

```

#!/perl
#####
## vim: set enc=cp1251 ts=3 sw=3 filetype=ada nobackup:
#####
## "THE BEER-WARE LICENSE" (Revision 42):
##
##     Alice Margatroid wrote this file.  As long as you retain this notice
##     you can do whatever you want with this stuff.  If we meet some day,
##     and you think this stuff is worth it, you can buy me a beer in return.
##
##     Alice.
#####

#####
## Задача: Произвести поиск всех файлов изображений, имеющих дубли. Вывести
##         отчет о найденных.
## Идея: Каждый уникален ровно до тех пор, пока не найден второй с такой же
##         подписью.
## Данные:
##     пусть $dir - точка отсчёта и единственный аргумент программы;
##     пусть %fls - список всех найденных файлов
## Формат:
##     %fls+-{Сигнатура}-+-{Полное имя файла}--{Ширина,
##         |                               |                               Высота,
##         |                               |                               Количество цветов}
##     |                               +-{Полное имя файла}--{Ширина,
##         |                               |                               Высота,
##         |                               |                               Количество цветов}
##     |                               (e.t.c.)
##     +-{Сигнатура}-+-{Полное имя файла}--{Ширина,
##         |                               |                               Высота,
##         |                               |                               Количество цветов}
##     |                               +-{Полное имя файла}--{Ширина,
##         |                               |                               Высота,
##         |                               |                               Количество цветов}
##     |                               (e.t.c.)
##     (e.t.c.)
##
## Алгоритм:
##     В $dir проводим рекурсивный поиск файлов изображений.
##     Для каждого найденного файла:
##     - Вычислим его подпись;
##     - Добавим запись в список найденных файлов;
##     - Ищем дальше.
##     Проверим количество записей в списке на ноль:
##     - Ноль значит, что дубли не найдены;
##     - Кончаем.
##     Выводим список найденных дублей.
##     Кончаем.
#####

use strict;
use warnings FATAL => 'all';
use Cwd 'realpath';
use File::Basename;

my $name = basename $0;

#####
## usage
##
## Назначение:
##     Печатает переданную информацию и краткую информацию об использовании.
## Исключения:
##     Отсутствуют.
## Побочные эффекты:
##     Завершает выполнение программы с ошибкой.
sub usage (;@) {

```

```

    if( scalar @_ ) {
        print STDERR "$name: $_\n" foreach(@_);
    }
    print STDERR "usage: $name <directory_with_images>\n";
    exit 1;
}

my $dir; eval { $dir = realpath shift @ARGV };
scalar @ARGV and usage; -d $dir or usage;

my %fls = ();

use File::Find; use vars qw/*fnm*/; *fnm = *File::Find::name;
use File::Type; my $ft = File::Type->new();

#####
##  wnt
##
##  Назначение:
##    Подпрограмма-обработчик; вызывается модулем File::Find для каждого
##    найденного файла.
##  Исключения:
##    Отсутствуют.
##  Побочные эффекты:
##    - Вызывает стороннюю программу identify из пакета ImageMagik.
##    - Добавляет записи в глобальный хэш %fls.
sub wnt {
    ( -f $_ and $ft->checktype_filename($_) =~ m/image/i ) or return;
    my $dat = qx(identify -quiet -format "%w|h|z|%" "$fnm") or return;
    chomp $dat;
    my( $w, $h, $d, $s ) = split '\\|', $dat;
    $fls{$s}{$fnm} = [ $w, $h, $d ];
    return;
}
File::Find::find( { wanted => \&wnt }, $dir );

foreach my $s (keys %fls) {
    ( my $n = scalar keys %{$fls{$s}} ) > 1 or next;

    print
        '-\x80, "\n",
        "There are $n doubles for signature [$s]\n";

    foreach my $f (sort keys %{$fls{$s}}) {
        print
            '\x4, "File: $f\n",
            '\x8, $fls{$s}{$f}->[0], 'x',
            $fls{$s}{$f}->[1], '@',
            $fls{$s}{$f}->[2], "\n";
    }
}

exit 0;

__END__

```