دوره آموزشی زبان برنامهنویسی نرمافزار DIgSILENT، یا به اختصار 'DPL در روزهای ششم و هفتم اسفندماه سال ۱۳۸۵ در شرکت متن بینالملل واقع در پژوهشگاه نیرو برگزار گردید. در روز اول، ابتدا مقدماتی پیرامون دلایل استفاده، ویژگیها و ساختار این زبان برنامهنویسی مطرح شد. سپس، با چند مثال ساده، با پارامترهای محیط DPL و نحوه برنامهنویسی در آن آشنایی حاصل شد و در پایان نحوه دسترسی و استفاده از المانهای شبکه قدرت و توابع تحلیلی در محیط DPL بیان شد که در ادامه به تفصیل آمده است. در روز دوم، مباحث پایانی روز قبل کامل شده و نحوه مدیریت پارامترها و انجام مطالعات با بیان چند مثال، آموزش داده شد.

۱- مقدمه ای بر DIgsilent

هر پروزهای که در نرمافزار DIgSILENT ایجاد میشود، دارای سه پوشه در محیط Data Manager مرد مرد مرد افزار DIgSILENT (مرد مرد خواهد بود. Study Case که شامل توپولوژی شبکه و گرافیک مرتبط با آن خواهد بود. Study Case (مرد مطالعاتی) که همه تنظیمات محاسباتی و مطالعات انجام شده مانند contingency، اتصال کوتاه، مطالعاتی) که همه تنظیمات محاسباتی و مطالعات انجام شده مانند contingency، اتصال کوتاه، پخش بار و غیره در آن ذخیره میشوند و در صورت نیاز قابل مشاهده و تغییر برای مطالعات جدید خواهند بود همه نتایج حاصل از مطالعات، چه به صورت گسسته مانند انجام پخش بار در هر ساعت جدید خواهند بود همه نتایج حاصل از مطالعات، چه به صورت گسسته مانند انجام پخش بار در هر ساعت و چه به صورت پیوسته مانند انجام پخش بار در هر ساعت و چه به صورت پیوسته مانند مطالعات، چه به صورت گسسته مانند انجام پخش بار در هر ساعت در مواهند بود همه نتایج حاصل از مطالعات، چه به صورت گسسته مانند انجام پخش بار در هر ساعت در مواهند بود همه نتایج حاصل از مطالعات، چه به صورت گسسته مانند انجام پخش بار در هر ساعت در مواهند بود همه نتایج حاصل از مطالعات، چه به صورت گسسته مانند انجام پخش بار در هر ساعت و چه به صورت پیوسته مانند انجام پخش بار در هر ساعت در مواهند بود همه نتایج حاصل از مطالعات دینامیکی یا حالت گذرا در پوشه مورد مطالعاتی قابل در معرسی هستند. برای انتخاب پارامترهایی که مطلوب برای ذخیره سازی هستند و ایجاد تنظیمات در دره می توان از آیکن Edit Result Variable و نوار ایزار استفاده نمود. گزینه Graphic نیز مواه، می توان از آیکن Edit Result Variable و در نوار ایزار استفاده نمود. گزینه Graphic نیز مواه، می ورد مطالعاتی، حاوی نمودارها، نمودار تک خطی می باشد. Paste می باشد که به راحتی قابل ایجا به عنوان شبکه مینا برای آموزش انتخاب شده است، شبکه و مود می می ورد می با برای آموزش انتخاب شده است، شبکه می مود می با برای آموزش انتخاب شده است، شبکه و قابل ذکر است، شبکه که در اینجا به عنوان شبکه مینا برای آموزش انتخاب شده است، شبکه مینا برای آموزش انتخاب شده است، شبکه و قابل ذکر است، شبکه و که در اینجا به عنوان شبکه مینا برای آموزش انتخاب شده است، شبکه و که در اینما و که در مینه در اینما و که در اینما و که درما و که

Nine Bus System میباشد که به راحتی میتوان آن را از طریق Demo فعال و استفاده نمود و یـا در صورت تمایل به عدم تغییر در پروژه اصلی، آن را Copy و در User خود ذخیره کرد. (DPL) DIgSILENT - خصوصيات زبان برنامەنويسى

اصلی ترین دلیل استفاده از DPL اتوماسیون است. به خصوص می توان کارهای تکراری را خیلی ساده بطور خودکار انجام داد. عنوان مثال، اگر بخواهیم برای بار منحنی روزانه ایجاد کنیم، ناچار هستیم زمان را بطور دستی تغییر داده و هر بار پخش بار بگیریم، اما با استفاده از برنامه نویسی می توان در زمان کمتر و با صرف انرژی کمتری این کار را انجام دهیم که در ادامه این برنامه به عنوان تمرین نوشته خواهد شد و برای تعیین نقطه شکست ولتاژ باید آنقدر توان اکتیو و راکتیو را بطور دستی تغییر دهیم تا ولتاژ بشکند، اما با برنامهنویسی این عمل تکراری، به راحتی قابل انجام است. DPL یک زبان برنامهنویسی شی گرا^۲ است و می تواند به هر المانی به صورت یک شی دسترسی داشته باشد. همچنین داری دستورات اصلی و Dtjec ها و متدهای متنوعی می باشد که تعدادی از این متدها و دستورات در ضمیمه آورده شدهاند. دستورات JPL بسیار گسترده بوده و در صورت نیاز می توان دستور موردنظر را در Imm یافته و مورد استفاده قرار داد. این تیاز می توان دستور موردنظر را در Imm یافته و مورد استفاده قرار داد. JPC در داخل Study تعریف می شوند و دستورات آن نیز بسیار مشابه با زبان برنامهنویسی ++

نکته نیز خالی از لطف نیست که اکثر دستورات در این زبان در صورت درست نوشته شدن به رنگ آبی در خواهند آمد که می تواند نشان مناسبی برای اطمینان از درست نوشته شدن دستور در محیط Case Sensitive است.

ویژگیهای متعددی برای این زبان می توان برشمرد. بسیار کوتاه می توان به استفاده از دستورات ورودی-خروجی جهت ایجاد interface با کاربر، امکان انجام هر نوع دستور محاسباتی موجود در DIgSILENT، قابلیت اجرای DPL به عنوان زیربرنامه در برنامه دیگر، ایجاد پایگاه داده و نمودار اشاره کرد که در ادامه سعی خواهد شد به هر یک از این موارد بپردازیم.

Object Oriented - ^r

DPL -۳های موجود در کتابخانه

تعدادی DPL Commands از پیش نوشته شده و تحت عنوان DPL Commands در Library اصلی در پایگاه داده قابل دستیابی هستند. نحوه دستیابی به آن در شکل (۱-۱) قابل مشاهده میباشد. از طریق گزینه Execute DPL Script با شکل قبل نوشته شده را ببینید. هر برنامه جدیدی که توسط کاربر نوشته شود، علاوه بر اینکه در Study مورد نظر وجود خواهد داشت،به این لست نیز افزوده خواهد شد.



شکل(DPL :(۱-۱) های موجود در کتابخانه اصلی

این برنامه ها از نوع Read Only بوده و تنها می توان آنرا اجرا نمود و یا مقادیر پارامترهای ورودی آن را تغییر داد. این DPL ها به کتابخانه مرجع آدرس داده شده اند که در قسمت Remote Script آن محل ارجاعش مشخص است. تنها راه تغییر آن ایجاد یک نسخه جدید از آن در User خود با Copy کردن و اعمال تغییرات در نسخه جدید است.

۴- نحوهی ایجاد DPL

نحوه ایجاد یک DPL جدید، بدین ترتیب است که در Data Manager ودر مورد مطالعاتی مطلوب، با کلیک روی گزینه (۲) با نام New Object و انتخاب گزینه DPL Command and more و فشردن Ok صفحه شکل (۲-۱) باز خواهد شد. البته از طریق Copy و Paste کردن از روی DPL از قبل موجود نیز می توان DPL جدید ایجاد کرد. با دوبار کلیک روی هر DPL از قبل موجود، همین پنجره باز خواهد شد که از این طریق می توان پارامترهای آن را در صورت امکان تغییر داد.

۵- معرفی بخش های مختلف یک DPL

۵-۱- برگه Basic Option

در برگه Basic Option موارد زیر قابل مشاهدهاند که کاربرد هر یک را آوردهایم: Name : نام مورد نظر برای DPL در این قسمت وارد خواهد شد. Input Parameters : برای در نظر گرفتن ورودی برای DPL از ایس ستون استفاده می شود. به خصوص زمانیکه می خواهیم از DPL به عنوان یک زیر برنامه در برنامه دیگر استفاده کنیم و پارامترهایی را به عنوان ورودی آن استفاده کنیم. از ستون اول برای تعیین نوع متغیر، ستون دوم، نام و ستون سوم مقدار آن خواهد بود.

int : عدد صحيح

double : اعشارى

string : رشتهای از کاراکترها

object : یک شی set : مجموعهای از اشیا نحوه دسترسی به این متغیرها از طریق برنامههای دیگر در ادامه توضیح داده می شود. مثلا می توان در محاسبه پخش بار برای منحنی بار روزانه، استپهای زمانی جهت انجام پخش بار را به عنوان پارامتر ورودی در نظر گرفت.



شکل (DPL صفحه DPL) صفحه

External Object : از این قسمت جهت معرفی المانهای خارجی اعم از خط، ترانسفورماتور و دیگر عناصر شبکه و همینطور Object مربوط به مطالعات شبکه مانند پخش بار، اتصال کوتاه و بطور کلی همه المانهای موجود در Study Case استفاده می شود. از این طریق می توان از آن المانها در متن برنامه استفاده کرد و پارامترهای آن را تغییر داد. در مثالی کاربرد ایسن بخش آورده خواهد شد. در ستون اول نام مطلوب جهت استفاده در متن برنامه ودر ستون دوم، با دوبار کلیک و باز شدن صفحه Data Manager المان خارجی موردنظر انتخاب خواهد شد.

General Selection : در این قسمت می توان مجموعهای از المانها را از شبکه انتخاب و آن را به عنوان یک Set در برنامه استفاده نمود. برای اینکار ابتدا باید المانهای موردنظر را در صفحه اصلی انتخاب کرد و با کلیک راست روی آنها و انتخاب General Set حفحهای شامل المانهای Study Case و با کلیک راست روی آنها و انتخاب General Set حفحهای شامل المانهای Study Case در میشود که فشردن Ok موجب ایجاد المان General Set در لیست المانهای عنوان یا می گردد که از طریق General Selection در صفحه المان انتخابی است و برنامه برای ایس مجموعه المان انتخابی اجرا خواهد شد.این مجموعه انتخابی در متن برنامه تحت عنوان Study شناخته می شود. این مجموعه المان، برای همه DPL ها مقداری ثابت بوده و هر بار که تعریف شود، جایگزین مجموعه قبلی خواهد شد.

البته روش دیگری نیز جهت تعریف مجموعه المان در این قسمت وجود دارد که استفاده از گزینه DPL Command Set بعد از انتخاب عناصر مطلوب می باشد که موجب ایجاد CPL Set Command Set در صفحه پایگاه داده می گردد و مانند قبل قابل انتخاب و استفاده خواهد بود. در Set این حالت به صورت خودکار و پیش فرض DPL Command Set برای General Selection در نظر گرفته می شود.

Advanced Option برگه -۲-۵

www.powerengineering.blogfa.com

در برگه Advanced Option نیز دو گزینه قابل رویت است. یکی گزینه Remote Script که در آن می توان به DPL دیگری ارجاع داد که این عمل همه Setting موجود را نادیده می گیرد و در صورت اجرا DPL مرجع اجرا خواهد شد و در این حالت توانایی اعمال تغییرات در برنامه نخواهیم داشت. گزینه دیگر Result Parameters است که در صورت نیاز به بازگرداندن پارامتری بعد از اجرای برنامه نوع و نام آن پارامتر در اینجا تعیین می شود. این امر در استفاده از DPL به عنوان زیربرنامه می تواند بسیار مفید واقع شود.

Script برگه Script

در برگه Script نیز متن اصلی برنامه نوشته می شود که در صورت استفاده از Script نیز متن اصلی برنامه نوشته می شود در قسمت پایینی این صفحه با کلیک روی التخ صفحه تنها آدرس DPL مرجع مشاهده می شود. در قسمت پایینی این صفحه با کلیک روی التخا open text editor مرجع مشاهده می شود که می توان متن برنامه را در آن نوشت. هنگام بستن از ما می پرسد که آیا تغییرات را در صفحه Script اصلی اعمال کند یا خیر. توضیحات اضافی در مورد عملکرد برنامه در برگه Description می تواند نوشته شود که برای اطلاع بیشتر کاربر دیگر از نحوه عملکرد برنامه است.

با فشردن Check میتوان محتویات فولدر DPL را مشاهده کرد. Close موجب بسته شدن صفحه DPL و ذخیره شدن تغییرات و Cancel و فشردن Yes موجب بسته شدن بدون اعمال تغییرات خواهد شد. Execute برنامه را اجرا نموده و Save تغییرات را در برنامه اعمال و ذخیره میکند. کلیک روی Check نیز اشتباهات برنامه را از نظر Syntax برمی گرداند. البتته تنها از لحاظ نوشتاری که اشتباهات در پنجره خروجی قابل مشاهده است. می توان از گزینه 🍽 (Break) در Toolbar صفحه اصلی، جهت توقف برنامه در حال اجرا استفاده نمود.

۶- زبان برنامهنویسی
 در اینجا مانند +++C، همه عبارات به ";" ختم می شوند.

-۱-۶ متغیرها در DPL

در ابتدای برنامه و قبل از نوشتن الگوریتم، باید متغیرهای مورد استفاده در متن برنامه را تعریف کرد که به شکل زیر است:

Variabletype varname1,varname2;

انواع متغیرها(Variabletype) در برنامهنویسی Digsilent عبارتند از:

int : عدد صحيح

double : اعشارى

string : رشتهای از کاراکترها

object : یک شی

set : مجموعهای از اشیا

۲-۶ برخی دستورات در DPL

اپراتورهای ریاضی و توابع خاص قابل استفاده خواهند بود.

علایم =+ و =- نیز برای کاستن یا افزودن مقدار به متغیر استفاده می شود. تعدادی از توابع استاندارد در جدول (۶-۱) دیده می شود. دستورات حلقه مانند for ،While II قابل استفادهاند که فرم آنها در برنامه هایی که در ادامه نوشته شده است قابل رویت است. عملگرهای منطقی مانند. (.not.)، (.and.)، (.or.) و غیره به عنوان شرط استفاده می شود. از علامت "!" برای نوشتن توضیحات در متن برنامه استفاده می گردد که به رنگ سبز بوده و کامپایلر هنگام اجرا آن را در نظر نمی گیرد. دستور tig track در هر جای برنامه که استفاده شود باعث خروج از برنامه خواهد شد. چند دستور در داخل حلقه استفاده می شود که در زیر لیست شدهاند.

جدول (۶-۱): توابع ریاضی استاندارد DPL

asin(x)	tan(x)	cos(x)	Sin(x)
$\cosh(x)$	sinh(x)	atan(x)	acos(x)
log(x) (basis 10)	$\ln(x)$	exp(x)	tanh(x)
sqrt(x) (square root)	max(x,y)	min(x,y)	abs(x)

- Break اجرای این دستور که در داخل حلقه استفاده می شود، موجب خروج از حلقه و رفتن
 مکان نما به اولین دستور بعد از حلقه خواهد شد.
- Continue: این دستور نیز داخل حلقه استفاده می شود و مکان نما را به سطر مربوط به شرط حلقه منتقل خواهد کرد.
- Input(varname, Text) با اجرای این دستور پنجرهای باز می شود که می توان از کاربر
 مقداری را گرفت. مقدار وارد شده در varname ذخیره می شود. 'Text' نیز متن توضیحی

است که در پنجره موردنظر نوشته میشود.

- ('Output' description=varname') این دستور در پنجره خروجی پس از نوشتن توضیحات Description مقدار متغیر varname را مینویسد.
- (Printf('Description%x',varname) متغیر مطلوب را بجای علامت % مینویسد. جزییات بیشتر در Manual قابل دسترسی است که توضیح آن از حوصله بحث خارج است.
- (Text%x',varname) این دستورات که آرگومانهای ورودی آن مانند دستور printf میباشد، برای ایجاد هشدار، اطلاع و پیغام خطا در پنجره خروجی استفاده میشود که به ترتیب دارای رنگهای جگری، سبز و قرمز میباشند.
- دستورات رشته زاز دستورات strcpy ،strlen ،strstr به ترتیب برای جستجوی رشتهای در رشته دیگر، محاسبه طول رشته، کپی رشتهای در رشته دیگر و مقایسه دو رشته استفاده می شود.
- Strftime(format) : با توجه فرم تعریف شده رشته ای درخروجی می دهد که مثالی از آن را در زیر می بینید.

Str = strftime('Today is %A, day %d of %B in the year %Y.');

- ()Setname.Clear : با این دستور همه المانهای موجود در یک set پاک خواهد شد.
- ()EchoOff : در مواقعی استفاده می شود که بخواهیم به علت عدم نیاز به المان خارجی، ارتباط DPL با اشیا خارجی را قطع و سرعت محاسبات را افزایش دهیم. همچنین اگر این فرمان را در ابتدای برنامه و بعد از معرفی متغیرها نوشته باشیم، در پنجره نمایش خروجی فقط مطالبی که در متنDPL اشاره کرده اید، نمایش داده می شوند و مطالب دیگر نمایش داده نمی شوند.

- ClearOutput : اگر میخواهید هنگام مشاهده خروجی های برنامه، مطالب قبلی موجود در
 پنجره نمایش، پاک شوند، این فرمان را در ابتدای برنامه و بعد از معرفی متغیرها، وارد نمایید.
- Set.SortToName (int R) : با این فرمان object های موجود در setی که نام آن در ابتدای دستور آمده است، بر اساس نام مرتب می شوند. اگر int R برابر 0 باشد، مرتب سازی از A و اگر برابر 1 باشد، از Z شروع می شود.
- object = Set .First() ایا این فرمان اولین object از set از set که نام آن پس از مساوی آمده
 است، در object که نام آن قبل از مساوی آمده است، وارد می شود.
- object = Set .Next() با این فرمان، object بعدی از set یکه نام آن پس از مساوی آمده
 است، در objectی که نام آن قبل از مساوی آمده است، وارد می شود.
- (mode, int iFH) معرفی شده فایلی با fopen (string Path, string Mode, int iFH) معرفی شده فایلی با مشخصه معرفی شده (Mode) باز میکند. iFH کدی است که موقع بستن فایل کارآیی دارد.
- (string ClassName) اولین object= GetCaseObject اولین Class Name از مساوی آمده است شده را از کیس مورد مطالعه انتخاب کرده و در objectی که نام آن قبل از مساوی آمده است قرار میدهد.
 - (object= GetCaseCommand (string ClassName) مشابه قبلی

P-۶- معرفي المان يا دستور به DPL به عنوان Object

سه Object از پیش تعریف شده وجود دارد که عبارتند از ۲۰ - SEL، متغیری است که در صورت تعریف Object از پیش تعریف شده وجود دارد که عبارتند از SEL قرار می گیرند. دیگری NULL که یک object خالی است که درون آن المانی نیست و سومی this که همان DPL در حال اجرا است. برای استفاده از المانهای شبکه (خطوط، ترانسفورماتورها و ...) و دستورات مطالعاتی (پخش بار، شرایط اولیه، شبیه سازی و ...) در متن برنامه، باید ابتدا آنها را به DPL مورد نظر معرفی کنیم. برای این منظور چندین روش مطرح وجود دارد.

۱- اخذ آن از Study case فعال در متن برنامه توسط فرمان GetCaseCommand یا
 ۱- اخذ آن از GetCaseObject فعال در متن برنامه توسط فرمان معرفی یک دستور خاص مناسب
 GetCaseObject که بصورت زیر انجام می گیرد. این شیوه برای معرفی یک دستور خاص مناسب
 است و معمولاً برای انتخاب المان ها کارآیی ندارد، زیرا قدرت انتخاب را به کاربر نمی دهد. نحوه
 استفاده از این فرمان در بخش ۶-۱ آمده است.

بطور مثال فرمانهایی که در ادامه میآیند، دستور "Load Flow Calculation" را با نام LFC(انتخاب نام به دلخواه انجام میگیرد)به برنامه معرفی میکند و سپس آن را اجرا میکند.

LFC=GetCaseCommand('ComLdf'); LFC.Execute(); LFC معرفی دستور استفاده می شود، مثلاً LFC در این روش لازم است در ابتدای برنامه نامی که برای معرفی دستور استفاده می شود، مثلاً

اگر آن کلاس در Study case وجود نداشته باشد، این دستور NULL برمیگرداند. نام کلاس نیز از عنوان پنجرهی دستور مورد نظر مشخص می شود که در شکل (۱-۶) برای پخش بار

نشان داده شده است.

Load Flow Calculat	ion - Study Case(1)\Load	Flow Calculation ComLdf	? 🔀
Iteration Control Ou Basic Options	puts Low Voltage Analysis Active Power Control	Advanced Simulation Options	Execute
df/secc			Close
 Network-representation Balanced, positive Unbalanced, 3-p 	on re sequence hase (ABC)		Cancel

شکل(۶-۱): نام کلاس دستور پخش بار

۲- استفاده از External Object که قبلا گفته شد. با استفاده از پنجره External Object، دستور یا المان مورد نظر را به DPL مورد نظر معرفی کرده و سپس در متن برنامه به راحتی می توان آن را فراخوانی کرد. در این روش نیازی نیست که دستور یا المان معرفی شده به DPL را دوباره در ابتدای متن برنامه معرفی کنیم.

DPL Comm	and - Study Case(1)	WPLscript1.Co	mDpl*			? 🛛
Basic Option	S Advanced Options S	Script Description	1			Execute
<u>N</u> ame	DPLscript1					Close
General Sel Input param	ection 🛛 🗲 🔶 eters:					Cancel
Ту	pe Name	Value	Unit	Description		Save
▶ 1						Check
						Contents
1					•	

شکل ۶-۲) استفاده از پنجره External Obgect

بطور مثال اگر بخواهیم دستور "Initial Condition" را در برنامه اجرا کنیم، ابتدا در پنجره External مورد مثل معرفی Object این دستور را با نام Pinc(انتخاب نام به دلخواه انجام می گیرد) به DPL مورد نظر معرفی کرده(شکل۶-۲)، سپس آن را در متن برنامه اجرا می کنیم.

Pinc.Execute();

۳- تعریف یک فیلتر در Contents

بر روی آیکن Contents در سمت راست پنجره DPL Command کلیلک کرده و در پنجره جدید DPL روی آیکن New Object کلیک میکنیم. در پنجره جدید (Element Selection) گزینه DPL کزینه را انتخاب میکنیم Command and more را انتخاب و از کرکرهی Element، گزینه General Filter را انتخاب میکنیم و OK میزنیم. این روند در شکل ۶-۳ بصورت تصویری ارائه شده است.

	Basic Options Advanced Options Script Description Program text	Execute Close
Cancel Contents Contents Contents Contents Contents Controllers/Prime Mover Units/Motor Driven Machines Composite Model Common Model Common Model DPL Command and more Others Element General Filter (SetFilt)	Element Selection - Settings\Default\Element Selection.IntNe ? X Elements	Cancel Save Check
Element General Filter (SetFilt)	Cancel Bus Net Elements Lypes for Net Elements Dentrollers/Prime Mover Units/Motor Driven Machines Composite Model Common Model Block Diagram DPL Command and more Qthers	Contents Contents
0 object(s) of 0 0 object(s) selected	Element General Filter (SetFilt)	

شکل ۶-۳) روند انتخاب General Filter

در پنجره جدید (پنجره Titer (پنجره General Filter)، نام فیلتر مورد نظر و Object هایی که می خواهیم انتخاب شوند را مشخص می کنیم. همچنین گزینه Relevant Objects For Calculation را انتخاب می کنیم General تا فیلتر بداند در کجا به دنبال Objectهای موردنظر جستجو کند. در شکل ۶-۴، پنجره Filter

		I OK
Name General Filter		Cancel
 Relevant Objects for Calculation Area Interconnecting Branches 	All Busbar Busbar Termin Synchr Line Genera 2-Wind 3-Wind Extern	+ Terminal al onous Machine I Load ing Transformer al Grid

General Filter پنجره (۴-۶

تا اینجا یک مجموعه (SetFilt) با نام دلخواه ما ساخته شده است که هنوز محیط برنامه آن را نمی شناسد. باید با استفاده از فرمان Get محتوای آن را گرفته و در یک set قرار دهیم. فرمان زیر محتوای SetFiltی با نام GensFilter را گرفته و در setی با نام Gens قرار میدهد. سپس محتوای set را بر حسب حروف الفبا مرتب کرده و اولین آن را انتخاب میکند و در tobjectی به نام Gen قرار میدهد.

Gens=GensFilter.Get(); Gens.SortToName(0); Gen=Lines.Firs; ۴- Copy کردن دستور مورد نظر از Study case مزبور و Patse کردن در فولدر DPL در
 مفحه data base.

در این قسمت، نام دستور مورد استفاده قابل تغییر است. نا گفته نماند اگر در ابتدای برنامه نام این دستور به عنوان یکی از انواع متغیرها معرفی شود، نرمافزار ایراد می گیرد. بنابراین در متن برنامه نیازی به معرفی دوباره آن نیست.

بطور مثال برای اجرای پخش بار، ابتدا همانطور که در شکل ۶-۳ نشان داده شده است، ابتدا پنجره Data Manager را باز کرده و با کلیک بر روی Study Case مزبور، دستور Load Flow را به Calculation را از پنجره سمت راست انتخاب و در DPL موردنظر خالی میکنیم. نام آن را به دلخواه تغییر داده و پنجره را می بندیم. در این مثال نام دستور پخش بار را LFC انتخاب کرده ایم (نام انتخابی باید مشخصات نام یک متغیر را داشته باشد بطور مثال نباید در بین حروف آن فاصله باشد).

شکل (۳-۶): نحوه دسترسی به یک المان در DPL

حال می توان به راحتی دستور پخش بار را در محیط برنامهنویسی اجرا کرد:

LFC.Execute();



هر المان یا دستور که از این طریق به برنامه معرفی شود، در قسمت Contents قـرار مـیگیـرد. بهتـر است از این روش فقط برای معرفی دستورات اسفاده شود.

۵- با استفاده از General Selection

در این روش ابتدا به طریقی که در بخش ۵-۱ بیان شد، المان یا المانهای موردنظر را انتخاب کرده و بعنوان یک set معرفی می کنیم (تا اینجا یک General Set ایجاد شده است، که می توان پس از باز کردن پنجره DPL ، در سربرگ Basic Option کرکره General Selection را باز کرد و با انتخاب گزینه dda، المانهای دیگری نیز به آن اضاف کرد). سپس با دستوری مشابه General می توان Lines می توان Lines دلخواه را (در اینجا AllLines) از مجموعه General Sel Set انتخاب کرد و در set موردنظر (در اینجا Selection) قرار داد.

برخی دیگر فرمانهای مشابه عبارتند از:

AllTypLne () ، AllSym () ، AllRelevant (string S |int i) ، AllOpenBrakers() ، AllLoads() فرمانهای زیر، تمام خطوط موجود در General Selection را می گیرد، در set یه نام Lines قرار داده، به ترتیب حروف الفبا مرتب کرده و اولین آن را در objectی به نام Line قرار می دهد.

Lines=SEL.AllLines; Lines.SortToName(0); Line=Lines.First();

v- نحوه دسترسی به پارامترهای یک object

هر المانی که از طرق مختلف به برنامه شناسانده می شود، مثلا از طریق Exernal Object یا با دستور GetCaseCommand در متن برنامه یا روشهای دیگر، دارای پارامترهایی می باشد که به راحتی قابل دستیابی هستند.

نحوه دسترسي به پارامتر خاصي از يک المان: ObjectName:ParameterName

اما برای اینکه نام پارامترها را بدانیم، باید گزینه فن (edit result variable) را از نوارابزار اصلی برنامه انتخاب و با دو کلیک روی monitor variable set و سپس انتخاب object موردنظر پارامترهای آن را دید. نوع پارامترها در variable set قابل تغییر است.

Edit Format for میتوان بر روی جعبه Result المانها در شبکه راست کلیک کرد و گزینه Result را انتخاب Edge Element را انتخاب کرده، در پنجره Form-Chang Settings، آیکن Input Mode را انتخاب کرده و در پنجره Predefined Variables گزینه Variable Selection Mode را انتخاب کنید. در اینصورت نام برخی از پارامترها را در پنجره Variable Selection Mode خواهیم دید. همچنین می توان در پنجره ای برخی از پارامترها را در پنجره Input Variables خواهیم دید. همچنین می توان در پنجره Variable Selection Mode گزینه کال User Selection Settings خواهیم دید. همچنین می توان پنجره Form-Chang Settings گزینه کال در انتخاب کرد در اینصورت در پنجره Select Variable Selection میتوان با فشردن آیکن Select Variable به نام همه منغیرهای مربوط به خط یا شینه دسترسی پیدا کرد (این کار را برای پنجرههای خط و شین میتوان انجام داد). برای گزینههایی مانند اتصال کوتاه، پخش بار و غیره، هنگامی که پنجره مرتبط با آن را برای تنظیم باز شود، با نگهداشتن ماوس روی هر گزینه یک Pal بالونی باز میشود که نام آن پارامتر که در Input Select Cariaba به تام آن پارامتر که در مورد استفاده و تغییر است، مینویسد. مثلا در شکل (۷-۱) برای اتصال کوتاه، برای Fault Type نام iopt_shc قابل رویت است. چند نوع مطرح اتصال کوتاه عبارتند از: تکفاز به زمین (spgf)، دو فاز (2psc) و سهفاز (3psc). همچنین می توان نوع آزمایش اتصال کوتاه را انتخاب کرد: VDE(0)، (1)IEC.

در DPL بسته به نوع شی دستورات مختلفی داریم که به صورت زیر بکار میرود.

Iret = Object.Function (arg1,arg2,....,argn)

برخی توابع داخلی DPL هستند. برخی نیز توابع برای همه المانها قابل اجرا هستند، اما برخی تنها برای نوع خاصی از المانها و متغیرها کاربرد دارند، مثلا تابع hasroute فقط برای خطوط و یا ()First تنها برای Set کاربرد دارد. در Manual این توابع در چهار گروه و متد object ،set، داخلی و خارجی کاملا جداسازی شدهاند. توابعی مانند Manual مختص AddVar،WriteDraw ،Write مختص Result Object

دو دستور EchoOn و EchoOff به ترتیب برای وصل و قطع ارتباط DPL با اشیا خارجی استفاده می شود. در مواقعی استفاده می شود که بخواهیم به علت عدم نیاز به المان خارجی ارتباط را قطع و سرعت محاسبات را افزایش دهیم و در پایان دوباره آن را برقرار نماییم.

	avanced uptions verification	Execute
shc/full/spgf/a	SC	
dathad	Select 'complete' method	Llose
	to calculate multiple faults.	Cancel
-ault Type	Single Phase to Ground Multiple Faults	
load Flow	contin Parameter Name: iopt_shc	
Phase	a 💌	
Fault Impedan	ce Short-Circuit Duration	
Resistance, R	f 0. Ohm <u>B</u> reaker Time 0.1 s	
Reactance, X	f <mark>0. Ohm</mark>	
Output		
🔽 On		
Command	contingency\Output	
Shows	Fault Locations with Feeders	
-Fault Location		
🥅 At all Bust	ars and <u>T</u> erminals	
Liser Selection	▼ ↓	
0301 00100000		

شکل (۲-۷): یافتن نام پارامتر Fault Type برای Short Circuit Command

دستور ResetCalculation در واقع کار آیکون 🔎 را انجام میدهد و نتایج محاسبات قبلی را پـاک خواهد کرد.

دستور ('AllRelevant''.ClassName در واقع کار آیکون 💒 را انجام میدهد، یعنی کلاس خاصی از المانها را می توان با آن انتخاب نمود و در یک Set قرار داد. البته اگر در آیکون 🚰 روی هر المان ماوس را نگهداریم، نام کلاس آن المان در Help بالونی نمایش داده خواهد شد. البته باید دانست که امکان استفاده از Function به عنوان شرط برای if بطور مستقیم وجود ندارد و بابد از یک متغبر واسطه استفاده کرد.

از دستور ()Delete برای پاک کردن یک شی یا مجموعهای از اشیا استفاده می شود که می توان در صورت نیاز آنها را از Recycle Bin در صفحه Data Manager بازیابی نمود (با کلیک راست و Restore کردن آن المان). از دستور ()ClearOutput برای پاک کردن پنجره خروجی استفاده می شود. دستور data نیز صفحه ورودی را پاک خواهد کرد. آیکون آت (window) در ata manager صفحه ورودی را به ما نشان می دهد که همه دستورات مطالعات انجام شده در آن ذخیره

می شود. می توان در صورت نیاز دستورات موردنیاز را ذخیره کرد. بدین صورت که با انتخاب آنها و کلیک راست و انتخاب گزینه save selected rows به صورت فایل cmd ذخیره کرد و در صورت نیاز با کلیک راست در صفحه input window و انتخاب گزینه open command file تکتک دستورات آن فایل را برای هر یروژهای اجرا نمود.

۱-۵)مثالهایی از برنامهنویسی در DPL و تشریح آنها

به عنوان مثال برنامهای آورده می شود که دو عدد را از ورودی گرفته و مجموع آنها را در خروجی با رنگ آبی می نویسد.

double x,y,z; input(x,'enter first number to add : ');

input(y,'enter second number to add : ');

z=x+y;

printf('\ce the result o sum is %f'',z);

مثال دوم برنامهای است که تمام خطوط را گرفته و طول آن را چک میکند، اگـر بـیش از ۱ کیلـومتر بود inform، اگر کمتر از ۱ کیلومتر بود error و اگر برابر بود، warning میدهد. خطوط از چند طریق قابل معرفی به برنامه هستند. اول اینکه می توان همه آنها را به عنوان المان External در صفحه Basic Option به آن افزود که کار چندان مناسبی نیست. در روش دوم میتوان از معرفی یک SEL استفاده کرد که به صورت comment در برنامه زیر نوشته شده است. در روش سوم می توان از دستور AllRelevant استفاده کرد که در زیر مشاهده می شود.

```
set S;
object O;
string name;
double length;
ClearOutput();
!S=SELL.AllLines();
S=AllRelevant('*.ElmLne');
O=S.First();
while(O)
 {
length=O:dline;
 name=O:loc name;
 if (length>1)
  Info('The Length of line %s is greater than 1',name);
 if (length<1)
   Error('The Length of line %s is smaller than 1',name);
 if (length=1)
   Warn('The Length of line %s is equal 1',name);
 O=S.Next();
 }
```

اگر بخواهیم در صورت وجود نتایج پخش بار از آنها استفاده کنیم و دوباره آن را انجام ندهیم و در زمان صرفهجویی نماییم، از دستور ()ValidLDF استفاده کنیم که اگر خروجی دستور ۱ باشد، یعنی نتایج وجود دارد و نیازی به انجام دوباره آن نیست.

مثال: برنامهای بنویسید که در ۵۰% خطی که در بخش المان خارجی برای برنامه تعریف می شود، طبق استاندارد ANSI اتصال کوتاه نامتقارن تکفاز ایجاد کند.

برای استفاده از دستور اتصال کوتاه باید از GetCaseCommand استفاده کرد و یا آن را در پوشه DPL مربوط کپی نمود. روش دیگر نیز تعریف آن به صورت المان خارجی است. در اینجا از روش اول استفاده میکنیم. از طرفی باید پارامترهای آن را تنظیم کنیم. در درجه اول نام آنها از همان روش نگهداشتن ماوس روی گزینه مربوط در صفحه اتصال کوتاه تعیین میشود. Object مورد نظر برای انجام اتصال کوتاه باید از طریق External Object تعریف شود یا از طریق دستور

('AllRelevant('ObjectName.ClassName') تعيين شود

!set S: object shc,O;!,Line4; ClearOutput(); shc=GetCaseObject('ComShc'); shc:iopt mde=2; ! ANSI Method ! Single Phase To Ground shc:iopt shc='spgf'; !S=AllRelevant('Line 4.ElmLne'); !Line4=S.First(); shc:iopt allbus=0; ! Deactivate Short Circuit For All Bus shc:shcobj=Line4; ! Select Object To Cal SHC shc:ppro=50; ! Percent Of Line shc.Execute();

روش دیگر تعیین نام پارامترهای خروجی برای استفاده در DPL محاسبات مانند بارگذاری یا ولتـاژ و جریان کلیک راست روی result box در شبکه و انتخـاب گزینـه Edit/Format Element است کـه در Line Panel نـام متغیـر و نـام مـورد اسـتفاده در DPL قابـل مشاهده است. اولین حرف از نام این پارامترها یکی از حروف زیر است که نـوع آن پـارامتر را نشـان میدهد که در زیر می بینید.

- S : Signal Parameters
- C : Calculation Parameters
- M : Measurment Parameters
- R : Reference Parameters
- T : Type Parameters

از دستور ('SoreToVar(0|1,'Parameter Name) می توان برای مرتب کردن اشیای درون یک Set براساس متغیر دلخواه به صورت صعودی یا نزولی (۱ یا ۰) استفاده نمود. برنامهای ارایه می شود که با Sweep کردن روی یک خط، در هر ۱۰ درصد آن خطا ایجاد نموده و توان اتصال کوتاه را در تک تک خطوط بدهد. همان روند قبلی باید تکرار شود، اما باید درون یک حلقه پارامتر درصد محل وقوع اتصال کوتاه یعنی iopt_ppro را تغییر داد و درون همان حلقه همه خطوط را دریافت و توان اتصال کوتاه هر یک

را در خروجی نوشت.

int i; double load; string name; set S; object shc,O; ClearOutput(); shc=GetCaseObject('ComShc'); shc:iopt mde=0;! ANSI Method

```
shc:iopt allbus=0;! Deactivate Short Circuit For All Bus
   shc:shcobj=Line4; ! Select Object To Cal SHC
   for (i=10;i<=100;i+=10)
   {
    shc:ppro=i;! Percent Of Line
    shc.Execute(); ! Execute Short Circuit Command
    S=AllRelevant('*.ElmLne');
    O=S.First();
    while(O)
     {
    load=O:m:Skss:bus1;
    name=O:loc name;
    printf('current of %s is %f .',name,load);
    O=S.Next();
    }
   }
در بحث قابلیت اطمینان گاہی اوقات نیاز داریم در درون یک شبی ماننـد Cubicle هـا شبی دیگـری
مانند فيوز،رله، CT و يا PT تعريف كنيم(مانند يك بي). از اين طريق بـراي شـبكه حفاظـت تعريف
              کردهایم. برای ایجاد یک شی درون شی دیگر از دستوری به شکل زیر بهره می گیریم.
```

Object.CreateObject('ClassName',name)

نام می تواند شامل دو قسمت باشد. مانند آنچه در برنامه زیر استفاده شدهاست تا بتوان برای هر فیوز نام متفاوتی برگزید. مثالی در زیر آورده شدهاست که برای هر Cubicle موجود در شبکه Nine Bus یک فیوز قرار می دهد.

در ابتدا باید همه Cubicle ها را انتخاب کرد. از SEL نمی توان استفاده کرد، زیرا شامل Cubiclها نمی گردد، البته می توان از طریق Edit کردن General Set، به آن Cubicle ها را نیز افزود اما کار دشواری خواهد بود. می توان از دستور ('ClassName')SEL.GetAll همه Objectها را انتخاب کرد و با دریافت پارامتر bus آن، به Cubicle دسترسی پیدا کرد و یا با دستور (Cubicle می توان به Cubicle موجود در اتصال ذکر شده در هر Cubicle موجود در اتصال ذکر شده در هر

المان دسترسی پیدا کرد. اما در زیر روش مشابه دیگری برای این کار استفاده شده است. المان دیگری که در DPL قابل تعریف و استفاده است، فیلتر است. در Data Manager و در قسمت New Object و New Object با انجام تنظیماتی که در شکل (۶-۱) دیده می شود می توان یک شی فیلتر به لیست المان های موجود در مورد مطالعاتی افزود. برای استفاده عادی از فیلتر می توان نام و نوع فیلتر و شرط موردنظر (در برگه Advance) را تعیین نموده و فیلتر را اجرا نمود. نتایج در خروجی قابل مشاهده خواهد بود. در تنظیمات در DPL نیز قابل استفاده خواهد بود که با دستور زیر انجام پذیر است:

intenalObject = FilterName.Get(); گاهی اوقات میخواهیم برای بار منحنی روزانه تعریف کنیم و پخش بار را در تمام ساعات انجام دهیم. بعد از مرور بر نحوه تعریف منحنی بار در DIgSILENT برنامه انجام پخش بار در تمام ساعات شبانهروز در ادامه آورده شدهاست.

برای تغییر زمان در DPL می توان شی Set Date & Time Calculation را با دستور GetCaseObject از پایگاه داده دریافت نموده و تغییرات مطلوب را در آن ایجاد نمود که نحوه دسترسی به پارامترهای آن قبلا بیان شد. البته دستورهای (time.settime(hour,min,sec) زمان را تنظیم کرد. مبدا زمانی این دستور از ابتدای سال جاری کامپیوتر است. دستورات ()time.Time و ()date.Date نیز زمان برنامه را با زمان دستگاه منطبق می کند.

اگر بخواهیم بعد از تعریف منحنی بار، پخش بار را در تمام ساعات انجام دهیم، یک راه آن تغییر زمان بطور دستی و از طریق آیکون زمان بطور دستی و از طریق آیکون بهتر استفاده از DPL است. یعنی با تغییر زمان در یـک حلقـه و انجـام پخـش بـار نتـایج را در تمـام ساعات شبانهروز در خروجی نشان داد.

برای تعریف منحنی بار، در پوشه library در پروژه موردنظر کلیک راست نموده و گزینه New>others را انتخاب می کنیم، سیس گزینه Charactristic و از منوی فعال شده گزینه New>others characteristic vector را انتخاب و بر ok کلیک می کنیم. پنجر وای تحت همین عنوان گشوده می شود. در آن باید در قسمت Scale، گزینه Select را انتخاب کرده و در پنجره گشوده شده، کلیک راست نموده و New>others را انتخاب میکنیم. پنجره دیگری باز خواهد شد که باید در آن گزینه Time Scale را انتخاب نماییم. پنجره Time Scale باز می شود که در آن نام و نوع زمان مطلوب وارد می شود و مقادیر Scale نیز وارد خواهد شد که همان مقادیر محور افقی خواهد بود که بط ور معمول برای بار روزانه از ۱ تا ۲۳ خواهد بود. پس از کلیک Ok صفحه اصلی Tr تا ۲۳ خواهد او ا دوباره باز خواهد شد که در آن مقادیر Scale نیز وارد شدهاند. درصد مربوط که می خواهیم در آن ساعت، در بار پایه ضرب شود را در ستون دوم یعنی Value وارد می نماییم. مثلا اگر بخواهیم توان اکتیو باری را با این منحنی تعریف کنیم، با استفاده از آیکون 🌌 و یا هر روش دیگر، خط موردنظر را انتخاب و در قسمت Flexible Data و در ستون Act. Pow. کلیک راست نموده و Charactristic>New Characteristic>Reference را انتخاب می کنیم. در ادامه با انتخاب Scale موردنظر عدد توان اکتبو تغییر رنگ خواهد داد.

در ادامه برنامه DPL مرتبط برای اجرای پخش بار در ساعات مختلف آورده شدهاست.

object LDF,O,LoadA; set Load; int i; ClearOutput(); O=GetCaseObject('*.SetTime'); LDF=GetCaseObject('*.ComLdf'); O:hour=0; O:min=0; O:sec=0; for(i=0;i<24;i+=1) { O:hour=2; LDF.Execute(); } multicle for the form of the form

Result.AddVars(ObjectName,'ParameterName')

به عنوان مثال برای ذخیره بار خطی به نام Line در Result داریم:

Result.AddVars(Line,'c:loading') البته یک بار استفاده از این دستور کافیست تا هر بار که پارامتر مربوطه مقدار جدیدی به خود می گیرد، به بردار مقادیر قبلی این پارامتر در Result اضافه شود و نیازی به استفاده مکرر برای هر بار ذخیره نیست دو دستور دیگر که برای المان نتایج بسیار پرکاربرد است، ()Result.Write و ()Result.WriteDraw است که اولی فقط نتایج را در خروجی می نویسد و دومی علاوه بر نوشتن در صورت وجود پنل، رسم نیز می نماید.