

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Хакасский государственный университет им. Н.Ф. Катанова» Институт
информационных технологий и инженерного образования Кафедра
Программного обеспечения вычислительной техники и автоматизированных
систем.

РЕФЕРАТ

по дисциплине: «Курс информатики на английском языке»

Выполнил:
Студент группы 36-1
Сибирев Пётр Романович

Особенности Perl

Общая структура Perl в общих чертах ведёт своё начало от языка Си. Perl — процедурный по своей природе, имеет переменные, выражения присваивания, блоки кода, отделяемые фигурными скобками, управляющие структуры и функции.

Perl также заимствует ряд свойств из языков программирования командных оболочек UNIX. Все переменные маркируются ведущими знаками, которые точно выражают тип данных переменной в этом контексте (например, скаляр, массив, хеш). Важно, что эти знаки позволяют переменным быть интерполированным в строках. Perl обладает множеством встроенных функций, которые обеспечивают инструментарий, часто используемый для программирования оболочки, например сортировку или вызов системных служб.

Perl заимствует массивы из Лиспа, регулярные выражения из AWK и sed, из AWK также позаимствованы хеши («ассоциативные массивы»). Регулярные выражения облегчают выполнение многих задач по парсингу, обработке текста и манипуляций с данными.

Perl 5 добавил поддержку сложных типов данных, первоклассных функций (замыкание как значение) и объектную модель. В последнюю входят ссылки, пакеты, выполнение методов от класса, переменные с лексическим объявлением области видимости, а также директивы компилятора (например, `strict`). Главнейшим усовершенствованием, представленным в Perl 5, стала возможность помещать код в «пакеты» (`package`) в качестве модулей для повторного использования. Ларри Уолл позже заметил, что «Весь замысел модульной системы Perl 5 сводился к поощрению роста культуры Perl, а не строчек кода».

Все версии Perl выполняют автоматическую типизацию данных и автоматический контроль над памятью. Интерпретатор знает тип и запросы памяти каждого объекта программы, он распределяет и освобождает память, производя подсчёт ссылок. Перевод одного типа данных в другой — например, числа в строку — происходит автоматически во время исполнения, невозможные для выполнения переводы типов данных приводят к фатальной ошибке.

Выражения и операторы Perl

Листинг кода:

```
#!/usr/bin/perl
use warnings;
$a = 3;
$b = 5;
$c = 8;
die "Старший и средний коэффициенты не должны одновременно быть нулевыми"
# Вычисление одного корня, если старший коэффициент равен 0
$x1 = -$c/$b if $a == 0;
# Вычисление дискриминанта уравнения, если
# старший коэффициент не 0
$d = $b**2 - 4*$a*$c if $a != 0;
# Вычисление корней, если старший коэффициент
# не 0 и дискриминант положителен
($x1 = (-$b + sqrt $d)/$a/2, $x2 = (-$b - sqrt $d)/$a/2)
    unless $a == 0 || $d < 0;
# Печать результатов
print "Коэффициенты:\n a = $a b = $b c = $c";
print "\tРешение:\n\t$x1"
    if defined $x1 and not defined $x2;
print "\tРешение:\n\t$x1\t$x2"
    if defined $x1 and defined $x2;
print "\tРешения нет!" unless defined $x1;
```

Коэффициенты:

$$a = 3 \quad b = 5 \quad c = 8$$

Решение:

$$-1.6$$