+)\s+(\d+)/ )
    {
        %{$name_hash{$1}} = ("age" => $2, "phone" => "-");
    }
}
```

Решение Задачи 2 (часть 2)

```
for( $i = 0; $i< $file2_size + 1; $i++)
{
    if( $file2[$i] =~ /(\w+)\s+([0-9-]+)/ )
    {
        if( defined $name_hash{$1} )
        {
            $age = ${$name_hash{$1}}{"age"};
        } else {
           $age = "-";
        }
        %{$name_hash{$1}} = ("age" => $age, "phone" => $2);
    }
}
```

Решение Задачи 2 (часть 3)

```
print "Name".(" " x 4)."Age".(" " x 4)."Phone\n";


 foreach (keys( %name_hash))
{
    printf("%-8s", $_);
    printf("%-7s%s\n", ${$name_hash{$_}}{"age"},
      ${$name_hash{$_}}{"phone"});
}
```

Спасибо за внимание.
Вопросы?