

1. @-Formules

Prefikso @ vartojimas kilęs nuo dažno jo vartojimo funkcijų varduose. @-formules dažniausiai vartojamos:

- reikšmės apskaičiuoti ir vaizduoti dokumentuose ar VIEW;
- automatiniam įvedamos lauke reikšmės transformavimui;
- įvedamų lauke reikšmių korektiškumui tikrinti;
- dokumento antraštes suformavimui;
- dokumentu išrinkimui;
- lauku įtraukimui ir šalinimui dokumentuose;
- vartojamos formos nustatymui atidarant dokumentą;
- ir pan.

Formules paprastai įvedamos spec. lange. Formulų įvedimo palengvinimui paprastai interfeisas leidžia pasirinkti reikiama funkcija iš leistinų funkcijų sąrašo.

Formulių užrašymo taisyklės. Formules sudaro vienas ar keletas operatorių, kurie vykdomi iš eilės. Kiekvienas operatorius gali turėti savyje:

- kintamuosius;
- konstantas;
- operacijos ženklus;
- raktinius žodžius;
- funkcijų kvietinius.

Sudarant formules reikia prisilaikyti tokių taisyklių:

- formule turi prasidėti raktiniu žodžiu, kreipiniu į funkciją, lauko vardu ar konstanta, o pasibaigti lauko vardu, konstanta arba kreipiniu į funkciją;
- operatoriai atskiriami vienas nuo kito kabliataškiu; argumentai kreipinyje į funkciją taip pat atskiriami kabliataškiu;
- funkcijų argumentai apskliaudžiami riestiniaisiai skliaustais, jei funkcija neturi argumentu – skliaustai praleidžiami;
- tarp dviejų operatorių gali būti bet kiek tarpų ir tuščių eilučių, po raktinio žodžio būtinai – bent vienas tarpas;
- skirtumo tarp didžiųjų ir mažųjų raidžių nėra. Raktiniai žodžiai automatiškai pervedami į didžiąsias raides.

Formulių pavyzdžiai:

- 1) 2.5 - konstanta;
- 2) Kaina + 2 - aritmetinė išraiška panaudojant konstantą, lauko vardą ir sudėties operaciją;
- 3) D := @Created - kintamajam D priskiriama dokumento sukūrimo data (:= - operatorius, @Created – funkcija be argumentų);
- 4) @prompt([OK] ; “Pakeitimu vykdymas”; “Atlikti pakeitimai pasiūsti paštu”). Šios formules vykdymo rezultate bus parodytas dialoginis langas su antrašte “Pakeitimu vykdymas”, pranešimu “Atlikti pakeitimai pasiūsti paštu” ir mygtuku OK. Cia [OK] –raktinis žodis;
- 5) SELECT @ALL – einamajai operacijai išrenkami visi dokumentai;
- 6) LastName + “, ” + FirstName - dviejų kintamųjų reikšmių įstatant tarp jų kableli (+ - operacija) ir tarpa konkatencija.

Konstantos . Tai:

- tekstines konstantas, kurios rašomos tarp dvigubų kabučių;

- skaičiai, sveiki ir slankaus kablelio;
- data ir laikas, kurie rašomi apskliaudžiant laužtiniais skliaustais, pvz. [11:30], [98.03.12], [98.03.12 11:30], kurių formatas priklauso nuo aplinkos.

Kintamieji. Tai einamojo dokumento laukų vardai arba laikini kintamieji. Formoje apibrėžti laukų vardai ir tipai – tai atitinkamų kintamųjų vardai ir tipai. Laukai vartojami įvairiais tikslais, bet dažniausiai: pradinei lauko reikšmei nustatyti ar panaudoti lauko reikšmę reiškinyje. Priskiriant naują reikšmę laukui, prieš jo vardą reikia pavartoti priskyrimo operatorių su raktiniu žodžiu FIELD:

FIELD FirstName := "Jonaitis"

Jei laukas FirstName egzistuoja, tai pakeičiama jo reikšmė. Jei tokio lauko dokumente nėra, tai jis sukuriamas ir jam priskiriama reikšmė. Laukui naują reikšmę priskirti galima ir naudojant funkciją @SetField:

@SetField(FirstName; "Jonaitis")

Tačiau funkcija @SetField naują reikšmę galima priskirti tik jau egzistuojančiam laukui. Formulėse galima vartoti laikinuosius kintamuosius, kurie egzistuoja tik formulėje. Jų sukūrimas ir inicializacija atliekami priskyrimo sakiniu:

TempVar := 1998;

Kintamojo tipas priklauso nuo priskiriamos reikšmės tipo, tai gali būti: Text, Time, Number, Boolean. Pakeisti laikinojo kintamojo reikšmę galima ir panaudojant funkciją @Set, pvz.:

@Set(TempVar; 1999).

Operacijos. Tai priskyrimo, palyginimo, logines, aritmetines ir kitokios operacijos.

Aritmetinės operacijos (prioritetu mažėjimo tvarka): - (minus), * (daugyba), / (dalyba), + (sudėtis), - (atimtis).

Palyginimo operacijos (tarpusavyje prioritetai lygus): = (lygu), <> (taip pat ir ><, !=, !=) (nelygu), <, <=, >, >=.

Logines: ! (loginis NE), & (loginis IR), | (loginis ARBA).

Visos operacijos turi savo prioritetus – jie tradiciniai. Žemiausia prioritetą turi priskyrimas (:=). Operacijų atlikimo tvarką galima keisti panaudojant riestinius skliaustus.

Operacijos gali būti atliekamos ne tik su vienetinėmis reikšmėmis (atomais), bet ir su sąrašais. Sąrašai gali būti konstruojami apjungimo į sąrašus operacija (:), pvz.:

a : b : c : d

"Vilnius" : "Kaunas" : "Klaipėda"

Visos operacijos su sąrašais gali būti atliekamos dviem būdais:

- panariui – kiekvienas vieno sąrašo elementas dalyvauja operacijoje su kito sąrašo atitinkamu elementu. Jei vienas iš sąrašų trumpesnis, tai vietoje trūkstamų elementų vartojamas paskutinis elementas;
- visi su visais – perrenkami visi galimi variantai.

Visos šios operacijos gali būti pritaikytos ir sąrašams – panariui. Visos jos (išskyrus minusą ir papildomus nelygybes variantus) turi savo atitikmenis "visi su visais" – priekyje pavartojus žvaigždutę (*). Lyginant du sąrašus, teigiamas rezultatas gaunamas kai sutampa bent viena reikšmių pora, pvz, sulyginimo

"Vilnius" : "Kaunas" = "Vilnius" : "Klaipėda"

rezultatas yra Tiesa.

Raktiniai žodžiai. Galimi penki raktiniai žodžiai:

- DEFAULT apibrėžia lauko reikšmę "pagal nutylėjimą". Jei dokumente lauko reikšmė vykdant formulę yra neapibrėžta, tai priskiriama nurodyta reikšmė, pvz.:

- DEFAULT FieldName := "Public";
- ENVIRONMENT leidžia nustatyti reikšmes aplinkos kintamiesiems, esamiems faile NOTES.INI. Tokių reikšmių nuskaitymui vartojama funkcija @Environment, pvz.:
ENVIRONMENT MAXIMAISED := "1";
VarName := @Environment(MAXIMAISED);
 - FIELD leidžia priskirti reikšmę laukui:
FIELD FieldName := Value;
 - REM – komentaras, kuris rašomas tarp dvigubų kabučių vienoje ar keliose eilutėse:
REM "Tai komentaras";
 - SELECT nustato dokumentų atrinkimo kriterijus. Bus atrinkti tik tie dokumentai, kurie tenkins sąlygą, einančią po žodžio SELECT, pvz.:
SELECT @All;
SELECT FirstName = "Jonaitis";
SELECT (Pareigos = "Profesorius" & Laipsnis = "hab. daktaras");

@-funkcijos . Formulėse gali būti vartojamos Lotus Notes funkcijos, kurių formatai:

@<funkcijos vardas>(<argumentas 1>; <argumentas 2>; ...; <argumentas n>);

Pvz.:

@Abs(-4);
@Created ;
@IsCategory("Yes", "No");
@Midlle(Company; 4; 4);

Argumentais gali būti ir spec. raktiniai žodžiai apskliausti laužtiniais skliaustais, pvz.:

@Prompt([OK]; "Response"; Y); (OK – raktinis zodis).

Panagrinėsime kai kurias iš jų.

@if - leidžia taip užrašyti operatorių, kad jo reikšmė priklausytų nuo loginio reiškinių reikšmės. Paprasčiausia forma:

@If(LogicalValue; Value1; Value2);

Pvz.:

@If(count > 100; "Daugiau uz 100"; "Maziau uz 100");

Pastarosios formulės rezultatas - viena iš dviejų formulėje pavartotų simbolių eilučių, priklausomai nuo sąlygos reikšmės.

Bendras funkcijos pavidalas:

@If(condition1; action1; condition2; action2; ...; condition99; action99;
else_action);

Sąlygos peržiūrimos iš kairės į dešinę ir kai tik randama patenkinta sąlyga, atliekamas greta esantis veiksmas – jį įvykdžius gauta reikšmė ir yra @if rezultatas. Jei tokios sąlygos nėra, atliekamas veiksmas else_action.

- @If(@Left(Code; 1) = "2"; "Vietinis"; "Uzmiestis");
- @If(@Length(PostIndex) = 4; @Success; @Failure("Blogas pašto indeksas"));

@Success gražina 1 (Tiesa) ir vartojama kartu su @If tikrinant lauko reikšmės korektiškumą; @Failure gražina pranešimą, apibūdinantį lauko reikšmės nekorektiškumą. @Success ir @Failure formulės vartojamos laukų Validation įvykiuose.

@IsUnavailable(Fieldname) – gražina, ar laukas Fieldname yra sukurtas einamajame dokumente.

PVZ.: @If(@IsUnavailable(TimesMailed); 0; TimesMailed + @IDocBeingMailed);

@IsUnavailable(TimesMailed) – ar dokumente nėra lauko TimesMailed (0 arba 1);
@IsDocBeingMailed – tikrinama einamoji dokumento būseną, ar jis šiuo metu yra siuntimo fazeje (0 arba 1).

@Prompt – leidžia parodyti dialoginį langą.

@Prompt([STYLE] : [NOSORT]; title; prompt; default; choices) ;

[STYLE] – raktinis žodis – būtinas parametras, apibrėžiantis lango pavidalą ir galintis įgyti vieną iš 9 reikšmių. Pvz.:

@Prompt([OK]; "Einamoji data ir laikas"; "Data: " + @Text(@Now)) ;

Vartotojui paspaudus mygtuką OK, funkcija gražins reikšmę 1. Dialoginis langas gali būti pavartotas ir reikšmių įvedimui. Įvedamam laukui galima nurodyti eikšmę:

@Prompt([OKCANCELEDIT]; "Vardo įvedimas"; "Veskite vardą";

@UserName);

Paspaudus OK bus gražinta įvesta simbolių eilutė, o paspaudus CANCEL jokia reikšmė negražinama, tai galima patikrinti palyginant rezultatą su ERR_CANCEL.

Pasirinkimo iš sąrašo pavartojimas:

@Prompt([OKCELLIST] : [NOSORT]; "Vartotojų sąrašas"; "Išsirinkite reikiama vartotoją"; "Jonas"; "Jonas" : "Petras" : "Antanas");

čia [NOSORT] – nebūtinas parametras, nusakantis, kad pateikiama sąrašą neišrūšiuoti. Paspaudus Cancel, jokia reikšmė negražinama.

@Do – leidžia nuosekliai įvykdyti keletą veiksmų – apskaičiuoti reiškinio reikšmes:

@Do(išraiška1; išraiška2; ...; išraiškaN) ;

Ši funkcija naudojama, kai norima keletą nuosekliai apskaičiuojamų reiškinio apjungti į vieną sudėtinį reiškinį. Šios formulės rezultatas – išraiškaN reikšmė.

Pateiksime formulę, kuria atidaromas dialoginis langas su klausimu vartotojui, ar jis nori redaguoti dokumentą? Jei atsakymas teigiamas, tai atidaromas naujas dialoginis langas, kuriame klausiama vartotojo vardo. Nutraukus dialogą, nutraukiamas ir formulės vykdymas. Vartotojui įvedus vardą ir paspaudus OK einamasis dokumentas atidaromas redagavimui. Jei vartotojas pirmajame lange atsisakė redaguoti dokumentą, bus pereinama prie kito dokumento.

@If(@Prompt([YESNO]; "Klausimas"; "Ar redaguoti šį dokumentą?");

@If(@Prompt([OKCANCELEDIT]; "Teigiamas atsakymas"; "Jūs išrinkote redagavimą. Jei reikšmė teisinga, spauskite OK"; @UserName)

!= ERR_CANCEL; @PostedCommand([EditDocument]); @Return(""));

@Do(@Prompt([OK]; "Neigiamas atsakymas"; "Jūs atsisakete redaguoti.

Paspauskite OK, pereiti prie kito");

@PostedCommand([NavigateNext])

)

)

Duomenų išrinkimo funkcijos

@DbColumn(class : "NoCache" ; server : database ; view ; columnNumber)

- "class" - gali būti "Notes" (jei nurodoma "" – tai atitinka "Notes") arba "ODBC".
- "NoCache" arba "" – nurodoma, rezultatai turi būti sukešuoti arba ne.
- "server"."database" – serveris ir DB. Jei nurodoma tuščia reikšmė – yra imama einamoji duomenų bazė.
- "view" – pavaizdavimo, iš kurio turi būti paimti duomenys, pavadinimas (vardas arba alias).
- "ColumnNumber" – view'e esančio stulpelio numeris, kuris turi būti gražinamas, funkcijos vykdymo rezultate.

@DbLookup(class : "NoCache" ; server : database ; view ; key ; fieldName) arba
@DbLookup(class : "NoCache" ; server : database ; view ; key ; columnNumber)

class : "NoCache" ; server : database ; view ; columnNumbe – parametrai analogiškai
@DbColumn funkcijai.

“key” – reikšmė, kurios ieškoma (t.y. funkcijos rezultate bus gražinamos reikšmės tik tų dokumentų, kurių pirmo rūšiuoto stulpelio reikšmės view’e bus lygios reikšmei “key”).

“fieldName” – lauko, kuris turėtų būti gražinamas funkcijos vykdymo rezultate, pavadinimas.

Kitos funkcijos . Visas Lotus Notes funkcijas galima suskirstyti i tokias grupes:

- išsišakojimas, blokas, grįžimas (*@If*, *@Do*, *@Return*);
- darbas su NOTES.INI failu (*@SetEnvironment*, *@Environment*);
- reikšmių priskyrimas laikiniams kintamiesiems ir laukams (*@SetField*, *:=*);
- dokumentų identifikatoriai ir kitų dokumentų laukų pasiekimas (*@DocumentUniqueID* – unikalus dokumento identifikatorius, *@InheritedDocumentUniqueID* – tėvinio dokumento unikalus ID);
- darbo su data ir laiku funkcijos (*@Date* – tekstą į datą ir laiką, *@Day* - diena, *@Hour* – valanda, *@Now*, *@Today*- einamasis data ir laikas, einamoji data);
- dokumento sukūrimo, keitimo ir kreipimosi į jį laikas (*@Created*, *@Modified*, *@Accessed*);
- darbo su simboliu eilutėmis funkcijos (*@Text* – reikšmė į tekstą, *@Word*);
- darbo su sąrašais funkcijos (*@Elements* – elementų kiekis, *@Subset* – sąrašo dalis, *@IsMember* – ar elementas priklauso sąrašui);
- logines funkcijos (*@Success* – gražina 1, *@False* - 0, *@Failure* – pranešimą);
- informacines funkcijos (*@Attachments/@AttachmentsNames* – prijungtų failų skaičius/vardų sąrašas, *@Author* – autorių sąrašas);
- funkcijos darbui su vartotoju vardais (*@UserName* – einamojo vartotojo vardas, *@UserPrivileges* – einamojo vartotojo teises, *@Name* – vartotojo vardo konvertavimui);
- dialogas su vartotoju (*@Prompt*);
- išrinkimo funkcijos; (*@DbColumn*, *@DbLookup*)
- laiškų siuntimo funkcijos (*@MailSend*).

@ - komandos. Lotus Notes turi dvi specialias funkcijas *@Command* ir *@PostedCommand*, leidžiančias atlikti Lotus Notes komandas, kurios nurodomos pirmu argumentu. Nuo vykdomos komandos priklauso kitų argumentų buvimas ir jų prasmė. Dauguma komandų atitinka Notes funkcijas atliekamas iš meniu, pvz.:

@Command([AddDatabase]; “Briedis”; “Document\\Trademrk.nsf”) ; - darbinėje erdvėje patalpinama DB, esanti serveryje “Briedis”, pakatalogyje Document, faile Trademrk.nsf. *@Command([AdminRegisterUser]* – atidaroma visa eilė dialoginių langų, realizuojančių naujo vartotojo sertifikavimą (užregistravimą).

@PostedCommand([EditDown]; “5”) - redaguojamame dokumente kursorių perveda 5 – iomis eilutėmis žemyn.

Funkcija *@PostedCommand* vykdoma po visų formulės funkcija įvykdymo, pvz.:

@PostedCommand([CommandName]; Argument);

@if(Condition; TrueStatement; FalseStatement) ;

FIELD X := “Text”;

Nors funkcija *@PostedCommand* stovi priekyje, ji bus vykdoma paskutinioji. Funkcija *@Command* vykdoma betarpiškai. Tačiau yra ir išimčių – kai kurios komandos, netgi vykdomos funkcija *@Command* veikia lygiai taip pat kaip būtų vykdomos kviečiant

@PostedCommand. Tokių komandų pvz.: *[FileCloseWindow]*, *[FileDatabaseDelete]* ir t.t.

Funkcija **@Return** – leidžia išreikštiniu būdu nurodyti gražinamą (formules) reikšmę. Formulės reikšmė – tai paskutinio operatoriaus gražinama reikšmė arba funkcijos *@Return* argumento reikšmė. Operatoriai su raktiniais žodžiais, paprastai, negražina jokios reikšmės.

Reikalavimai formulių gražinamai reikšmei priklauso nuo vartojamo konteksto, pvz.,

- dokumentu išrinkimo formulė (Replication formula, Selection Formula) kiekvienam dokumentui turi gražinti reikšmę 1 (TRUE) arba 0 (FALSE);

- formos formulė (Form formula) turi gražinti formos vardą. Tokia formulė gali būti nurodyta pavaizdavimo savybės (properties). Lotus Notes egzistuoja tokia formos parinkimo dokumento vaizdavimui tvarka:

- forma, saugoma dokumente – dokumentui forma išlieka nepakitus net, jei pati forma po dokumento sukūrimo buvo redaguota;
- forma, apibrėžta formule;
- forma, nurodyta dokumento lauke *form* (dažniausiai tai forma, pagal kurią dokumentas buvo sukurtas);
- forma, vartojama DB-je pagal nutylėjimą.

Tokios formules pvz.:

@If(@IsNewDoc; “Naujos diskusijos pradžia”; “Pagrindine tema”) ;