

Вступ

Введення в систему автоматизації «1С:Підприємство». Основні поняття і об'єкти системи.

Мета: Ознайомлення із структурою системи 1СП. Вивчення основних понять та знайомство з основними об'єктами системи. Типи даних в системі 1СП. Структура вбудованої програмної мови.

Література:

- 1) М.Г.Радченко, Е.Ю.Хрусталева. 1С:Предприятие 8.2. Практическое пособие разработчика. Ст. 143-194.
- 2) http://1c-uroki.ru/lessons/kurs1C_1. Урок 5.
- 3) http://1c-uroki.ru/articles/konstrukcii_vstroennogo_jazyka_1C_8.3.

Структура системи 1СП

На логічному рівні система «1С:Підприємство» (надалі – 1СП) виглядає наступним чином (рис.1):



Рис. 1. Структура 1СП на логічному рівні

Існує інформаційна база системи, яка в свою чергу складається із бази даних (це ті дані, які вводять користувач) та конфігурації (те, що розробив розробник-програміст системи 1СП).

Конфігурація за своєю суттю є деяким описом, тобто алгоритмом поведінки програми, а не движком. Іншими словами, це налаштування системи, опис структури БД і механізмів по їх зберіганню, накопиченню та витягуванню.

Система 1СП містить технологічну платформу, що дозволяє користувачу, з однієї сторони, описувати конфігурацію, а з іншої, платформа 1СП дозволяє виконувати те, що написав користувач, тобто це середовище виконання.

Технологічна платформа 1СП це закрита частина, куди вносити зміни користувачі не можуть, а можуть використовувати тільки те, що розроблено розробниками 1СП. Конфігурація, навпаки, по своїй ідеології, відкрита, вносити зміни користувачам дозволено.

Система 1СП є як і закрита, так і відкрита одночасно. Тобто, будь-які зміни в поведінці бізнес-логіки системи користувачі внести зможуть, а написати або перевизначити стандарту поведінку системи є не можливим або дуже важким (наприклад, створювати нові класи об'єктів в 1СП є не можливим).

Так як конфігурація, з однієї сторони, є описом не тільки того, що буде бачити користувач, але є й описом того, як це буде зберігатись в середині БД, тому конфігурацію і БД не розділяють, і їхнє об'єднання називають інформаційною базою (ІБ). Дані із інформаційної бази без використання конфігурації не мають смислу.

На фізичному рівні платформа 1СП може працювати в двох режимах – це файл-серверний та клієнт-серверний.

Файл-серверний варіант (рис.2). ІБ (тобто дані та конфігурація) зберігаються в окремому файлі на комп'ютері клієнта або мережі. Файл називається 1Cv8.1CD, це і є наша ІБ з якою буде працювати сама платформа 1СП. Дана платформа буде працювати на клієнтських машинах.

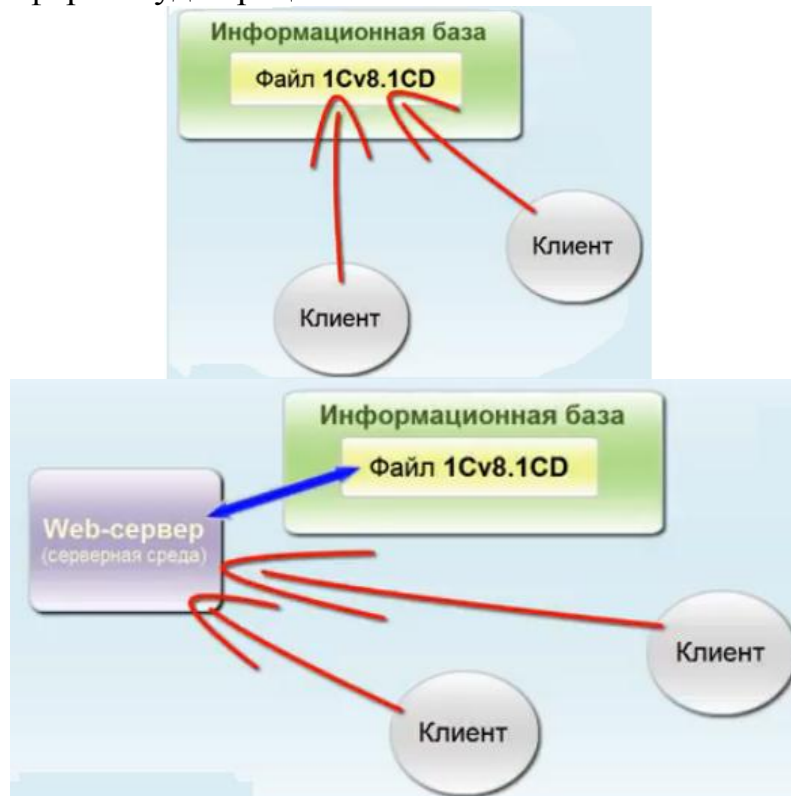


Рис. 2. Файл-серверний режим роботи платформи

Клієнт-серверний варіант (рис.3). Окремо існує ІБ, причому вона буде працювати під управлінням сервера БД, тобто SQL-сервера. На даний момент система 1СП підтримує 4 SQL-сервера: Microsoft SQL Server, PostgreSQL Server, IBM DB2 Server, Oracle Server. В загальному, всі ці сервери працюють, як файлове сховище, тому чекати, що 1СП буде використовувати різні адміністративні функції цих серверів не потрібно, це вже лягає на плечі адміністраторів БД, які повинні оптимізувати роботу серверів. 3 SQL Server

працює сервер 1СП, це окремий програмний продукт з яким в свою чергу можуть працювати або клієнтські машини або веб-сервер.

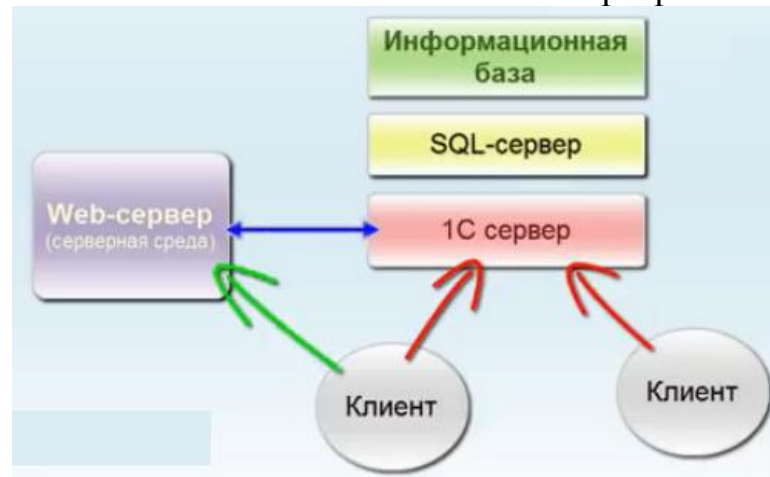


Рис. 3. Клієнт-серверний режим роботи платформи

Під'єднуватись до веб-серверу можна, або в локальній мережі, або через мережу Інтернет. В такому випадку, вся завантаженість буде лягати тільки на сервер 1СП. Клієнти будуть тільки запитувати систему на надання визначеної інформації, сервер 1СП перетворить запити у зрозумілі запити SQL серверу, SQL сервер опитує БД та повертає серверу 1СП деяку відповідь, сервер 1СП виконує деякі додаткові обробки та повертає клієнту або вибірку даних, якщо це визначений режим роботи клієнта, або картинку на якій показано частину цієї вибірки.

В системі 1СП існує варіанти роботи з конфігурацією (види клієнтів): товстий (звичайний), тонкий та web-клієнт.

Звичайний клієнт – це клієнт старої версії системи 1СП, він більш вимогливий до ресурсів, викачує з сервера БД вибірку даних, працює з ним напряду на стороні клієнта, може звертатись безпосередньо до даних конфігурації (до даних серверу), тому з'єднання з звичайним клієнтом не може бути не стабільним, повинно бути високошвидкісним.

Тонкий клієнт – показує картинку, всі дані знаходяться на стороні сервера і зберігаються у вигляді сеансів, при з'єднанні тонкого клієнту із сервером або веб-сервером, можливий не стабільний зв'язок.

Поняття об'єктів конфігурації

Конфігурація описує структуру даних, які користувач буде використовувати в режимі роботи 1СП. Крім цього, конфігурація описує всілякі алгоритми обробки цих даних, містить інформацію про те, як ці дані повинні будуть виглядати на екрані і на принтері, і т.д. Надалі платформа 1СП на підставі цього опису створить базу даних, яка буде мати необхідну структуру, і надасть користувачеві можливість працювати з цією базою даних.

Для того, щоб систему ІСП можна було швидко і легко налаштовувати на потрібні прикладні завдання, весь опис, який містить конфігурація, складається з деяких логічних одиниць, званих об'єктами конфігурації.

Об'єкт – це деяка сутність, що має певне «призначення». В загальному випадку, об'єкт може мати набір властивостей (деякі тільки на читання, деякі на модифікацію) і набір методів (що дозволяють працювати з «областю» об'єкту).

Ми можемо створювати тільки об'єкти певних видів. Але кожного виду об'єктів ми можемо створити стільки, скільки нам потрібно. Об'єкти одного виду відрізняються від об'єктів іншого виду тим, що мають різні властивості (точніше кажучи, різний набір властивостей). Об'єкти можуть взаємодіяти один з одним, і ми можемо описати таку взаємодію.

Об'єкти конфігурації володіють різною поведінкою, і вона залежить від виду об'єкта. Одні об'єкти можуть виконувати деякі дії, інші цих дій виконувати не можуть, зате у них є свій власний набір дій.

«Складні» об'єкти конфігурації складаються з більш «простих», і одні й ті ж «прості» об'єкти можуть входити до складу складних об'єктів. Така структура дозволяє спростити роботу з об'єктами конфігурації, бо якщо ми знаємо, як працювати з будь-яким «простим» об'єктом, то в будь-якому «складному» об'єкті, до складу якого він входить, ми будемо працювати з ним все тим же чином.

І, нарешті, найважливіша якість об'єктів конфігурації – це їх прикладна спрямованість. Об'єкти конфігурації не просто деякі абстрактні конструкції, за допомогою яких розробник намагається описати поставлене перед ним завдання. Вони являють собою аналоги реальних об'єктів, якими оперує підприємство в ході своєї роботи.

В системі ІСП виділяють такі види об'єктів:

1) Об'єкти конфігурації. Саме з об'єктами цієї групи доводиться мати справу в процесі конфігурування. Вони розташовуються в дереві метаданих конфігурації. Об'єкт конфігурації володіє набором властивостей (їх склад визначається видом об'єкту), методів у таких об'єктів немає. Дуже часто об'єкти конфігурації є «електронними» аналогами реально існуючих об'єктів прикладної області.

2) Об'єкти вбудованої мови. Ці об'єкти використовуються при написанні алгоритмів обробки інформації. Частина з них підтримується вбудованою мовою спочатку, частина з'являється після додавання в конфігурацію об'єкту.

Всі об'єкти конфігурації, які існують в системі ІСП, утворюють декілька основних видів. Кожний вид об'єктів конфігурації є якраз тими «будівельними елементами», з яких створюватиметься конфігурація. Розбиття об'єктів за видами можна побачити в дереві об'єктів конфігурації, вони знаходяться на першому його рівні (рис.4).

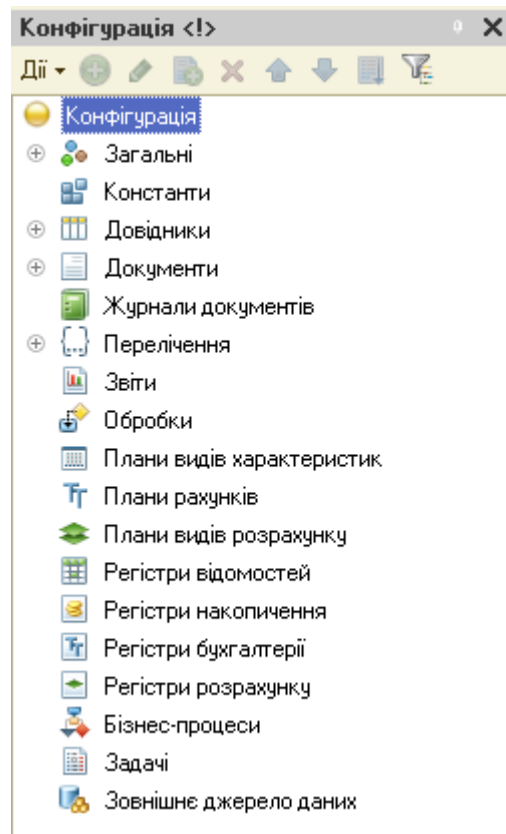


Рис. 4. Вікно дерева об'єктів конфігурації

Можна сказати, що дерево об'єктів конфігурації – основний інструмент, з яким працює програміст-розробник. Дерево об'єктів конфігурації містить в собі практично всю інформацію про те, з чого складається конфігурація і являє собою вікно, яке представляє всю конфігурацію у вигляді дерева, кожна гілка якого описує певну складову конфігурації. Кореневі гілки дерева об'єднують об'єкти конфігурації, логічно пов'язані між собою і мають загальне призначення, наприклад, довідники, документи, журнали документів, перелічення і т.д. Всі елементи дерева конфігурації є складовою комплексної конфігурації системи 1СП.

Всі об'єкти конфігурації можна розділити на три основні групи:

1) Загальні об'єкти. Група допоміжних об'єктів конфігурації, за допомогою яких здійснюється створення конфігурації, механізмів взаємодії користувачів із обліковими даними.

2) Прикладні об'єкти. Їх перелік можна побачити на першому рівні дерева метаданих (виключаючи групу «Загальні»).

3) Підпорядковані об'єкти. До таких об'єктів відносяться «Реквізити», «Табличні частини» і т.д.

До прикладних об'єктів відносяться об'єкти наступних видів:

- **Константи.** Призначені для зберігання постійних, умовно-постійних величин.

- **Довідники.** Списки однорідних елементів даних. Використовуються для зберігання нормативно-довідкової інформації.

- Плани видів характеристик. Призначені для опису множини однотипних об'єктів аналітичного обліку.
- Документи. Служать для введення інформації про здійснювані операції в системі.
- Журнали документів. Служать для відображення списків документів різних видів.
- Загальні об'єкти. Група допоміжних об'єктів конфігурації, за допомогою яких здійснюється створення конфігурації, механізмів взаємодії користувачів з обліковими даними.
- Перелічення. Списки значень, що задаються на етапі конфігурування.
- Плани видів розрахунку. Призначені для опису множини однотипних об'єктів механізмів розрахунку.
- Звіти. Засіб отримання вихідної інформації.
- Обробки. Використовуються для виконання різних дій над інформаційною базою.
- Плани рахунків. Сукупність синтетичних рахунків.
- Регістри відомостей. Служать для зберігання інформації, склад якої розгорнутий по певній комбінації значень і при необхідності розгорнутий в часі.
- Регістри накопичення. Служать для накопичення інформації про наявність і рух засобів.
- Регістри розрахунків. Служать для накопичення інформації про періодичні розрахунки.
- Регістри бухгалтерії. Використовуються для віддзеркалення в бухгалтерському обліку інформації про господарські операції.
- Бізнес-процеси. Використовуються для реалізації «процесного» принципу роботи. Даний принцип дозволяє автоматизувати процес проходження і контролю ланцюгів подій, операцій.
- Задачі. Сумісно з бізнес-процесами реалізують процесний принцип. Вони дозволяють вести облік завдань по виконавцям і слугують відображенням просування бізнес-процесів по точкам маршруту.

Залежно від виду об'єкту конфігурації, об'єкт може мати різні підлеглі групи об'єктів. Склад підлеглих об'єктів залежить від типу об'єкту. Для об'єкту «План видів розрахунку» склад підлеглих об'єктів представлено на рис.5.

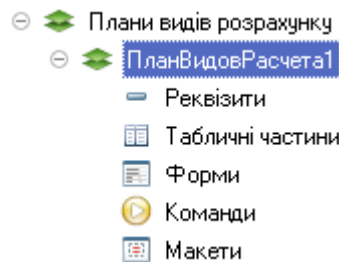


Рис. 5. Склад підлеглих об'єктів до об'єкту «План видів розрахунку»

Можливі підлеглі об'єкти:

- Реквізити – додаткова інформація про об’єкт, доступна тільки в межах цього об’єкту.
- Табличні частини – набори додаткової інформації про об’єкт, представлені у вигляді таблиць.
- Реквізити табличних частин – склад табличної частини об’єкту, доступні тільки в межах табличної частини об’єкту.
- Форми – використовуються для введення, перегляду і редагування інформації.
- Макети – табличні документи, призначені для формування друкованих форм об’єкту.
- Графи – графи журналу документів.
- Вимірювання – для реєстрів це об’єкти конфігурації, в розрізі яких враховуються дані в реєстрі.
- Ресурси – дані, що враховуються в реєстрі.

Типи даних в системі ІСП

Розрізняють три основні групи типів даних в ІСП:

- 1) Примітивні типи.
- 2) Призначені для користувача типи.
- 3) Інші типи, що не відносяться до примітивних і «призначених для користувача», але підтримка, яких присутня у вбудованій мові спочатку.

До примітивних типів даних відносяться: Число (десятькове число), Рядок (рядок фіксованої, або необмеженої довжини), Дата (дата, час) та Булево. Окрім вище-перелічених існує ще ряд типів, які відносяться до примітивних: це «Тип», «Не визначено», «Null». Існує також універсальний тип «Сховище значень». Якщо визначити реквізиту такий тип, то в ньому можна зберігати «все що завгодно» (включаючи двійкові дані, картинки, файли).

Примітивні типи даних спочатку визначені в системі, і їх набір обмежений.

Поряд з примітивними в будь-якій конфігурації типами можуть існувати типи даних, які визначаються тільки конкретною конфігурацією. Тобто такі типи, які не присутні в конфігурації постійно, а з’являються в результаті того, що додані деякі об’єкти конфігурації.

Наприклад, після того як ми створили об’єкт конфігурації довідник «Склади», відразу ж з’явилося декілька нових типів даних, пов’язаних з цим довідником. Серед них, наприклад, СправочникСсылка.Склади. І якщо тепер ми вкажемо для будь-якого реквізиту цей тип даних, то зможемо зберігати в ньому посилання на конкретний об’єкт довідника «Склади».

Об’єкти конфігурації, які можуть утворювати нові типи даних, називаються типу-утворюючими (призначеними для користувача). Слід ще раз зазначити, що ці типи даних не підтримуються платформою спочатку і існують тільки в конкретному прикладному рішенні.

Вбудована програмна мова системи

Мова є предметно-орієнтованою. Вона підтримує спеціалізовані типи даних предметної області, що визначаються конфігурацією системи. Робота з цими типами даних в мові організована з використанням об'єктної техніки.

Оскільки система поєднує в собі візуальні і мовні засоби конфігурування, використання вбудованої мови в системі має подіє-залежну орієнтацію, тобто мовні модулі використовуються в конкретних місцях для обробки окремих алгоритмів, що налаштовуються в процесі конфігурації.

Програмний код поміщається в «модулі». Місце розміщення конкретного програмного модуля надається конфігуратором в тих точках конфігурації, які вимагають опису специфічних алгоритмів функціонування. Ці алгоритми слід оформляти у вигляді процедур або функцій, які будуть викликані самою системою в наперед передбачених ситуаціях.

Існують модулі наступних видів:

1) Модуль додатку. Модуль розташовується в кореновому розділі конфігурації. В ньому розташовуються процедури-обробники подій, які ініціалізуються при старті і закінченні роботи системи, визначення (з ключовим словом «Експорт») змінних, процедур, функцій доступних в будь-яких точках конфігурації. В ньому не рекомендується реалізовувати процедури, функції, що виконують обробку даних (необхідні розрахунки).

2) Модуль зовнішнього з'єднання. В модулі можуть розташовуватися змінні, що експортуються, процедури і функції, а також процедури-обробники подій «ПриНачалеРаботыСистемы()» і «ПриЗавершенииРаботыСистемы()», що використовуються в режимі зовнішнього з'єднання.

3) Загальні модулі. Розташовуються в окремій гілці дерева метаданих. Можуть бути розбиті по підсистемах і містять процедури і функції. Ті з них, які зазначені з використанням ключового слова «Експорт» доступні з усіх модулів конфігурації.

4) Модулі прикладних об'єктів. Модулі розташовуються у вітках конфігурації, в яких містяться самі об'єкти (до них відносяться довідники, документи, звіти, обробки, реєстри) і є властивостями цих об'єктів.

5) Модулі набору записів. Модулі присутні у реєстрах будь-якого виду. В них можуть бути розташовані напередзадані процедури «ПриЗаписи», «ПередЗаписью».

6) Модулі форм. Ці модулі містяться у формах конфігурації. Модуль форми може містити змінні, процедури, функції, що реалізують алгоритми «поведінки» форми.

Кожний програмний модуль пов'язаний з іншими частинами конфігурації. Цей зв'язок називається контекстом виконання модуля.

Розрізняють два види контексту:

- Глобальний контекст. Утворюється із значень властивостей і методів глобального контексту функціями вбудованої мови і мовними конструкціями, змінними, процедурами і функціями програмного модуля додатку,

процедурами і функціями загальних модулів, оголошених за допомогою ключового слова «Експорт».

- Локальний контекст модуля. Утворюється тим конкретним місцем конфігурації, для якого використаний програмний модуль. Локальний контекст визначає набір доступних тільки даному модулю об'єктів.