

Perl语言高级编程专题

Lesson 11

周晓方

courses@xfzhou.homeftp.org

Perl和网络， Perl和Win32

MAC地址, IP地址, 端口号

- MAC地址:
 - 网卡绑定的唯一48bit地址, 六段16进制表示
- IPv4地址:
 - 32bit网际网地址, 表示成四段十进制数10.10.10.1
 - 127.x.x.x是localhost, 表示本地, 用于开发调试
 - 10.x.x.x, 172.16-32.x.x, 192.168.x.x等用于内网
 - 子网的第一个IP用于路由, 最后一个IP用于广播
 - 互联网上IP地址和域名之间有对应关系(DNS)
- 端口号:
 - 1~65535的一个数字, 常用协议有固定的端口号(1~1023)
 - http 80, ftp 21, https 443, smtp 25, pop3 110, etc
- 每台PC可以装多个网卡, 每个网卡对应一个MAC地址, 但可绑定多个IP地址, 每个IP可以有多个域名

网络常用工具(以Windows为例)

- ping 域名或IP 了解远程机器的相应情况
 - nslookup 域名或IP 检测DNS信息
 - tracert 域名或IP 列出到达指定IP的路径
 - netstat -p tcp 列出TCP连接的端口号和情况
 - ipconfig (/all) 列出本机网卡的MAC/IP地址等
-

以下这些是微软网络的命令

- net share 查看本机共享资源
- net view /domain 查看域、工作组名
- net view \\pc_name 查看其他机器的共享资源

ftp(File Transfer Protocol)入门

- 命令行ftp的常用命令

- open 域名或IP [端口号] 并输入用户和口令
- ls或dir 列出远程目录的文件列表
- cd/pwd/mkdir 改变/显示/创建远程目录
- lcd 改变本地目录
- ! 进入shell
- bin/ascii 传输模式
- get/mget *name* 下载单个文件/一组文件
- put/mput *name* 上传单个文件/一组文件
- prompt 是否逐一提示(传多个文件时)
- hash 传输文件时是否提示进度
- close 关闭连接
- quit 退出

Freeware win32 ftp/http
Server: xitami/xipro

Perl脚本连接ftp站点

- 面向对象格式，完全摹仿命令行方式

- `$ftp = Net::FTP->new($host [, %options,例如port=>21等])`
- `$success = $ftp->login($user, $password)`
- `$ftp->binary/ascii / hash / pwd / ls / dir / cwd(dir) / delete(file)...`
- `$ftp->get(远端文件名 [, 本地文件名 [, 起点]])` 可以断点续传
- `$ftp->put(本地文件名 [, 远端文件名])`上传文件
- 打开句柄读远端文件
`$fh = $ftp->retr(文件)`
- 打开句柄写远端文件
`$fh=$ftp->stor(文件)`
`$fh=$ftp->appe(文件)`
- 远端文件的修改时间
`$time=$ftp->mdtm(file)`
- 得到远端文件大小
`$size=$ftp->size(文件)`

```
use Net::FTP; #列目录并下载打印index.txt文件
use constant HOST => 'localhost';
use constant USER => 'anonymous';
#connect, login, and list directory
my $ftp = Net::FTP->new(HOST) or die $@;
$ftp->login(USER) or die $@;
print $ftp->pwd, "\n";
my @dir = $ftp->dir;
print join("\n", @dir), "\n";
#download a file and dump to STDOUT
my $fn = 'index.txt';
my $fh = $ftp->retr($fn);
warn "Can't find $fn" if not $fh;
print <$fh> if $fh;
$ftp->quit;
```

- (续上页)断点续传下载的一种写法:
\$offset = (stat(\$file))[7] || 0; #如果文件大小>0则续传
\$ftp → get(\$file, \$file, \$offset);
- 如何透过防火墙/代理访问ftp:参考Net::Config

Net::FTP的父对象——Net::Cmd

- command(\$command)方法发送命令
ok方法返回成功与否
response方法返3位结果码的第一位
code方法返回3位数值型结果码
message方法返回文本消息
结果码在400以上的表示错误信息，例如正常login
一个ftp服务器后会有类似这样的反馈:

331 Anonymous access ok: give e-mail address as password

Password:*****

230 User anonymous logged-in

发邮件的协议SMTP:25

- 下面是一次SMTP(Simple Mail Transfer Protocol) 对话:

连接SMTP服务器	C:>telnet mailserver.com 25	220连接成功
	220 mailserver.com Welcome to RaidenMAILD SMTP service v1903, Wed, 26 May 2004 13:21:40 +0800, (C)2001-2003	
打招呼,告知域名	HELO my_domain.com	250好的
告知发件人邮箱	250 my_domain.com mail from: userA@mailserver.com	
告知收件人邮箱	250 (EMPTY) Sender OK rcpt to: <userB@mailserver.com>	250邮箱正确
开始写信	250 userB@mailserver.com recipient verified data	250邮箱正确
<u>信的内容</u>	354 start mail input; end with <CRLF>.<CRLF> <u>hello!</u>	354写信提示
<u>.表示信结束</u>	. . 250 mail received, ready to proceed.	250信收到
礼貌地退出	quit	
	221 RaidenMAILD SMTP service signing off	221服务结束
C:>_	失去了跟主机的连接。 C:>_	

收邮件的协议POP3:110

- 下面是一次POP3(Post Office Protocol 3)对话:

连接POP3服务器	C:>telnet mailserver.com 110 <i>+OK Welcome to RaidenMAILD POP3 service v1903, Wed, 26 May 2004 13:51:56 +0800,(C)2001-2003</i>	<i>+OK</i> 连接成功
登录 用户名	user USER2 <i>+OK USER USER2 OK</i>	<i>+OK</i> 有这个用户
登录 口令	pass **** <i>+OK Password OK, enter mailbox now.</i>	<i>+OK</i> 登录成功
列出所有邮件	list <i>+OK 1 112 1 112</i>	<i>+OK</i> 有1封信 编号、长度
看第1封信	. retr 1 <i>+OK 112 octets Received: from local([192.168.11.111]) by RaidenMAILD([10.11.11.1]); Wed, 26 May 2004 13:24:58 +0800 hello!.</i>	来源和日期 信的内容
删除第1封信	. dele 1 <i>+OK mail 1 deleted</i>	<i>+OK</i> 删除成功
礼貌地退出	quit <i>+OK RaidenMAILD POP3 service signing off</i>	<i>+OK</i> 结束服务
C:>_	失去了跟主机的连接。 C:>_	

用Perl发信 (no spam, 严禁发送垃圾信件)

• Net::SMTP的例子

Freeware win32 smtp/pop3 Server:

hMailServer, etc. Domain must be a CName.

Editing host file if necessary.

C:/Windows/System32/drivers/etc/hosts

```
#!/usr/bin/perl -w
use strict;
use Net::SMTP;
use constant HOST => '你的SMTP邮箱服务器地址';
use constant USER => '你的用户名';
use constant SENDER => '发信人@域名';
use constant RECIPIENT => '收信人@域名';

my $smtp = Net::SMTP->new(HOST);
#-----SMTP需要身份认证?-----认证功能需要安装和使用Authen::SASL模块
print "Authenticate Password for " . USER . ":";
my $password = <STDIN>;
chomp $password;
$smtp->auth(USER, $password);
#-----开始发信-----
$smtp->mail(SENDER);
$smtp->recipient(RECIPIENT); # 可以同时给多人($addr1, $addr2, $addr3...)
$smtp->data(<<"HERE");
hello
HERE
$smtp->quit;
```

用Perl发信(II)

- 身份认证需要安装Authen::SASL模块,见CPAN
\$smtp→auth('用户名','密码');
- 数据可以分多段发送,并发送信封:
\$smtp→data();
\$smtp→datasend(
 "From:发方\n",
 "To: 收方\n",
 "Subject: 主题subject\n";
\$smtp→datasend(@data); #可分多次调用datasend
\$smtp→dataend; #表示信体结束
- 可以在quit之前反复\$smtp→mail(...)发送多封信件
- 提示:发送带附件的信要用到MIME::Tools或者MIME::Lite等模块,还会遇到utf/gb编解码等事务

用Net::POP3收信

```
$pop3 = Net::POP3→new('服务器地址')      #连接
$message = $pop3→login(用户名,密码)      #登录

$last = $pop→last                          #消息编号是1~$last, 如果$last>=1
$pop→delete(消息编号)                    #给消息做删除标记,quit时才删除
$pop→reset                               #复位, 取消所有的删除标记
$hashref = $pop→list                      #返回散列引用, (消息编号→长度)
$size=$pop→list(消息编号) #返回指定消息的长度

$arrayref = $pop→get(消息编号)            #读取消息, 返回数组引用
$handle = $pop→getfh(消息编号)            #读取消息, 返回可读句柄
    print <$handle> .....

$pop→quit                                #退出
```

用Net::HTTP下载网页

- 下载三个网页(超文本, jpeg, 错误连接)并显示

```
use Net::HTTP;
my ($path, $http, $code, $mess, %h, $buf, $n);
$http = Net::HTTP->new(Host => "animal.vine.homeftp.org", KeepAlive => 1)
    or die $@;
foreach $path ("/", "/041/04131A.jpg", "/bad") {
    $http->write_request(GET => $path, 'User-Agent' => "Mozilla/5.0");
    ($code, $mess, %h) = $http->read_response_headers;
    print STDERR "$path\ncode : $code, mess : $mess, hash pairs :\n";
    print STDERR "$k = $v\n" while ($k, $v) = each %h ;
    print "-----\n";
    next if $code ne "200";
    binmode STDOUT;
    while (1) {
        $buf;
        $n = $http->read_entity_body($buf, 1024);
        die "read failed: $!" unless defined $n;
        last unless $n;
        print $buf;
    }
    print "\n===== \n\n";
}
```

运行结果

• 注意Code/mess/%h的返回值

连接超文本

```
/
code : 200, mess : Ok, hash pairs :
Content-Type = text/html
Connection = Keep-Alive
Last-Modified = Sat, 31 Jan 2004 13:47:34 GMT
Server = Xitami/Pro
Content-Length = 521
Keep-Alive = timeout=30, max=25
-----
<html>
<body>
<li><a name="04131">Jan, (以下省略, 共521字节)
=====
```

连接JPEG二进制图像

```
/041/04131A.jpg
code : 200, mess : Ok, hash pairs :
Content-Type = image/jpeg
Connection = Keep-Alive
Last-Modified = Sun, 08 Feb 2004 07:56:10 GMT
Server = Xitami/Pro
Content-Length = 12073
Keep-Alive = timeout=30, max=25
-----
??? JFIF(二进制内容, 以下省略, 共12073字节)
```

连接无效网页时

```
/bad
code : 404, mess : Not found,
hash pairs :
Content-Type = text/html
Server = Xitami/Pro
Content-Length = 117
-----
```

ActivePerl的Win32模块

- 使用ActivePerl,程序增加use Win32;
- 错误编号和错误字符串(Win32有时不用\$!)
 - Win32::GetLastError()
 - Win32::FormatMessage(Win32::GetLastError())
- 操作系统的版本信息
 - (\$serve_pack, \$major, \$minor, \$build, \$platform_id)
= Win32::GetOSVersion()
 - 其中platform_id, 0 = win32, 1=win9x, 2=winnt
 - 判断Win类型 Win32::IsWinNT(), Win32::IsWin95()
 - Perl的编号: ~~\$PerlBuild = Win32::BuildNumber();~~

BuildNumber() is Active Perl only

实例version.pl

```
use Win32;

my ($service_pack, $major, $minor, $build, $platform_id) =
    Win32::GetOSVersion();
my ($perlbuild) = Win32::BuildNumber();
my ($is95, $isnt) = (Win32::IsWin95(), Win32::IsWinNT());

print << "Here";
(省略,详见version.pl)
Here
1;
```

某XP上运行的结果

Win32::GetOSVersion() returns:

```
service_pack  Service Pack 1
major.minor   5.1
build         2600
platform_id   2
```

Win32::BuildNumber() returns:

635

Win32::IsWin95() returns 0

Win32::IsWinNT() returns 1

某98上运行的结果(安装的ActivePerl版本不同)

Win32::GetOSVersion() returns:

```
service_pack  A
major.minor   4.10
build         67766446
platform_id   1
```

Win32::BuildNumber() returns:

805

Win32::IsWin95() returns 1

Win32::IsWinNT() returns 0

另一些函数

- 用户名、计算机名、域名
 - Win32::LoginName(), Win32::NodeName(), Win32::DomainName()
- 设置和获取当前工作目录
 - Win32::SetCwd('路径'), Win32::GetCwd()
- 文件系统类型
 - (类型, 标志, 目录串长度)=Win32::FsType()
- 下一个可用驱动器
 - Win32::GetNextAvailDriver()
- 开机以来的时间（毫秒）
 - Win32::GetTickCount()

Windows的剪贴板

- Win32::Clipbrd;
- 目前支持3种格式，文本、图片和文件
- 清空和设置Empty(), Set('..')
- 例子程序监视剪贴板内容

```
use Win32::Clipboard;
my $clip = Win32::Clipboard();
while (1) {
    if ($clip->WaitForChange(10000)) {
        print "Clipboard changed. Format is ";
        if ($clip->IsText()) {
            print "text. ", $clip->GetText(), "\n";
        } elsif ($clip->IsFiles()) {
            print "files.\n", join("\n", $clip->GetFiles()), "\n";
        } elsif ($clip->IsBitmap()) {
            print "bitmap.\n";
            open B, ">b.bmp";
            binmode B;
            print B $clip->GetBitmap;
            close B;
        } else {print "not supported : ", join(", ", $clip->EnumFormats), "\n";}
    } else {
        print "Clipboard not changed, go on waiting...\n"
    }
}
```

多OS剪贴板通吃模块:

```
use Clipboard;
Clipboard->copy("blabla");
print Clipboard->paste;
```

其他一些Win32模块

- Win32::Console 控制Console窗口
- Win32::ChangeNotify 监视目录文件改变
- Win32::File 修改、获得文件属性
- Win32::FileSecurity NT文件权限控制
- Win32::Internet wininet.dll, http, ftp, 等
- Win32::NetResource 共享文件和打印机
- Win32::ODBC ODBC数据库接口
- Win32::Process Win32系统的进程管理函数
- Win32::Registry注册表
- Win32::Shortcut快捷方式
- Win32::IPC/Semaphore/Mutex多进程线程协调
-

Win32::Pipe 命名管道named pipe

- 带名字的管道,可以让同一个域上的进程交换数据,其中管道源必须是NT/2K/XP,不能是9X

```
use Win32;          #管道主机端代码, 先运行这段代码
use Win32::Pipe;
die "Must be NT to create a named pipe.\n" if not Win32::IsWinNT();
my $pipe = new Win32::Pipe("My Perl Pipe");
die "Fail to create pipe\n" if not $pipe;
die "Fail to connect pipe\n" if not $pipe->Connect;
my $msg = "I'm " . Win32::LoginName . " on " . Win32::NodeName . "\n" .
    "I'm running script $0.";
print "Send to Client:\n$msg\n";
die "Fail to send message\n" if not $pipe->Write($msg);
print "Read from Client:\n";
print $pipe->Read();    $pipe->Disconnect;    $pipe->Close;
```

```
use Win32;          #管道客户端代码, 后运行这段代码
use Win32::Pipe;

my $pipe = new Win32::Pipe("\\\\.\\pipe\\My Perl Pipe");
die "Fail to create pipe\n" if not $pipe;
my $msg = "I'm " . Win32::LoginName . " on " . Win32::NodeName . "\n" .
    "I'm running script $0.";
print "Send to Server:\n$msg\n";
die "Fail to send message\n" if not $pipe->Write($msg);
print "Read from Server:\n";
print $pipe->Read();    $pipe->Disconnect;    $pipe->Close;
```

Win32::Sound 播放声音

- Volume(音量), Play(文件,标志), Stop()

```
#!/usr/bin/perl -w
use strict;
use Win32;
use Win32::Sound;

Win32::Sound:: Volume ('100%'); #或Volumn(左,右)
Win32::Sound:: Play ("CLEAR.WAV", SND_ASYNC);
my($hz, $bits, $channels)
    = Win32::Sound:: Format ("CLEAR.WAV");
Win32::Sleep(500);
print "Clear.wav : ${hz}Hz, ${bits}B, ${channels}C\n";
sleep(1);
Win32::Sound:: Stop ();

1;
```

即时生成、播放波形

- 产生并播放一段正弦波形

```
use Win32::Sound;
$WAV = new Win32::Sound::WaveOut(44100, 8, 2);
$data = "";
$counter = 0;
$increment = 441/44100;
# Generate 44100 samples ( = 1 second)
for $i (1..44100) {
    # Calculate the pitch
    # (range 0..255 for 8 bits)
    $v = sin($counter/2*3.14) * 128 + 128;
    # "pack" it twice for left and right
    $data .= pack("cc", $v, $v);
    $counter += $increment;
}

$WAV->Load($data);      # get it
$WAV->Write();           # hear it
1 until $WAV->Status(); # wait for completion
$WAV->Save("sinus.wav"); # write to disk
$WAV->Unload();          # drop it
```

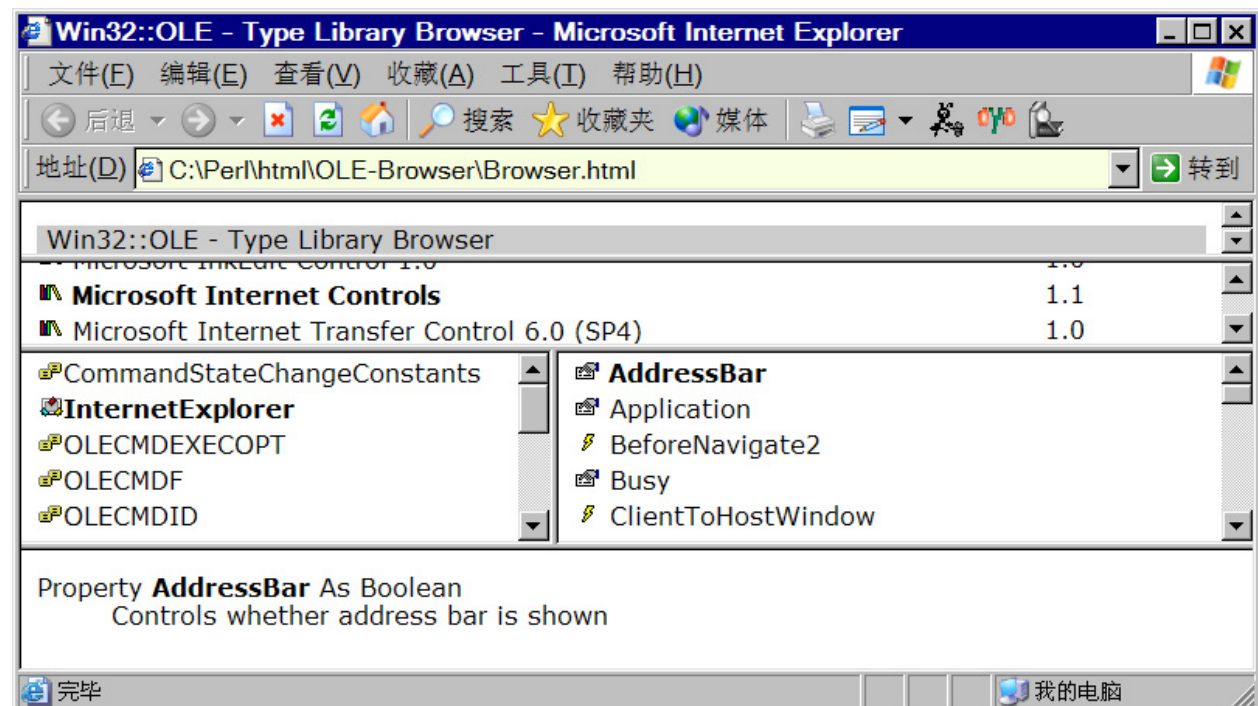
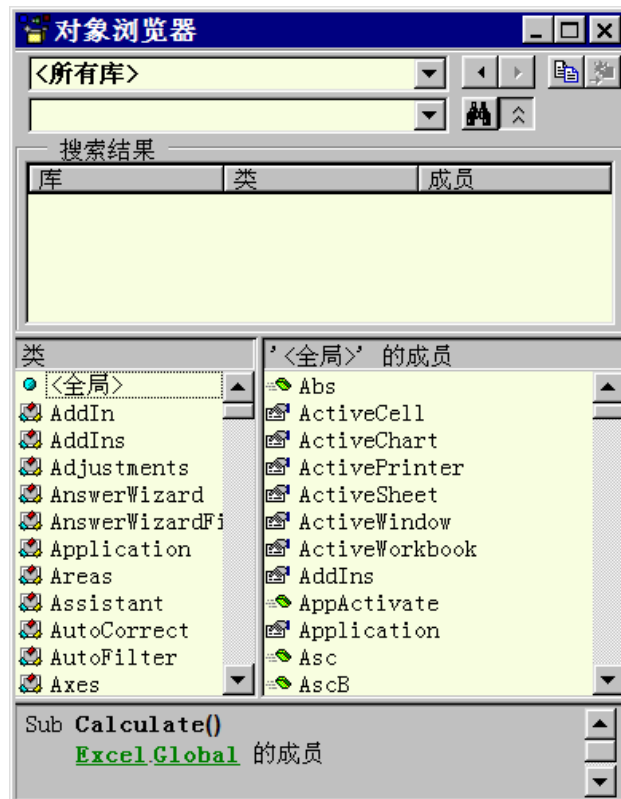
通过Win32::OLE打开IE

- Perl是Controller，启动/控制各种OLE Server
- 创建、调用方法、访问属性：
 - \$app = Win32::OLE->new('OLE Prog ID/Class ID', 'Destructor');
 - 远程启动时采用Win32::OLE-> new(['machine', '...ID'], 'Destructor');
 - OLE的出错信息不在\$!中，要用Win32::OLE->LastError()
 - \$result = \$app->Method(arguments);
 - \$result = \$app->{Property}; \$app->{Property} = exp;
- 参考ActivePerl帮助文档的OLE-Browser和VB手册

```
use Win32::OLE;
$ie = Win32::OLE->new('InternetExplorer.Application.1', 'Quit')
#或直接用CLSID new('{0002DF01-0000-0000-C000-0000000000046}',...
    || die "CreateObject: " . Win32::OLE->LastError;
$ie->{Visible} = 1;
$ie->Navigate("http://www.google.com/");
while ($ie->{Busy}) {
    print "Busy\t";
    sleep(1);
}
print "Done.\n"; #打开浏览器，连接Google主页，连接过程
sleep(2);        #中显示busy，完成后等2秒，最后退出
```

OLE-Browser和其他帮助

- Active Perl 帮助文件自带，要用到Perlscript功能
- 更详细的帮助，参考各个应用程序的帮助
- Office类应用程序有VB帮助和对象浏览器
 - 例如: Excel->Tool->宏->Visual Basic编辑器->[F2]



控制Winword的例子

- new()不指定Destructor, Perl结束后应用程序不退出
- 采用缺省参数时, 填写undef, 例如SaveAs的语法是:
 - *expression.SaveAs(FileName, FileFormat, LockComments, Password, AddToRecentFiles, WritePassword, ReadOnlyRecommended, EmbedTrueTypeFonts, SaveNativePictureFormat, SaveFormsData, SaveAsAOCELetter)* (参考Winword的VB帮助文件)
 - 在下面例子中用的是\$doc->SaveAs('ole.doc',undef,undef,"111")

- 要用好OLE, 就要熟悉各种OLE Server的编程接口, 以及COM/DCOM的知识, 仅仅学会Perl是远远不够的。

```
use Win32::OLE;                #本例用于Word2000
my $line = "Hello World!";    #不适合Word97
my $wd = Win32::OLE->new('Word.Application')
    || die "CreateObject: " . Win32::OLE->LastError;
$wd->{Visible} = 1;
my $doc = $wd->Documents->Add();
$wd->Selection->Font->{Size} = 100;
$wd->Selection->Font->{SmallCaps} = 100;
$wd->Selection->Font->{Animation} = 3;
$wd->Selection->TypeText($line);
$doc->SaveAs('ole.doc',undef,undef,"111");
```


将数据写入Excel文件

- 打开Excel的另一种办法

```
$Excel = Win32::OLE->GetActiveObject('Excel.Application')
```

- Excel对象层次

- Workbooks对象 `$wbs = $Excel->Workbooks`

- Workbook对象 `$wb = $wbs->Add()` 新建一个book
一个Workbook包含多个Worksheet和Chart对象
`$wb->SaveAs($name, ...)`

- Worksheet对象 `$sh = $wb->Worksheets(1)`

- Range对象 `$R = $Sh->Range("A1:D5")`
`$R->{Value} = [[第一行], [第二行], [第三行], ...]`
`$R->Font->{Bold} = 1` #选定范围单元格的字体属性

- 参考Excel的VB文档，参见实例程序excel.pl

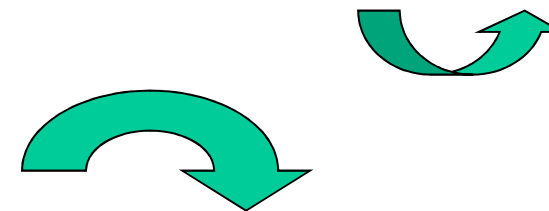
录制VBA宏→Perl OLE脚本

- 工具→宏→录制新宏→...各种操作...→停止
工具→宏→宏→编辑
- 找到关键的VB语句，转换为Perl OLE语句

```
Sub 宏1()
Range("A3:A5").Select
With Selection
    .HorizontalAlignment = xlGeneral
    .VerticalAlignment = xlBottom
    .WrapText = False
    .Orientation = 90
    .AddIndent = False
    .ShrinkToFit = False
    .MergeCells = False
End With
Columns("A:A").EntireColumn.AutoFit
End Sub
```

张三
李四
王五

三
四
五



```
$Range->{Orientation} = 90;
$Sheet->
    Columns("A:A")->
        {EntireColumn}->AutoFit();
```

从Excel文件读取数据

- 打开Excel的第三个方法：打开现有的文件

`$Book = Win32::OLE->GetObject(Win32::GetCwd . '\OLE2.xls','Quit')`

直接获得WorkBook对象

```
use Win32;
use Win32::OLE qw(in with);
use Win32::OLE::Const 'Microsoft Excel';
$Win32::OLE::Warn = 3; # die on errors...
my $Book = Win32::OLE->GetObject(Win32::GetCwd .
    '\OLE2.xls','Quit') or die "Can't open OLE2.xls";
my $Sheet = $Book->Worksheets('Perl_OLE');
my $data = $Sheet->Range("B3:E5")->{Value};
my ($row, $cell);
foreach $row (@$data) {
    foreach $cell (@$row) {
        print "$cell\t";
    }
    print "\n";
}
```

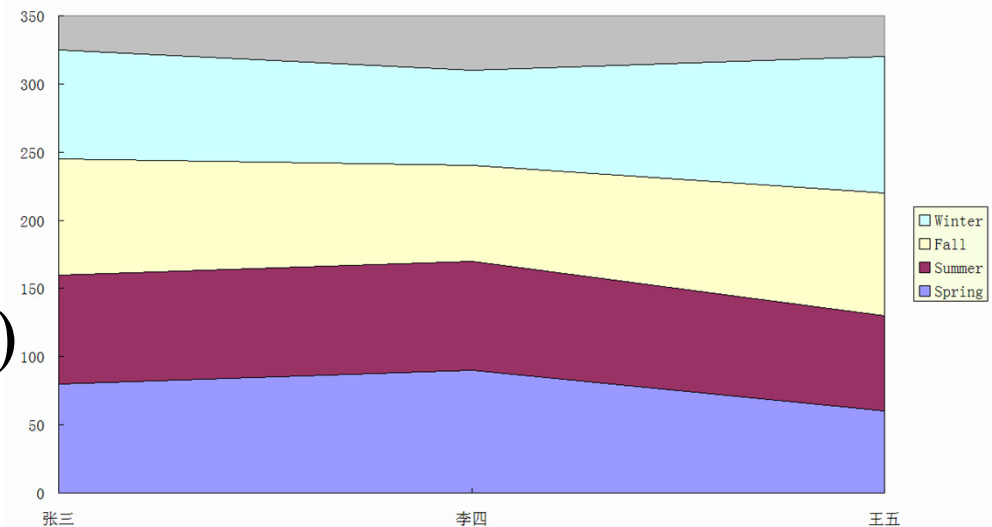
```
|| D:>perl excel3.pl ||
|| '80' '80' '85' '80' ||
|| '90' '80' '70' '70' ||
|| '60' '70' '90' '100' ||
|| D:> _ ||
```

无Office读取Excel文件:

```
use Spreadsheet::ParseExcel;
use Spreadsheet::ParseExcel::FmtUnicode;
#use Spreadsheet::ParseXLSX
use Spreadsheet::XLSX;
```

用Excel产生图表

- 引入Excel VB程序的各种常数
`use Win32::OLE::Const 'Microsoft Excel';`
- Chart对象的常用方法、属性:
 - {Name}, {ChartType}, {HasTitle}
 - SetSourceData({Source=>\$Range, PlotBy=>常数})
 这里用了传递参数给OLE方法的第二种形式.
 - ChartTitle->{Text}
- 将图表转换成图片
 - \$Chart->Export(文件名,图形过滤器,交互)
- 参考excel4.pl



回家作业9(See next page)

- ~~(Windows 32平台) Win32::IE::Mechanize模块, 在CPAN找到该模块, 阅读手册并下载安装~~
- ~~使用上述模块, 编写程序依次完成下列操作:~~
 - 打开IE浏览器
 - 打开主页<http://xfzhou.homedns.org/>
 - 点击courses, 点击今年的perl课件主页联接, 最后点击homework页面
 - 上面每个操作之间都停留1秒钟(sleep 1;)
 - 退出perl程序, 但不关闭IE浏览器
- ~~学号-09.pl~~
- ~~注: Win32::IE::Mechanize is out of date. See new homework No.9 on next page, plase.~~

Win32界面互动模块:

```
use Win32::GUI::TaskAutomate;  
use Win32::GUI::Robot;  
use Win32::GuiTest;  
use Prima;    # 图形存取
```

回家作业10

- 用Perl脚本生成一张幼小衔接的数学练习题：
 1. 随机产生40题100以内的正整数加减法题目，运算结果也在100以内。
 2. 结果文件格式请二选一：
 - a) 保存到 学号.pdf 文件中（用PDF::Create模块）
 - b) 保存到 学号.doc 文件中（用Win32::OLE模块）
 3. 排版和字体格式等适中即可，不必过度模仿样例
 4. 学号-10.pl

