1. Що таке інформаційне суспільство?

*Інформаційне суспільство* – постіндустріальне суспільство, в якому головними продуктами виробництва є інформація та знання, і більша частина працюючих зайнята обробленням інформації.

1. Дайте визначення інформатизації.

Інформатизаціяекономіки – це процес її інтенсифікації на базі сучасних ІТ та створення індустрії інформаційних послуг.

інформатизація *−* це більш широкий підхід порівняно з комп'ютеризацією, що містить як створення відповідної технічної бази, так і модернізацію організаційно-економічних, юридичних і “людських” чинників.

1. Дайте визначення комп’ютеризації.

Комп’ютеризація – процес розвитку та впровадження комп’ютерів, що забезпечують автоматизацію інформаційних процесів та технологій.

*Комп'ютеризація* – процес забезпечення окремих людей та виробничих колективів комп'ютерною і телекомунікаційною технікою та відповідним програмним забезпеченням.

Інформатизація та комп'ютеризація *−* це різні поняття. Комп’ютеризація *−* це цілісний процес формування нового автоматизованого інформаційного середовища для ефективного застосування наслідків інформатизації, а інформатизація *−* це більш широкий підхід порівняно з комп'ютеризацією, що містить як створення відповідної технічної бази, так і модернізацію організаційно-економічних, юридичних і “людських” чинників.

1. Що таке єдиний інформаційний простір?

*Єдиний інформаційний простір*  – це сукупність інформації, технологій її обробки, збереження та передачі, що функціонують на основі єдиних принципів і за спільними правилами.

1. Що таке технологія? Що таке інформаційна технологія? Яку структуру має інформаційна технологія?

*Технологія* — це комплекс наукових та інженерних знань, реалізованих у матеріальних, технічних, трудових факторах виробництва, способах їх поєднання для створення товарів та послуг з певними визначеними вимогами.

Згідно з визначенням ЮНЕСКО *інформаційні технології (ІТ)* − це комплекс взаємопов’язаних наукових, технологічних, інженерних дисциплін, що вивчають методи ефективної організації праці людей, зайнятих обробкою та зберіганням інформації, обчислювальну техніку, методи організації взаємодії з людьми та виробничим обладнанням, їх практичне застосування, а також пов’язані з цим обробленням соціальні, економічні та культурні проблеми.

1. Що таке інформаційна система? Яку структуру має інформаційна система ?

*Інформаційна система* – це сукупність засобів збору, зберігання, передачі, оброблення інформації в певній ПрО для досягнення поставленої мети в процесі управління.

Автоматизована ІС – сукупність інформації, різних методів і моделей, апаратних, програмних, організаційних, технологічних засобів і відповідних фахівців. Отже, інформаційна система – це організаційно впорядкована сукупність фахівців, інформаційних ресурсів та інформаційних технологій, зокрема з використанням засобів обчислювальної техніки і зв’язку, що реалізовують інформаційні процеси – отримання вхідних даних; обробку цих даних і/або зміну власного внутрішнього стану (внутрішніх зв’язків/відносин), видачу результату або зміну свого зовнішнього стану (зовнішніх зв'язків/відносин). За допомогою ІС надається можливість встановлення зв'язку між всіма елементами бізнес-процесів підприємства, що покращує можливості планування, контролю і регулювання процесів. Інформаційна система, при формуванні якої використаний принцип зворотного зв'язку на всіх рівнях управління і сучасні ІКТ, забезпечує зв'язок між елементами системи управління і елементами бізнес-процесів, тобто між всіма етапами прийняття рішень, також надає можливість накопичення даних, аналізу і моделювання.

1. Дайте основні види забезпечення інформаційної системи.

Автоматизована інформаційна система підприємства вирішує наступні завдання:

1) фінанси: облік і амортизація основних засобів, розрахунки з кредиторами і дебіторами, планування і здійснення платежів, формування актів звірок та іншої звітності, облік передоплати, формування прогнозу надходжень, обробка банківської виписки, формування управлінської, бухгалтерської та податкової звітності;

2) управління персоналом: підбір і наймання персоналу, підтримка організаційних структур, штатного розкладу, облік вакансій, ведення персональних даних про спіробітників, ведення і розрахунок табеля, підтримка бюджету і розрахунок заробітної плати, адміністрування заходів щодо проведення навчання;

3) постачання і логістика: формування і затвердження заявок на придбання товарно-матеріальних цінностей і послуг, ведення контрактів з постачальниками і їх аналізом, формування кредиторських рахунків-фактур щодо товарно-матеріальних цінностей, реєстрація надходжень в цехи і на центральні склади, планування і облік руху товарно-матеріальних цінностей і складских залишків, проведення загальної і часткової інвентаризації запасів.

4) збут: ведення контрактів з покупцями, ведення замовлень на продаж і планування відвантажень, формування пакету відвантажувальних документів і відвантаження товарно-матеріальних цінностей, автомаматичне формування дебіторських рахунків-фактур зщодо відвантаженої продукції;

5) планування: формування прогнозу продаж і загальних графіків попиту, планування виробництва і випуск виробничих завдань з урахуванням доступних виробничих потужностей, автоматичне формування заявок на придбання. Використовуються різні методи планування: потреба для виробництва, страховий запас, мінімаксне планування.

6) Основне і допоміжне виробництво: ведення виробничих специфікацій, технологічних карт, замовлень на технологічні зміни; крізний облік замовлень, планування виробництва і закупівель; випуск виробничих завдань, оцінка виробничої діяльності підприємства у режимі реального часу, реєстрація всіх параметрів якості і результатів випробувань, облік незавершеного виробництва; відстежування виконання кожної виробничої операції, контроль норм і облік фактичного використання виробничих ресурсів, персоналу, а також витрачання материалів на виробництво; розрахунок собівартості виготовленої продукції;

7) управління активами підприємства: ведення ієрархій активів для устаткування основних цехів; ведення специфікацій і технічних карт на регламентні і циклічні роботи, планування регламентних робіт; облік всіх видів обслуговування, зокрема поточного і капітального ремонту, реконструкції і модернізації; контроль витрачання матеріалів і запчастин; облік ресурсів (обладнання, персонал, інструмент); калькуляція собівартості замовлень на роботи; формування звітності.)

1. Що є предметом ітології?

Упродовж останнього десятиліття відбулося становлення нової науки, що вивчає ІТ, − *ітології*. Її предмет – ІТ та процеси їх створення та застосування. Ітологія – це така ж фундаментальна наука, як філософія та математика.

1. Що таке середовище відкритих систем?

Середовище відкритих систем (ВС) − вичерпний і узгоджений набір міжнародних стандартів ІТ та профілів функціональних стандартів, що описують інтерфейси, послуги і формати, щоб забезпечити інтероперабельність і мобільність застосувань, даних та персоналу, описують середовище, яке надає відкрита система для використання.

1. Які існують критерії класифікації інформаційних технологій?

територіальною ознакою або сферою діяльності – державні, територіальні (регіональні), галузеві, об’єднань, підприємств або установ;

за призначенням – інформаційно-пошукові, інформаційно-довідкові, інформаційно-керуючі, системи підтримки прийняття рішень, інтелектуальні ІС;

ступенем централізації обробки інформації – централізовані ІС, децентралізовані ІС, розподілені інформаційні системи;

ступенем інтеграції функцій – багаторівневі з інтеграцією за рівнями управління (підприємство – об'єднання, об'єднання – галузь тощо), багаторівневі ІС з інтеграцією за рівнями планування тощо;

видом обробки інформації – фактографічні, документальні, документально-фактографічні, мультимедійні, текстові тощо;

оперативністю обробки інформації – система реального часу, система оперативної обробки транзакцій, система пакетної обробки;

видами діяльності – система автоматизованого проектування, автоматизовані ІС, автоматизовані системи управління технологічними процесами, корпоративні ІС.

1. Назвіть поширені класифікації інформаційних систем.

За рівнем або за сферою діяльності: державні, територіальні (регіональні), галузеві, підприємств або організацій, технологічних процесів.

За призначенням або особливістю об'єктів управління: адміністративні, виробничо-технічні, соціальні,транспортні тощо.

За рівнем автоматизації процесів управління: інформаційно- пошукові, інформаційно- керівні, інформаційно- довідкові, СППР, інтелектуальні ІС тощо.

За рівнем централізації обробки інформації: централізовані ІС, децентралізовані ІС, ІС колективного використання.

За рівнем керованого виробничого процесу: 1) для виробництва з дискретним процесом (машинобудування,приладобудування); 2) з неперервним

## Тести

1. Інформація – це:

в) це сукупність відомостей про матеріальний і духовний світ, про закономірності й тенденції його розвитку, які можна відтворювати шляхом передачі усним, письмовим або електронним способом.

2. Синтаксична адекватність – це:

б) відображає формально-структурні характеристики інформації і не зачіпає її смислового змісту.

4. Єдиний інформаційний простір – це:

а) сукупність інформаційних ресурсів інформаційно-економічних просторів суб’єктів економічної діяльності і технологій їх обробки, зберігання та передачі, інформаційних систем і телекомунікаційних мереж, які функціонують на основі єдиних принципів та загальних правил;

5. Інформаційна технологія – це:

а) це система методів, процесів та способів використання обчислювальної техніки і систем зв’язку для створення, збору, передачі, пошуку, оброблення та поширення інформації з метою ефективної організації діяльності людей;

1. Що таке економічна інформація?

*Економічна інформація* – це сукупність відомостей, що характеризує виробничі відносини в суспільстві та соціально-економічні процеси, які слугують для керування цими процесами та колективами людей у виробничій і невиробничій сферах.

Це економічні дані, що відображають через систему натуральних, трудових і вартісних показників характер планової та фактичної виробничо-господарської діяльності, причинні взаємозв’язки між системою управління та об’єктами управління. Економічна інформація буває біржовою, фінансовою, комерційною, статистичною тощо.

1. Які відомо види економічної інформації?

Економічна інформація буває біржовою, фінансовою, комерційною, статистичною тощо.

1. Що розуміють під структурою подання економічної інформації?

Особливість інформації, що описує бізнес-процеси підприємств, полягає в тому, що масиви інформації зберігаються у вигляді певних однакових структур. Приклади таких інформаційних об’єктів: товари, деталі, вузли механізмів, комплектуючі для машин. Для формалізації економічної інформації використовують такі структури:

* *списки -* простий в реалізації спосіб, але не передбачає класифікації. Пошук потрібного об’єкта відбувається перебором, що вимагає певного часу;
* *дерева -* об’єкти ієрархічно поділені за категоріями, що може мати підкатегорії. Проте цей підхід не передбачає альтернативного поділу за іншими ознаками (вагою, конфігурацією, середовищем існування тощо);
* *багатовимірні куби -* для всіх об’єктів існує один набір ознак класифікації. Такий підхід враховує всі можливі ознаки, проте має інший недолік: певні ознаки частини об’єктів несумісні з точки зору змістовності. Кількість категорій набагато більша при такому підході порівняно з класифікацієюоб’єктів за допомогою дерева. Цей підхід є основним при побудові постреляційних баз даних, сховищ даних, що ефективно та компактно зберігають дані у вигляді багатомірних кубів тощо.

1. Що називають класифікацією економічної інформації?

Класифікація − поділ множини об'єктів на підмножини за їх подібністю або відмінністю згідно з прийнятими методами і підходами.

1. Які існують методи класифікації економічної інформації?

Метод класифікації − це сукупність правил створення системи класифікаційних угруповань і їх зв'язків між собою.

Найбільш поширені методи класифікації об'єктів: ієрархічний, фасетний, дескрипторний. Ці методи відрізняються стратегією застосування класифікаційних ознак. Ієрархічний метод класифікації характеризується тим, що початкова множина об'єктів техніко-економічної інформації послідовно поділяється на класи першого рівня поділу, далі − на угруповання наступного рівня тощо.

1. Які існують методи кодування економічної інформації?

Порядкова (реєстраційна), Серійно-порядкова система, Послідовна система ,Паралельна система, **Штрихове кодування**

1. Розкрийте сутність єдиної системи класифікації та кодування техніко-економічної інформації.

Класифікації та кодуванню техніко-економічної інформації відводиться особлива роль, оскільки вони є засобами, що забезпечують взаємний обмін інформацією між людиною і ЕОМ. Кодування техніко-економічної інформації на основі системи класифікації дозволяє безпосередньо за кодом об'єкту стверджувати про його характеристики (конструкційні, технологічні, експлуатаційні).

Класифікація і кодування — це дві взаємодоповнюючі частини одного процесу − перетворення різноманітної економічної інформації з природної мови на формалізовану мову ЕОМ. У процесі згаданого перекладу вони виконують різні функції. Для їх поглибленого вивчення слід навести основні терміни й поняття, що використовуються в цій області.

## Тести

1. Джерелом внутрішньої економічної інформації є:

б) бухгалтерський облік і звітність;

2. Інформаційна база підприємства – це:

а) сукупність інформації про об’єкт управління, що використовується при функціонуванні ІС;

3. Елементи логічної структури даних − це:

а) реквізит;

4. Система кодування економічної інформації буває:

б) серійно-порядкова система;

5. Класифікація − це:

а) поділ множини об'єктів на підмножини за їх подібністю або відмінністю згідно з прийнятими методами і підходами;

1. У чому суть інженерної і наукової діяльності?

Мета інженерної діяльності полягає у створенні техніки, технології та ефективного їх використання в системі суспільного виробництва. Поза цим інженер позбавлений предмета своєї діяльності.

Науко́ва дія́льність — це інтелектуальна творча діяльність, спрямована на одержання та використання нових знань. Основними її формами є фундаментальні та прикладні наукові дослідження.

1. У чому специфіка програмної інженерії як інженерної діяльності?

Основні проблеми, що постають перед програмною інженерією, пов’язані з інтеграцією створеного раніше ПЗ у нові розробки (legacy challenge), роботою в розподіленому гетерогенному середовищі (heterogeneity challenge) та обмеженнями часу, що відводиться на розроблення інформаційних продуктів (delivery challenge).

*Основні роздiли програмної інженерії:*

* аналiз вимог до ІС, яку треба створити;
* детальний проект ІС;
* кодування;
* тестування системи;
* процес супроводження програмного продукту;
* керування конфiгурацiєю;
* забезпечення якостi розроблення;
* забезпечення вiдповiдностi розроблення вимогам її замовникiв та забезпечення вiдповiдностi кодiв проекту;
* процес удосконалення отриманого програмного продукту.

Еталонна модель програмної iнженерiї визначається взаємодiєю трьох факторiв: процесів, продуктів та ресурсів.

1. Який вигляд мають продукти програмної інженерії?

Усi продукти програмної iнженерiї являють собою певнi описи - тексти вимог до розроблення, узгодження домовленостей, документацiю, тексти програм, iнструкцiї щодо експлуатацiї тощо. Головні ресурси програмної iнженерiї, що визначають ефективнiсть розроблень, - це час та вартiсть.

1. Які бувають головні ресурси програмної інженерії, що визначають ефективність розробок програмного забезпечення?

Головні ресурси програмної iнженерiї, що визначають ефективнiсть розроблень, - це час та вартiсть.

1. Що таке життєвий цикл розробки програмного забезпечення?
2. *Життєвий цикл* ПЗ − певна послiдовнiсть фаз або стадiй вiд моменту прийняття рішення про необхідність створення ПЗ до повного вилучення ПЗ з експлуатації
3. Які бувають етапи процесу розробки програмного забезпечення?

* *основні* процеси (придбання, доставка, розроблення, експлуатація, супровід);
* *організаційні* процеси (управління, удосконалення, навчання);
* *допоміжні* процеси (документування, забезпечення якості, верифікація, атестація, аудит, загальна оцінка тощо).
* *процес придбання*, що iнiцiює життєвий цикл ІС та визначає її покупця;
* *процес розроблення*, що визначає дiї органiзацiї-розробника інформаційного продукту;
* *процес постачання*, що визначає дiї пiд час передачi розробленого продукту покупцю;
* *процес експлуатацiї*, що означає дiї з обслуговування системи пiд час її використання - консультацiї користувачiв, вивчення їхнiх побажань тощо;
* *процес супроводження*, що означає дiї з керування модифiкацiями, пiдтримки актуального стану та функцiональної придатностi, iнсталяцiї та вилучення версiй систем у користувача.

1. Що таке UML?

Уніфікована мова моделювання UML (Unified Modeling Language) є мовою для визначення, представленя, проектування і документування програмних систем, організаційно-економічних систем, технічних систем та інших систем різної природи. UML містить стандартний набір діаграм і нотацій найрізноманітніших видів.

UML − це наступник того покоління методів ООАП, які з'явилися в кінці 1980-х і на початку 1990-х років. Створення UML фактично розпочалося в кінці 1994 р., коли Граді Буч і Джеймс Рамбо почали роботу щод об'єднання їх методів Booch і OMT (Object Modeling Technique) під егідою компанії Rational Software. До кінця 1995 р. вони створили першу специфікацію об'єднаного методу, названого ними Unified Method, версія 0.8. Тоді ж в 1995 р. до них приєднався творець методу OOSE (Object-Oriented Software Engineering) Івар Якобсон. Таким чином, UML є прямим об'єднанням і уніфікацією методів Буча, Рамбо і Якобсона, проте доповнює їх новими можливостями. Головними в розробці UML були наступні цілі:

* надати користувачам готову до використання виразну мову візуального моделювання, що дозволяє їм розробляти осмислені моделі і обмінюватися ними;
* передбачити механізми розширюваності і спеціалізації для розширення базових концепцій;
* забезпечити незалежність від конкретних мов програмування і процесів розробки.
* забезпечити формальну основу для розуміння цієї мови моделювання (мова повинна бути одночасно точним і доступним для розуміння, без зайвого формалізму);
* стимулювати зростання ринку об'єктно-орієнтованих інструментальних засобів;
* інтегрувати кращий практичний досвід.

1. На якій фазі життєвого циклу розробки програмного забезпечення фіксується контракт між замовником і виконавцем розробки?

Етап придбання

1. Що таке верифікація інформаційного продукту?

*Верифікація* − перевірка відповідності реалізації системи специфікаціям результатів проектування й опису компоненти.

Верифікація (визначення того, що поведінка програми відповідає специфікації на цю програму).

1. Що таке валидація інформаційного продукту?

*Валідація* − перевірка відповідності створеного ПЗ потребам та вимогам замовника. Це дорогий процес, що забезпечує високу якість програмного коду. Валидація дозволяє підтвердити, що програмне забезпечення є коректною реалізацією початкових умов у системі й провадиться після завершення кожного етапу розроблення цього забезпечення.

1. У чому полягає супроводження інформаційних продуктів?

*Супроводження* − це роботи з внесення змін до ІС після того, як її було передано користувачеві для експлуатації. На відміну від обладнання, яке з часом потребує ремонту, ПЗ не "зношується", тому процес супроводження націлений на підтримку передовсім еволюціонування системи, тобто на зміну її функцій та властивостей.

1. Що таке помилка в інформаційному продукті?

*Помилка* − це стан програми, при якому генеруються неправильні результати.

1. Які бувають класи помилок у програмах?

ненавмисне відхилення розробників від робочих стандартів або планів реалізації;

специфікації функціональних та інтерфейсних вимог без дотримання стандартів розроблення;

недосконала організація процесу розроблення.

1. Хто входить до команди тестувачів?

Як правило, команда тестувачів не залежить від штату розробників ІС. Деякі члени цієї команди є досвідченими тестувачами або навіть професіоналами в цій галузі. До них належать аналітики, програмісти, інженери-тестувачі, котрі присвячують увесь свій час проблемам тестування систем.

1. За якими параметрами оцінюють якість ІС?

Функціональність, надійність, зручність застосування, ефективність, супроводжуваність

1. Що таке повторне використання в програмній інженерії?

*Повторне використання -* це використання для нових розроблень будь-яких фрагментів інформації, здобутих під час розроблення інших ІС.

## Тести

1. Модель життєвого циклу ПЗ є:

б) це структура, що визначає послідовність виконання і взаємозв'язки процесів, дій, задач протягом життєвого циклу;

2. Детальне проектування ПЗ включає такі задачі:

а) розроблення і документування програмних інтерфейсів ПЗ і БД;

в) розроблення і документування попередніх вимог до тестів і плану інтеграції ПЗ.

3. Структурний підход проектування систем використовує:

б) UML-технологію;

4. Верифікація − це :

б) перевірка відповідності реалізації системи специфікаціям результатів проектування й опису компоненти;

5. *Техноробочий проект* − це:

в) Жодне з перерахованих.

(*Технічний проект* + *Робоча документація* = *Техноробочий проект*)

1. Що таке інформаційна технологія?

*Інформаційна технологія* – сукупність економіко-математичних методів, певних методик, програмно-технічних засобів для зберігання, передачі, обробки інформації в певній предметній області для досягнення поставленої мети.

1. У чому полягає відмінність між процесом комп'ютеризації та інформатизації?

Інформатизація та комп'ютеризація *−* це різні поняття. Комп’ютеризація *−* це цілісний процес формування нового автоматизованого інформаційного середовища для ефективного застосування наслідків інформатизації, а інформатизація *−* це більш широкий підхід порівняно з комп'ютеризацією, що містить як створення відповідної технічної бази, так і модернізацію організаційно-економічних, юридичних і “людських” чинників.

1. Назвіть складові інформаційійної технології.

цілеспрямованість;

доцільність;

наявність компонентів та структури;

взаємодія із зовнішнім середовищем;

системна повнота;

регулярність процесів;

динамічність.

1. Що таке автоматизоване робоче місце?

*Автоматизоване робоче місце* (АРМ) − це програмно-технічний комплекс, що забезпечує автоматизацію функцій його діяльності, поєднуючи комплекс технічних, програмних, інформаційних та інших засобів.

1. Поясніть сутність розподіленої інформаційної системи.

Це цифрова інформаційна середа, функціональні об’єкти якої при взаємодії з керуючими комп’ютерами розбиваються за погодженими каналами у відповідності із закладеним алгоритмом. Робочими елементами інфраструктури виступають мережі, а під об’єктами розуміються інформаційні повідомлення, одиниці даних і технологічних матеріалів.

1. У чому полягає підхід композиції сервіс-орінтованих прикладних застосувань?

Це парадигма, що призначена для проектування, розробки і управління застосуваннями у мережевому середовищі. Перехід до СОА відкриває можливість для ефективної реалізації розподілених обчислень у мережі Internet. Цей стандарт дає змогу вирішувати інтеграційні розподілені задачі.

Сервіс-орієнтірована архітектура допомагає ІТ-підрозділам підприємств ефективно вирішувати нові завдання, що постають перед ними. СОА призначена для інтеграції застосувань за допомогою “зв'язування” послуг без написання нового програмного коду. Бізнс-система може вчасно реагувати на зміни умов середовища шляхом налагодження процесу обміну повідомленнями в автоматизованому режимі.

СОА перетворилася на практичну методологію створення інформаційних систем. За допомогою засобів, що входять в SOA Foundation (IBM), сучасні підприємства можуть оперативно створювати рішення щодо взаємодії динамічних систем (DS) на основі СОА. Це набір інтегрованого програмного забезпечення, заснований на відкритих стандартах і складається з продуктів, що входять в сімейство програмного забезпечення проміжного рівня WebSphere.

Головна перевага СОА полягає в тому, що сервісна архітектура може бути використана як засіб для відображення бізнес-процесів на програмно-технічні ресурси і в подальшому забезпечувати розподілену підтримку даних і постійний реінжиніринг бізнес-процесів.

1. Розкрийте поняття Internet-, Extranet-, Intranet-технології.

Мережа Internet − універсальна база даних і знань. Технологія реалізації мережі Internet є перспективною для реалізації обміну даними усередині корпоративних мереж підприємства.

**Екстране́т** ([англ.](https://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) *extranet*) — це захищена від несанкціонованого доступу [корпоративна мережа](https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D1%80%D0%BF%D0%BE%D1%80%D0%B0%D1%82%D0%B8%D0%B2%D0%BD%D0%B0_%D0%BC%D0%B5%D1%80%D0%B5%D0%B6%D0%B0), що використовує [Інтернет-технології](https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82-%D1%82%D0%B5%D1%85%D0%BD%D0%BE%D0%BB%D0%BE%D0%B3%D1%96%D1%97) для внутрішньокорпоративних цілей, а також для надання частини корпоративної інформації і корпоративних [застосунків](https://uk.wikipedia.org/wiki/%D0%97%D0%B0%D1%81%D1%82%D0%BE%D1%81%D1%83%D0%BD%D0%BE%D0%BA) діловим партнерам компанії.

Переваги технології Intranet: простота в пошуку інформації і навігація; зручне і компактне подання інформації; наявність великої кількості служб; можливість електронного документообігу усередині підприємства; розміщення реклами на сайті підприємства; створення корпоративного Web-сервера.

Технологія Intranet − це створення локальної інформаційної системи клієнт-серверної архітектури з урахуванням обмежень: протоколи обміну − HTTP і FTP (File Transfer Protocol, протокол передачі файлів), основна форма подання інформації − HTML.

Intranet передбачає високу пропускну здатність каналів зв'язку між клієнтом й сервером і використання як стандартних серверів і клієнтів (HTTP-сервер і браузер), так і стандартних механізмів розширення можливостей системи, наприклад CGI.

## Тести

1. Інформатизація − це:

б) організований соціально-економічний і науково-технічний процес створення оптимальних умов для задоволення інформаційних потреб та інтенсифікації економіки на основі впровадження ІТ, створення передумов переходу до інформаційного суспільства;

2. Інформаційна послуга – це:

а) задоволення потреб споживача в інформаційних потребах;

1. Що називають проектом?

*Проектом* називають захід, призначений для створення унікальних продуктів або послуг, що має конкретну мету, для якого встановлений термін реалізації плану і визначений бюджет.

*Проект −* цілеспрямоване, заздалегідь пророблене і заплановане створення або модернізація фізичних об'єктів, технологічних процесів, технічної й організаційної документації для них, матеріальних, фінансових, трудових й інших ресурсів, а також управлінських рішень і заходів щодо їхнього виконання.

*Проект* – це набір документів, розташованих в логічній послідовності, в яких описується розробка бізнес-ідеї, її реалізація та експлуатація до моменту досягнення мети інвестора.

1. Що таке життєвий цикл проекту?

**Життевий цикл проекту**(ЖЦП) — проміжок часу від моменту появи, зародження **проекту** і до моменту його ліквідаці, завершення.

1. Які Ви знаєте автоматизовані системи управління ІТ-проектом?

це Microsoft Project, Primavera Project Planner, Time Line 6.5, Artemis Views, Spider Project, Open Plan.

1. Які Вам відомо програмні продукти управління проектами?

Project Expert, Інвестор фірми ІНЕК, Альт-інвест фірми Альт (Санкт-Петербург), FOCCAL фірми Центрінвестсофт, ТЕО-ІНВЕСТ Інституту проблем управління РАН; COMFAR (Computer Model for Feasibility Analysis and Reporting) і PROPSPIN (Project Profile Screening and Pre-appraisal Information system), створені в UNIDO − Організації Об'єднаних Націй по промисловому розвитку.

1. У чому полягає особливість програми COMFAR ?

Методичною основою даної програми є методика Організації по промисловому розвитку ООН (так звана методика ЮНІДО). Запропонований фахівцями ЮНІДО підхід до побудови типового бізнес-плану дозволяє не впустити суттєвих моментів в описі поточної або планованої діяльності підприємства, а також представити результати у вигляді, що найбільш відповідають для сприйняття західними фінансистами. Дана методика виконує роль єдиної бази, що дозволяє спілкуватися між собою фахівцям у сфері інвестиційного проектування і фінансового аналізу. Більшість відомих на даний момент комп'ютерних систем для інвестиційного проектування опираються на цю методику, і, зокрема, всі розглянуті вище вітчизняні програми. Програма COMFAR є практичною реалізацією цієї методики у вигляді комп'ютерної програми, яка може бути придбана в представництві ЮНІДО.

Програма є універсальним інструментом для опису і розрахунку всіх основних етапів інвестиційного проектування. Інвестиційна діяльність може бути детально описана завдяки наявності кількох самостійних розділів. Програма дозволяє використовувати кілька способів нарахування амортизації. Опис операційної діяльності складається з дуже докладного опису витрат виробництва, які можуть бути віднесені на номінальну потужність виробництва або одиницю продукції. Програма продаж розраховується по кожному виду продукції. Причому витрати виробництва можна розрахувати як для кожного виду продукції, так і в цілому по підприємству. Проте складні схеми формування запасів матеріальних ресурсів і схем їх оплати, що часто зустрічаються в практиці, нестандартні схеми продаж проведеної продукції і ряд інших ситуацій не можуть бути адекватно описані з урахуванням прийнятих в програмі допущень. В умовах перехідної економіки ці негативні моменти утрудняють використання цієї програми, особливо для поточного планування. Фінансова діяльність включає опис акціонерного і венчурного капіталу, довгострокових і короткострокових позик, розподіл прибутків.

## Тести

1. Управління проектами розглядається як:

б) універсальна мова спілкування між учасниками проекту;

2. Проектний цикл – це:

в) проміжок часу між моментом появи проекту і моментом його

закінчення.

3. Автоматизована система управління ІТ-проектом призначена:

а) для інформаційної підтримки інфраструктури управління;

4. Щоб переглянути розподілені за часом дані проекту у програмному продукті Office Project Standard 2007, необхідно:

в) усі відповіді невірні.

5. Яку програму можна рекомендувати консалтинговим фірмам як основу для розробки індивідуальної моделі функціонування підприємства?

а) Альт - Інвест;

**Розділ №6**

Контрольні питання

1. **Що являє собою концепція планування потреби виробництва в матеріальних ресурсах**

MRP – концепція планування потреби виробництва в матеріальних ресурсах, яка для визначення потреб використовує інформацію про структуру і технологію виробництва кінцевого продукту, календарний план виробництва, дані складських запасів, договорів поставки матеріалів і комплектуючих тощо.

1. **Опишіть концепцію управління виробничим підприємством, що базується на взаємопов’язаному плануванні виробничих потужностей, потребі в матеріалах, фінансах і кадрах.**

MRPII (Manufacture Resources Planning II, планування виробничих ресурсів) – концепція управління виробничим підприємством, заснована на взаємопов’язаному плануванні виробничих потужностей, потребі в матеріалах, фінансах і кадрах.

MRPII методологія підтримує оперативне планування продукції і матеріалів в натуральних одиницях, фінансове планування у вартісних одиницях.

1. **Що таке ERP-система?**

Інтегровані рішення за назвою «комплексне планування ресурсів підприємства» (ERP) покращують якість прийняття рішень, продуктивність і прибутковість. ERP-системи дозволяють ефективно планувати комерційну та виробничу діяльність підприємства.

1. **До якого типу систем належить “1С:Предприятия 8.0”**

До ERP.

1. **Перелічіть найпоширеніші промислові інформаційні системи підприємств.**

***System21,*** *iRenaissance,* ***Microsoft Dynamics AX, Microsoft XAL,*** *Oracle E-Business Suite*, ***mySAP Enterprise Resource Planning (mySAP ERP), 1С:Предприятие 8.0,*** SIKE, ***Галактика Business Suite,*** *Business Control*, *Віртуоз,* ***infor:COM,*** *Smart RetailSuiteTM* , ***SyteLine ERP , Microsoft Navision*,** BAAN−виробництво, ***IFS Applications, Millennium ERP* ,** *ERP “NOVA”*

1. **У чому полягає специфіка ERPII-систем?**

Поява концепції ERPII викликана тим, що для сучасних ERP-систем став характерний розвиток нових функціональних можливостей, пов'язаний з виходом за традиційні рамки оптимізації й автоматизації транзакційних процесів усередині підприємства.

Розширення сфери застосування ERPII виявляється у тому, що нова концепція охоплює також невиробничі сфери. Властиві їй функції відображають тепер і специфіку конкретної галузі або напряму діяльності.

ERP II принципи:

* інформаційне забезпечення всіх процесів в їх взаємозв'язку;
* одноразове введення даних в місці їх виникнення з прямим застосуванням у всіх функціональних напрямах;
* застосування єдиної бази даних, правил і процедур;
* можливості безперервного розвитку відносно нових корпоративних управлінських методів і технологій, комунікаційних і технічних засобів, інформаційного і функціонального обхвату.

1. **Що таке технологія ASP?**

*ASP* (Application Service Providing/Provision) – це технологія використання елементів інформаційних технологій на умовах орендної плати(не плутати з Active Server Pages (ASP) − технологіею Microsoft для Windows систем, що дозволяє створювати динамічні Web-сторінки).

1. **У чому полягає сутність CSRP-систем?**

CSRP-система − інтегрована електронна інформаційна система управління, що реалізовує концепцію CSRP. Призначення CSRP − створення товарів з підвищеною цінністю для покупця, тобто продуктів, які повністю задовольняють специфічному набору вимог кожного конкретного покупця.

Тести

1. MRPII методологія це :

а) планування потреб у сировині і матеріалах для виробництва;

б) управління корпоративними ресурсами;

в) планування і управління всіма виробничими ресурсами підприємства.

1. Microsoft Axapta це:

а) комплексне ERP-рішення;

б) ERPII-рішення;

в) MRPII-рішення.

1. Oracle E-Business Suite 11i включає функціональні блоки:

а) Oracle ERP і Oracle CRM ;

б) Oracle E-Hub і Oracle SCM ;

в) жодне з перерахованих.

1. Інтегрована система “Технокласс це :

а) комплексне ERP-рішення;

б) ERPII-рішення;

в) CSRP-рішення.

1. Стандарт, що не базується на використанні XML, це:

а) Open-EDI Reference Model;

б) BizTalk Framework ;

в) RosettaNet.

**Розділ №7**

Контрольні питання

1. **Які фактори впливають на прийняття управлінських рішень?**

1) **Ступінь ризику** (величина ймовірності, що характеризує можливість невиконання системою своєї цільової задачі з урахуванням впливу небезпечних внутрішніх і зовнішніх дій на систему), у даному випадку розуміємо, що завжди існує ймовірність прийняття неправильного рішення, яке може несприятливо впливати на організацію.

2) **Час**, який відводиться менеджерові для прийняття рішення. На практиці більшість керівників не мають можливості проаналізувати усі можливі альтернативи, відчуваючи дефіцит часу.

3) Ступінь підтримки менеджера колективом – цей фактор враховує те, що нових менеджерів сприймають не відразу. Якщо порозуміння й підтримки інших менеджерів і підлеглих не достатньо, то проблему слід усувати за рахунок своїх особистих рис, які повинні сприяти виконанню прийнятих рішень.

4) Особисті **якості менеджера** – один з найбільш важливих факторів. Незалежно від того, як менеджери приймають рішення і відповідають за них, вони повинні мати здібності до того, щоб приймати правильні рішення.

5) **Політика** організації – суб’єктивний фактор при прийнятті рішення. Статус, влада, престиж, легкість виконання – усе це може вплинути на прийняття певного рішення.

1. **У чому полягають особливості поведінкової моделі Г.Саймона?**

Поведінкова модель має такі основні характеристики: **1) ОПР не має повної інформації щодо ситуації прийняття рішення або всіх можливих альтернатив; 2) ОПР не здатна передбачити наслідки реалізації кожної можливої альтернативи**. Саме почуття характеризує суб’єктивний характер прийняття рішень, це знаходить своє відображення у перевагах ОПР.

Враховуючи ці особливості Г.Саймон сформулював два основних поняття поведінкової моделі:

**1) поняття “обмеженої раціональності**”, яке означає, що люди можуть тільки намагатися прийняти раціональне рішення, але їх раціональність завжди є обмеженою, тобто теоретично завжди існує рішення краще за прийняте;

**2) поняття “досягнення задоволеності”.** Оскільки досягти “повної раціональності” неможливо, менеджери розуміють під цим вибір, який є достатньо добрим за конкретних умов.

1. **Охарактеризуйте модель раціональної технології прийняття рішень.**

Модель поділяється на 4 етапи: **I етап.** Діагноз проблеми, **II етап.** Накопичення інформації про проблему, **III етап.** Розробка альтернативних варіантів, **IV етап.** Оцінка альтернативних варіантів, **V етап.** Прийняття рішення.

1. **Дайте загальноприйняту класифікацію СППР.**

Залежно від специфіки розв'язуваних задач і використовуваних технологічних засобів процесу створення систем можна виділити такі СППР: спеціалізовані СППР; СППР-генератори; СППР-інструментарій.

1. **Опишіть типову архітектуру СППР.**

З функціональної точки зору СППР включає такі компоненти: сервер сховища даних; інструментарій OLAP; інструментарій Data Mining.

У підсистемі введення даних OLTP (Online transaction processing) реалізується операційна (транзакційна) обробка даних. Для їх реалізації використовують звичайні системи управління базами даних (СУБД).

У підсистемі зберігання інформації використовують сучасні СУБД і концепцію сховищ даних. Концепція сховища даних передбачає розділ структур зберігання даних для оперативної обробки даних і виконання аналітичних запитів.

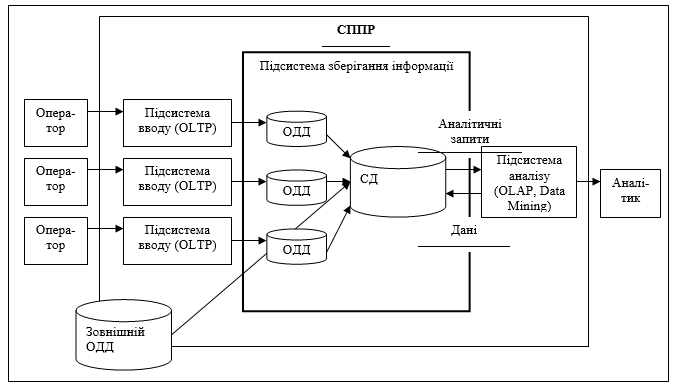
Підсистема аналізу може включати:

1) підсистему інформаційно-пошукового аналізу на базі реляційних СУБД і статичних запитів з використанням мови SQL (Structured Query Language);

2) підсистему оперативного аналізу. Для реалізації таких підсистем застосовується технологія оперативної аналітичної обробки даних OLAP (On-line analytical processing), використовуючи концепцію багатовимірного представлення даних;

3) підсистему інтелектуального аналізу. Дана підсистема реалізує методи і алгоритми здобуття даних Data Mining.

1. **Яка структура СППР на основі сховищ даних?**



1. **Дайте основні поняття технології оперативного аналітичного оброблення даних OLAP.**

У випадку використання сховища даних технологія OLAP забезпечує доступ до даних в термінах, звичних для аналітика.

Виконання операцій над даними здійснюється OLAP-машиною. За місцем розміщення OLAP-машини поділяються на OLAP-клієнти і OLAP-сервери. OLAP-клієнт проводить побудову багатовимірного куба і обчислення на клієнтському ПК, а OLAP-сервер одержує запит, обчислює і зберігає агрегатні дані на сервері, видаючи тільки результати.

1. **Які сучасні тенденції розвитку СППР?**

Останніми роками технологія рішення інформаційних задач значно змінилася. Інтенсивно розвивається і упроваджується нова інформаційна технологія рішення задач управління. Нова інформаційна технологія ставить своєю метою забезпечення простоти процесу взаємодії користувача з комп'ютером з виключенням необхідності регулярного супроводу.

Основна ідея нової технології, покликана забезпечити проблеми інтерпретації, полягає в тому, щоб розглядати систему понять ПрО і відповідність між нею і системою понять формальної моделі як початкову інформацію для вирішення прикладних завдань.

Нова інформаційна технологія грунтується перш за все на інтелектуальних технологіях і теорії штучного інтелекту.

При впровадженні систем прийняття рішень організаційна структура підприємства і його бізнес-процеси перебудовуються таким чином, що ефективність виробництва збільшується. Процесне управління, як спосіб радикального підвищення конкурентоспроможності бізнесу в сучасній, орієнтованій на клієнта економіці, завойовує все більше прихильників. Сама ідея процессного управління не нова, а сплеск інтересу до нього обумовлений появою нового інструментарію і методології під назвою BPM − Business Process Management.

Тести

1. Що входить до функції управління:

а) планування, організація і контроль;

б) контроль, мотивація і програмування;

в) жоден з перерахованих.

1. Слабоструктурованим проблемам притаманні такі особливості:

а) не існує багато варіантів рішень;

б) вимоги щодо вартості й часу визначаються повністю;

в) реалізація рішень пов'язана з ризиком щодо великих обсягів ресурсів.

1. За ступенем обробки даних при аналізі даних не виділяють задачі:

а) інформаційно-пошукові;

б) оперативно-аналітичні;

в) жодне з перерахованих.

**Розділ №8**

Контрольні запитання

1. **Що називають інтелектуальною інформаційною системою?**

**Інтелектуальна інформаційна система** (**ІІС**) – це один із видів [автоматизованих інформаційних систем](https://uk.wikipedia.org/w/index.php?title=%D0%90%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82%D0%B8%D0%B7%D0%BE%D0%B2%D0%B0%D0%BD%D0%B0_%D1%96%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D1%96%D0%B9%D0%BD%D0%B0_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0&action=edit&redlink=1), інколи ІІС називають системою, основану на знаннях. ІІС є комплексом програмних, лінгвістичних і логіко-математичних засобів для реалізації основного завдання: здійснення підтримки діяльності людини і [пошуку інформації](https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D1%96%D0%B9%D0%BD%D0%B8%D0%B9_%D0%BF%D0%BE%D1%88%D1%83%D0%BA) в режимі розширеного діалогу [природньою мов](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D1%80%D0%BE%D0%B4%D0%BD%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0)ою.

1. **Що таке експертна система**

*Експертна система* – це складна ІС, що оперує знаннями певної ПрО з метою надання рекомендацій або вирішення проблеми

1. **Які типи експертних систем Ви знаєте?**

ЕС DENDRAL, ЕС PROSPECTOR, *OMEGAMON*

1. **У чому полягає сутність інтелектуальних систем аналізу і звітності на підприємствах?**

Дані системианалізу дозволяють отримувати корисні для підприємства відомості з величезної кількості інформації, що зберігається в корпоративних базах даних.

1. **Дайте загальну характеристику BI-платформи MicroStrategy.**

* ВІ-платформа запропонована компанією *MicroStrategy*.
* обраний підхід ROLAP дозволив уникнути проблеми різкого збільшення обсягу даних, підтримує велику аналітичну функціональність.
* Для забезпечення масштабованості за кількістю користувачів компанія реалізувала систему Business Intelligence в триланковій конфігурації, тобто із сервером MicroStrategy Intelligence Server, а потім - у чотириланковій.
* MicroStrategy надає технології Business Intelligence рівня підприємства.

1. **Що таке OLAP-системи?**

*OLAP* (On-Line Analytical Processing) − технологія оперативної аналітичної обробки даних, що використовує методи і засоби для збору, зберігання і аналізу багатовимірних даних в цілях підтримки процесів прийняття рішень.

Основне призначення OLAP-систем − підтримка аналітичної діяльності, довільних запитів користувачів-аналітиків. OLAP є аналітичним інструментом і спочатку базувався на багатовимірних базах даних (ББД).

1. **У чому особливість BI-платформи-Cognos?**

Рішення Cognos у сфері Business Intelligence надає інформаційну інфраструктуру, що дозволяє кожному співробітникові підприємства краще відстежувати діяльність підприємства, − в будь-якому розрізі даних і з будь-яким ступенем деталізації. Cognos дозволяє зв’язати результати водночас у розподіленому інформаційному середовищі.

1. **Що таке інтелектуальний аналіз даних Data Mining?**

*Інтелектуальний аналіз даних* (ІАД, Data Mining) або розвідка даних − термін, що застосовується для опису здобуття знань в базах даних, дослідження даних, обробки зразків даних, очищення і збору даних. Це процес виявлення кореляції, тенденцій, шаблонів, зв’язків і категорій.

*Data mining* − це процес виявлення в необроблених даних раніше невідомих нетривіальних, практично корисних і доступних інтерпретації знань, необхідних для прийняття рішень у різних сферах діяльності

1. **Чим вирізняються технології Text Mining ?**

*Text Mining* − такий набір технологій і методів, призначених для здобуття інформації з текстів на основі сучасних ІКТ, що дозволяє проявити закономірності, які можуть приводити до отримання корисної інформації і нових знань користувачами

1. **Що називають агентно-орієнтованими системами?**

Новий напрям розвитку сучасних інтелектуальних інформаційних технологій пов'язаний з програмними агентами (ПА) – парадигмою програмування, що дозволяє перейти на більш інтелектуальний рівень взаємодії користувача з програмним і апаратним забезпеченням.

1. **Що таке віртуальне підприємство?**

*Віртуальне підприємство –* тимчасова або постійна сукупність географічно роз'єднаних осіб (фізичних/юридичних) з визначеними базовими компетенціями, економічна діяльність яких здійснюється при обов'язковій участі ІКТ в інформаційно-економічному просторі з метою отримання прибутку

1. **Що таке організаційна онтологія підприємства?**

Організаційна онтологія – інформаційна модель суб’єкта економічної діяльності, яку можуть використовувати інші СЕД в своїх ментальних моделях для планування взаємодії з цим СЕД.

Тести

1. Експертна система – це:

1. система, що здатна прогнозувати свою поведінку і реакцію середовища на свої дії із здатністю навчатися, тобто використовувати знання в нових, невідомих ситуаціях;
2. складна ІС, що оперує знаннями певної ПрО з метою надання рекомендацій або вирішення проблеми;
3. інтерактивна система, що забезпечує особам, які приймають рішення, засоби доступу до даних і моделей для вирішення проблем.

2. Бізнес-аналітика (ВІ) – це:

1. метазнання у вигляді набору евристик, які дозволяють знайти прийнятне рішення для певної підмножини вхідних даних;
2. результат процесу здобуття знань про ПрО для прийняття оптимізованих рішень;
3. організація розподіленого доступу кінцевих користувачів та аналіз структурованих кількісних даних і інформації.

3. OLAP– це:

* 1. cукупність засобів багатовимірного аналізу даних, накопичених у сховищі даних;
  2. предметно-орієнтоване, прив'язане до часу і незмінне зібрання даних для підтримки процесу управлінських рішень;
  3. оперативна обробка транзакцій.

4. Data mining – це:

1. алгоритми, що дозволяють знайти задовільне рішення для аналітично нерозв’язуваних проблем через послідовний підбір і комбінування параметрів з використанням механізмів, що нагадують біологічну еволюцію;
2. процес виявлення в необроблених даних раніше невідомих нетривіальних, практично корисних і доступних інтерпретації знань, необхідних для прийняття рішень у різних сферах діяльності;
3. такий набір технологій і методів, призначених для здобуття інформації з текстів на основі сучасних ІКТ, що дозволяє проявити закономірності, які можуть приводити до отримання корисної інформації і нових знань користувачами.

5. Видом віртуальної організації є :

а) віртуальне підприємство;

б) мережева організація;

в) традиційне підприємство.

**Розділ №9**

Контрольні запитання

1. **Що таке Hypertext Transport Protocol ?**

*HTTP* (Hypertext Transport Protocol) − комунікаційний протокол передачі гіпертексту, основне завдання якого полягає у встановленні зв’язку користувача із Web-сервером і забезпеченні доставки HTML-сторінок Web-броузеру клієнта.

1. **Як Ви розумієте Web-представництво у вигляді сайту?**

Розміщення інформації про підприємство на сайті

1. **Що таке Wiki-технологія?**

Wiki-технологія − це технологія побудови Web-сайта, що дозволяє відвідувачам брати участь у редагуванні його вмісту (як у виправленні помилок, так і додаванні нових матеріалів та посилань на них) і не вимагає від користувачів використання спеціальних програм, реєстрації на сервері і знання HTML.

1. **Які Вам відомо засоби створення Web-сайтів?**

* *Java.*
* *CGI* (Common Gateway Interface)
* *SSI* (Server Side Includes)
* *PHP* (Hypertext Preprocessor, препроцесор гіпертексту)
* *ASP* (Active Server Pages, активні сторінки сервера)
* *VBScript* або Visual BASIC Script (Visual Beginners All-purpose Symbolic Instruction Code Script, візуальний символічний універсальний командний код для початківців)
* *Flash*
* *XML* (Extensible Markup Language, розширювана мова розмітки)
* DHTML

1. **Чому сайт – це засіб підтримки і розвитку підприємства?**

Створивши один раз Web-сайт, підприємство одержує доступ до широкої аудиторії і може взаємодіяти зі своїми покупцями протягом цілодобово протягом року 24x7 − 24 години на добу і 7 днів на тиждень. Сайт (site) − це розділ сервера, присвячений певній тематиці.

Надання доступу до інформації у режимі реального часу дає змогу компаніям взаємодіяти з усіма більш якісно задовольняти потреби своїх клієнтів.

1. **Що таке Web-послуга?**

Web-послуги слугують для зв’язку гетерогенних застосунків на основі використання єдиних стандартів і протоколів. Завдяки Web-послугам функції будь-якої прикладної програми стають доступними через Internet, тобто можна створювати розподілені застосунки, компоненти яких вільно взаємодіятимуть один з одним.

1. **Які технології підтримують Web-послуги?**

Технології Web2 та Web3. Їх основою є соціальні мережі, спільна робота, спрямована на розробку інформаційних ресурсів. На основі цих нових технологій функціонують корпоративні блоги, енциклопедії Wiki тощо.

1. **Що таке архітектура, орієнтована на послуги?**

Термін, що використовують для опису Web-послуг

Тести

1. Блог – це:
2. мова розмітки тексту;
3. технологія побудови Web-сайта, що дозволяє відвідувачам брати участь у редагуванні його контента;
4. поповнювана через Web-інтерфейс колекція записів.
5. HTML – це:
6. мова програмування, створена для генерації сторінок на Web-сервері і роботи з базами данних;
7. мова, що інтерпретується, вбудовується в html-документ з метою включення до складу Web-сторінки інтерактивних елементів;
8. фундаментальна, базова технологія Internet − мова розмітки гіпертексту.
9. Web-сайт − це:
10. інформаційний Web-ресурс компанії, метою якого є інформування споживачів, партнерів, постачальників для надання послуг;
11. електронний магазин;
12. сукупність методів, процесів, комунікацій, мереж та програмно-технічних засобів, що забезпечує збір, зберігання, оброблення та передачу інформації з метою підвищення ефективності діяльності людей.
13. UDDI являє собою:
14. стандарт передачі повідомлень через Internet;
15. інформаційно-пошукову машину;
16. механізм виявлення Web-послуг.
17. Web-сервер – це:
    1. сукупність зв’язаних між собою Web-сторінок, що презентують підприємство та його послуги;
    2. компоненти прикладної логіки;
    3. компоненти доступу до даних.

**Розділ №10**

1. **Які стандарти існують для корпоративних інформаційних систем?**

ERP, MRP, MRPII, CSPR, ERPII

1. **Яким вимогам і стандартам відповідає КІС SAP R/3 ?**

ERP

1. **Охарактеризуйте основні етапи впровадження та експлуатації КІС.**

Етап I. Проведення інформаційного дослідження організації.

Етап II. Вибір архітектури КІС.

Етап III. Вибір СУБД.

Етап IV. Вибір системи автоматизації документообігу (САД).

Етап V. Вибір системи управління електронними документами (СУЕД).

Етап VI. Вибір спеціалізованого ПЗ.

Етап VII. Вибір СППР.

1. **Які основні критерії вибору КІС?**

Практика впровадження КІС виявляє ряд факторів, які необхідно враховувати в процесі вибору КІС: повноту функціональних можливостей системи; рівень реалізації функціональних модулів системи; вартість та тривалість впровадження; вплив системи на бізнес та бізнес-процеси підприємства; ефективність використання системи на підприємстві.

1. **Які Вам відомо класифікації КІС?**

|  |
| --- |
| **Клас систем** |
| Великі інтегровані системи |
| Середні інтегровані системи |
| Малі інтегровані системи |
| Локальні системи |

1. **У чому полягає суть і поняття КІС?**

До ІС, які можуть постійно адаптуватися до запитів всіх користувачів, зовнішніх і внутрішніх, і реагувати на всі зміни у режимі реального часу, можна зарахувати *корпоративні*.

1. **До якої групи виробничих КІС відносять: SAP R/3 , ERP Global?**

**До великих інтегрованих систем**

1. **Що таке клієнт-серверна архітектура?**

Архітектура клієнт-сервер є одним із архітектурних шаблонів програмного забезпечення та є домінуючою концепцією у створенні розподілених мережних застосунків і передбачає взаємодію та обмін даними між ними. Вона передбачає такі основні компоненти:

* набір серверів, які надають інформацію або інші послуги програмам, які звертаються до них;
* набір клієнтів, які використовують сервіси, що надаються серверами;
* мережа, яка забезпечує взаємодію між клієнтами та серверами.

Сервери є незалежними один від одного.

Тести

1. До якої групи КІС відносять: R3, Oracle Application?

1) Малі інтегровані КІС;

2) Середні інтегровані КІС;

3) Великі інтегровані КІС.

2. Якому стандарту відповідає КІС, де автоматизовано планування всіх виробничих ресурсів в ув'язці із замовленнями, фінансами, електронною комерцією?

1)MRP;

2)MRP-II;

3) ERP.

3. Якому стандарту відповідає КІС, де автоматизовано планування всіх виробничих ресурсів в ув'язці із фінансами, електронною комерцією?

1)MRP;

2)MRP-II;

3) ERP.

4. До якої групи виробничих КІС відносять “1С:Предприятие”?

1) Малі інтегровані КІС;

2) Середні інтегровані КІС;

3) Великі інтегровані КІС.

5. До якої групи виробничих КІС відносять Scala ?

1) Малі інтегровані КІС;

2) Середні інтегровані КІС;

3) Локальні ІС.

**1. General Info about SAP and LeverX**

SAP є одним із провідних світових виробників програмного забезпечення для управління бізнес-процесами, розробляючи рішення, які сприяють ефективній обробці даних та обміну інформацією між організаціями.

Заснована в 1972 році, компанія спочатку називалася розробкою програм системного аналізу (Systemanalyse Programmentwicklung), пізніше скорочено SAP. З тих пір вона виросла з невеликого підприємства з п’ятьох осіб до багатонаціонального підприємства зі штаб-квартирою у Вальдорфі, Німеччина, з понад 105 000 співробітників по всьому світу.

З впровадженням оригінального програмного забезпечення SAP R/2 і SAP R/3 компанія SAP встановила глобальний стандарт програмного забезпечення для планування ресурсів підприємства (ERP). Тепер SAP S/4HANA виводить ERP на новий рівень, використовуючи потужність обчислень у пам’яті для обробки величезних обсягів даних та підтримки передових технологій, таких як штучний інтелект (AI) та машинне навчання.

Інтегровані програми компанії об’єднують усі частини бізнесу в інтелектуальний набір на повністю цифровій платформі, замінюючи таким чином застарілу платформу, керовану процесами. Сьогодні SAP налічує понад 230 мільйонів користувачів хмари , понад 100 рішень, що охоплюють усі бізнес-функції, і найбільший портфель хмар з усіх постачальників.

LeverX надає клієнтам повний спектр послуг від планування до підтримки рішень SAP після запуску. Розробка "під ключ" з урахуванням специфіки діяльності та побажань замовника – правильний шлях до успішної цифрової трансформації бізнесу.

Наша місія – допомогти компаніям досягти максимальної віддачі від інвестицій у рішення SAP.

Ми використовуємо багаторічний досвід впровадження SAP-систем, щоб забезпечити максимальний ефект від автоматизації бізнес-процесів.

**2. Common architecture and landscape in classical SAP systems** (<https://www.tutorialspoint.com/sap_basis/sap_basis_system_landscape_architecture.htm> )

Архіектура системи SAP визначається як розташування серверів SAP. В ідеалі в середовищі SAP існує трисистемний ландшафт. Архітектура системи складається з сервера розробки (Dev), виробничого сервера (PROD) і сервера забезпечення якості (QAS).

Архітектура SAP визначається як технологічна структура системи SAP, і вона змінюється з часом на відміну від ландшафту системи. Архітектура SAP змінюється з новим програмним забезпеченням.

Запит у SAP System Landscape − РОЗРОБКА > Якість > ВИРОБНИЦТВО у напрямку.

Системний ландшафт включає в себе:

*Сервер пісочниці*

При реалізації проекту на початкових етапах використовується пісочниця, де виконуються всі налаштування, конфігурація.

*Сервер розробки*

Далі потрібно зберегти конфігурацію на сервері розробки та зберегти в запитах робочого середовища, і це потрібно перенести на виробничий сервер.

*Виробничий сервер*

Можна розглядати це як завершальний або найбільш витончений етап, на якому робота виконується, коли проект знаходиться у виробництві/запуску. Усі зміни, які вимагає клієнт, виконуються в середовищі DEV, а пізніше запит транспортується до виробництва (PROD).

3. **Application server**

Сервер додатків (AS)

Сервер додатків «тип використання» (AS) відіграє дуже важливу роль. Він відповідає за постійне зберігання даних та їх обробку. Для цього він пропонує інтерфейси програмування, які дозволяють іншим компонентам SAP NetWeaver, таким як SAP NetWeaver Portal, запитувати дані, редагувати їх або зберігати. Основним сховищем даних NetWeaver AS зазвичай є центральна база даних.

Сервер додатків, по суті, пропонує два інтерфейси програмування, ABAP і Java. Або іншими словами, «тип використання» сервера додатків у SAP NetWeaver містить два «типи використання»: сервер додатків ABAP (AS ABAP) і сервер додатків Java (AS Java). Обидва сервери додатків працюють з різними даними (різними схемами баз даних, які, тим не менш, можуть бути розташовані в одній базі даних).

**4. Main control buttons and commands (документація на сайті сапу)**

**Рядок заголовка:** містить назву поточного екрана.

**Рядок меню:** вміст рядка меню змінюється з кожним екраном. Переглядаючи меню в ньому, ви можете знайти всі функції на поточному екрані. Меню «Система» та «Довідка» присутні на кожному екрані, а пункти меню, що там містяться, не можна змінити.

**Командне поле:** Тут ви вводите команди, які потрібно виконати.

Стандартна панель інструментів: містить командне поле та серію кнопок. Вони ніколи не зміняться ні вигляду, ні положення, ні функції, і вони будуть присутні на кожному екрані. Деякі з них можуть бути неактивними, якщо їхня функція зараз недоступна.

**Панель інструментів програми:** змінюється на кожному екрані. Відображає кнопки, які надають швидкий доступ до пунктів меню для цього екрана.

**Меню інтерфейсу:** дає змогу налаштувати характеристики інтерфейсу користувача, отримати доступ до буфера обміну Windows та створити графіку.

**Область екрана:** Це велика область посередині екрана, яка відображає дані звіту або екран діалогової програми.

**Рядок стану:** Показує повідомлення, ідентифікатор системи, номер сеансу, номер клієнта, індикатор режиму вставки/надпису та поточний час.

**Команди**

auth - Для автентифікації користувача SAP CC.

debug - активувати режим налагодження програми.

doc - переглянути документацію.

exit - зупинити програму.

file - керувати вхідним файлом.

help - переглянути список команд.

normal - знову активувати звичайний режим програми.

shortcuts – переглянути список шорткатів.

status - переглянути поточний статус предметів, що підлягають оплаті.

template - переглянути модель вхідного файлу.

add - додати нову властивість до згенерованих елементів, що підлягають оплаті.

asi - визначити ідентифікатор послуги.

ausi - визначити ідентифікатор користувача для послуги.

cd - визначити дату споживання.

cic - вказати ідентифікатор класу предметів, що підлягають оплаті, включених у створені елементи, що підлягають оплаті.

mod - змінити значення властивості.

nb - визначити кількість предметів, що підлягають оплаті.

rem - видалити властивість.

**5. Namespace (Z, Y, defined on projects) (**[**https://www.sapdatasheet.org/abap/dtel/namespace.html**](https://www.sapdatasheet.org/abap/dtel/namespace.html)**, http://saphelp.ucc.ovgu.de/NW750/EN/4e/df088be37c1f65e10000000a42189d/content.html)**

[Простори імен - це ідентифікатори, призначені виключно SAP, які дозволяють клієнтам SAP, партнерам SAP і самому SAP розробляти компоненти і продукти SAP на основі додатків SAP без ризику виникнення зіткнень назв (при доставці на зовнішні SAP Systems або імпорті зовнішніх продуктів у внутрішні системи).](https://www.reverso.net/%D0%BF%D0%B5%D1%80%D0%B5%D0%B2%D0%BE%D0%B4-%D1%82%D0%B5%D0%BA%D1%81%D1%82%D0%B0#sl=eng&tl=null&text=Namespaces%20are%20identifiers%20assigned%20exclusively%20by%20SAP,%20which%20enable%20SAP%20customers,%20SAP%20partners%20and%20SAP%20itself%20to%20develop%20SAP%20components%20and%20products%20on%20the%20basis%20of%20SAP%20applicatio) Об'єкти в ABAP Workbench присвоюються простору імен шляхом розміщення зарезервованих префіксів простору імен перед назвою об'єкта. Ідентифікатор простору назв починається і закінчується символом «/» (delimiter) і може бути довжиною не більше 10 символів.

Діють такі конвенції про іменування:

Для класів програм:

Простір назв SAP:

Прикладні класи SAP завжди починаються з таких префіксів:

Від 0 до 9 (наприклад 0FLIGHT)

А до W

Простір імен партнерів і клієнтів:

Прикладні класи партнерів або клієнтів завжди починаються з таких префіксів:

Х — Y

Префікс простору назв /MY\_COMPANY/

**6. Transport requests and their types, transport tasks**

<https://blogs.sap.com/2016/02/17/all-about-transport-request/>

<https://www.guru99.com/what-is-transport-request-how-to-importexport-it-check-logs.html>

<https://www.tutorialscampus.com/sap-basis/transport-request.htm>

**Transport request** - це колекція інформації про зміни конфігурації, необхідні для переміщення одного джерела до одного або декількох пунктів призначення. Запит на транспортування створюється для змін в таблиці конфігурації або програмі.

Кожен transport request містить одну або кілька конфігурацій змін, які також називаються завданнями зміни(**transport task**). Задача зміни призначається тільки одному користувачеві і являє собою набір об'єктів, які змінюються зазначеним користувачем. Transport request може містити кілька завдань, які виконуються декількома користувачами. Transport request містить декілька завдань, які зберігаються як файли, що зберігаються в папці вікон. Завдання не є самотранспортованим, але є частиною транспортного запиту Транспортний запит може бути випущений тільки після того, як всі завдання всередині транспортного запиту будуть виконані, випущені або видалені. Рух транспортних запитів від одного клієнта до іншого здійснюється базовою командою. Транспортний запит записує інформацію про тип зміни, мету транспорту, категорію запиту та цільову систему, яка відома як Зміна запиту. Транзакція SE09 використовується для транспортування транспортного запиту Від одного клієнта до іншого. Запити на зміни називаються в стандартному форматі, показаному нижче.

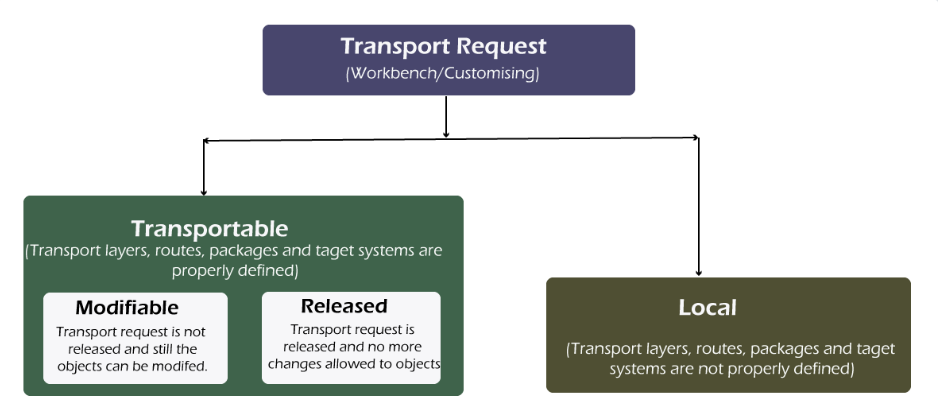
<SID>K<Number>

SID - ідентифікатор системи.

К - вказує ключове слово/алфавіт.

Number - може бути що завгодно в діапазоні від 900001 до 999999.

Системні адміністратори не можуть змінювати ці запити на зміну.



There are four **types of transport requests**

* Local request
* Change request
* Transport of copies
* Workbench request

**Types of Transport Request:**

* K type – with change in integrated system to consolidated system.
* C type – without change in integrated system to consolidated system.
* T type – move the one system to another system.

**7. Main transactions: SE01, SE10, SE11, SE16, ST22, SE24, SE37, SM36, SM37, SE38, SM50, SM59, SE80, SE84, SE91, and others**

| **T-Codes** | **Description** |
| --- | --- |
| SU01 | User Maintenance |
| SU01D | User Display |
| SU10 | Mass Changes to User masters, Mass User maintenance |
| AL08 | Global User overview (Lists all the users of all instances of the SAP System) |
| AL11 | Display CCMS operating system files |
| AL12 | Buffer Synchronization |
| DB02 | Missing Database objects and space requirements |
| DB13 | Scheduling (Used for scheduling various standard jobs like checkdb, backups, etc….) |
| SM01 | Lock Transactions |
| SM02 | Post system messages in case of planned or scheduled downtime of SAP Systems |
| SM04 | Local User list (It lists users belonging to only that selected instance) |
| SM12 | Display and delete SAP locks |
| SM13 | Display update records |
| SM21 | System log (It contains very important system messages and will be useful for troubleshooting most of the issues) |
| SM28 | SAp initial installation consistency check (same as SICK t-Code) |
| SM30 | Call view maintenance |
| SM31 | Table maintenance |
| SM35 | Batch input monitoring |
| SM36 | Schedule background jobs |
| SM37 | Overview of background jobs |
| SM50 | Overview of work processes |
| SM51 | Overview of instances |
| SM56 | Nubmer range buffer |
| SM58 | Asynchronous RFC error log |
| SM59 | RFC connections maintenance (creation, modification,etc…) |
| SM63 | Display / Maintain operation modes |
| SM64 | Trigger an event |
| SM66 | Global work process overview |
| SM69 | Maintain external operating system commands |
| SMLG | Logon group maintenance |
| SMLI | Language import |
| SMLT | Language addministration |
| PFCG | Role maintenance |
| OSS1 | SAP service marketplace logon |
| SA38 | Execute an ABAP program |
| SCC1 | Copy Client via transport request |
| SCCL | Local Client copy |
| SCC3 | Client copy log |
| SCC4 | Client administration |
| SCC5 | Delete Client |
| SCC6 | Client import |
| SCC7 | Client import post processing |
| SCC8 | Client export |
| SCC9 | Remote Client copy |
| SP01 | Spool control (used to view spool related details) |
| SP02 | Display output request details |
| SP11 | TemSe table of contents |
| SP12 | TemSe Administration |
| SPAD | Spool administration and printer configuration |
| SPAM | SAP support pack manager (Used for performing SAP support pack upgrade) |
| SPAU | Display modified objects in the runtime environment. (Used during SAP support pack upgrade) |
| SPDD | Display modified DDID objects (Used during SAP support pack upgrade) |
| SPRO | Customizing initial screen |
| FILE | Related to archiving (Used to map assignment between logical and physical names) |
| RZ01 | Graphical background job scheduling monitor |
| RZ03 | Control panel for operation modes and server states. (Operation mode can be chosen) |
| RZ04 | Maintenance of operation modes and instances |
| RZ10 | Profile maintenance and changing SAP parameter values |
| RZ20 | Alert monitor |
| RZ21 | Alert monitor customizing |
| SALE | IMG application link enabling |
| I18N | Internationalization (To add new language) |
| SICF | Activate / Deactivate servces |
| SSAA | Administration activities (SAp System Administration Assistant) |
| SMMS | Message server monitor |
| DB54 | Create logical system |
| SE01 | Transport organizer |
| SE03 | Workbench organizer tools |
| SE06 | Setup workbench organizer |
| SE07 | Display status of transport system |
| SE09 | Workbench organizer |
| SE10 | Customizing organizer (User to view transport requests based on TYPE, USER, etc….) |
| SE16 | Display table content |
| SE80 | Object navigator (ABAB development workbench) |
| SICK | SAP initial installation consistency check |
| ST01 | SAP system trace |
| ST02 | View all the SAP R/3 buffer statistics |
| ST03 | Perform workload analysis |
| ST04 | Performance Overview (View various statistics of RDBMS activities) |
| ST05 | Performance Analysis (To take SQL trace) |
| ST06 | Operating System monitoring |
| ST07 | Application monitoring |
| ST10 | Table call statistics |
| ST11 | Display developer trace |
| ST22 | ABAP runtime error analysis (ABAP dumps) |
| STMS | Transport management system |
| STUN | R/3 performance menu |
| SXI\_CACHE | Cache status for PI System |
| SLDCHECK | SLD functionality |
| SXI\_MONITOR | Monitor SXI for PI |

**SE11** − Dictionary definitions

**SE24** − Class builder

**SE37 SAP tcode** - ABAP Function Modules

**SE38** is used for the task : ABAP Editor

SE84 - OBJECT NAVIGATOR

SE91 (Message Maintenance) is a standard SAP transaction code available within R/3 SAP systems depending on your version and release level

**8. Packages. Local Packages**

(https://help-sap-com.translate.goog/docs/SAP\_HANA\_PLATFORM/400066065a1b46cf91df0ab436404ddc/016a60fe929a4e9e89bbb3b6f7aad409.html?locale=en-US&version=2.0.02&\_x\_tr\_sl=en&\_x\_tr\_tl=uk&\_x\_tr\_hl=ru&\_x\_tr\_pto=wapp)

У SAP HANA пакунок зазвичай складається з набору об’єктів сховища, які можна транспортувати між системами *.*Кілька пакетів можна об’єднати в одиницю доставки (DU).

Пакет SAP HANA визначає простір імен, у якому існують об’єкти репозиторію. Кожен об’єкт сховища призначається пакету, і кожен пакет має бути призначений певній одиниці доставки. У сховищі кожен об’єкт однозначно ідентифікується комбінацією наступної інформації:

* Назва пакета
* Ім'я об'єкта
* Тип об'єкта

**Примітка**Кілька об’єктів одного типу можуть мати однакове ім’я об’єкта, якщо вони належать до різних пакетів.

Перш ніж почати процес розробки пакета, зверніть увагу на такі важливі моменти:

* Ієрархія пакетів

Кожен постачальник використовує спеціальний простір імен, а створена вами ієрархія пакетів дозволяє зберігати різні елементи програми в логічному порядку, у якому легко орієнтуватися.

* Тип пакету

Пакети можуть бути структурними і неструктурними; деякі пакети містять вміст; інші пакети містять лише інші (під)пакети.

* Умови найменування пакетів

Існують рекомендації та обмеження щодо назв пакетів, наприклад, максимальна довжина імені та які символи не можна використовувати.

## Умови найменування пакетів

До назв пакетів застосовуються такі правила:

* Дозволені символи

У назвах пакетів дозволені малі/верхні літери (aA-zZ), цифри (0-9), дефіси (-), підкреслення (\_) і крапки (.). Точки в назві пакета визначають логічну ієрархію. Наприклад, "abc" визначає пакет "a", який містить підпакет "b", який, у свою чергу, містить підпакет "c".

**Типи пакетів репозиторіїв**

SAP HANA дозволяє використовувати різні типи пакетів, які призначені для використання в певних сценаріях.

Служби додатків SAP HANA надають або дозволяють такі **типи** пакетів :

* Структурна

Пакет містить лише підпакети; він не може містити об'єкти репозиторію.

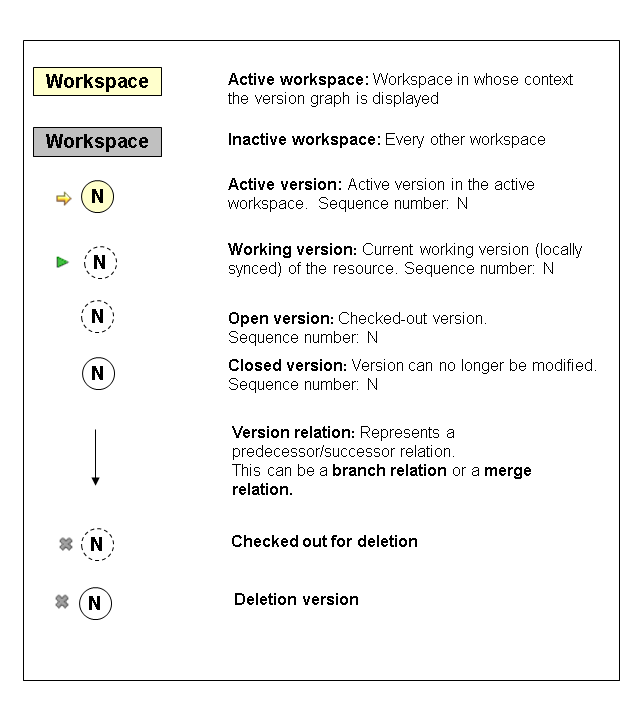
* Неструктурні

Пакет містить як об’єкти репозиторію, так і підпакети.

У розділі «Локальні об’єкти» ви можете знайти об’єкти розробки, наприклад звіти, які не мають цільової системи для транспортування. Системна бібліотека містить усі пакунки, розроблені SAP або вашою компанією/замовником. Якщо ви часто розробляєте пакет, ви також можете позначити його як улюблений.

**9. System of versioning control.**

**Контроль версій (Version control)** – це механізм, який допомагає підтримувати історію переглядів ресурсу розробки та відстежувати внесені до нього зміни. Він визначає набір обмежень щодо того, як можна змінити ресурс розробки. Ресурс розробки, який відповідає обмеженням, визначеним контролем версій, називається версійним ресурсом. Коли версійний ресурс змінено або видалено, для ресурсу створюється нова версія. Унікальний порядковий номер пов’язаний з кожною версією ресурсу, створеної в певній робочій області. Цей порядковий номер визначає порядок, у якому були створені версії в цій робочій області. Репозиторій часу розробки (DTR) графічно представляє зв’язок між різними версіями версійного ресурсу у вигляді графіка версій.



Механізм контролю версій DTR відстежує такі зміни:

* Додавання ресурсу в репозиторій.
* Модифікація ресурсу в репозиторії.
* Видалення ресурсу зі сховища.
* Вирішення конфліктів за допомогою відкидання злиття.

У всіх перерахованих вище випадках створюється нова версія ресурсу.

**Version control system** – система для керування змінами та керування ними, коли команда людей спільно працює над одним набором файлів.

(<https://help.sap.com/saphelp_nw73/HELPDATA/EN/49/0fdc31326d209ce10000000a42189d/frameset.htm>  
<https://help.sap.com/doc/saphelp_scm700_ehp01/7.0.1/en-US/4d/7cf875219b4e06e10000000a421944/content.htm>)

**10. Main functions of debugger**

Головні функції дебагеру:

Коли SAP-програміст пише програму та виконує її, за умови що вона повертає помилку чи дамп, SAP-дебагер використовується для перевірки та вивчення помилок під час рантайму.

1. **ABAP data types**

(Документація SAP: <https://help.sap.com/SAPhelp_nw73/helpdata/en/fc/eb2fb2358411d1829f0000e829fbfe/frameset.htm> )

Усі мови програмування розрізняють різні типи даних із різним використанням для зберігання чи відображення значень і числові дані для обчислень. Атрибути, про які йде мова, описуються за допомогою типів даних.

Типи даних можна розділити на елементарні, довідкові та складні.

*Елементарні типи* (*Elementary Types)*

Це типи даних фіксованої або змінної довжини, які не складаються з інших типів.

Ви також можете визначити власні елементарні типи даних у ABAP за допомогою оператора TYPES. Ви базуєте їх на попередньо визначених типах даних. Наприклад, ви можете визначити тип даних P\_2 з двома знаками після коми на основі попередньо визначеного типу даних P. Потім ви можете використовувати цей новий тип у своїх оголошеннях даних.

*Типи посилань (Reference Types)*

Типи посилань — це глибокі типи даних, які описують опорні змінні, тобто об’єкти даних, які містять посилання. Посилальну змінну можна визначити як компонент складного об’єкта даних, такого як структура або внутрішня таблиця, а також як одне поле.

*Складні типи даних (Complex Data Types)*

Складні типи даних складаються з інших типів даних. Тут розрізняють структуровані (Structures) типи та типи таблиць (Internal Tables).

* *Structures*

Екземпляри структурованих типів відомі як структури і використовуються в програмах ABAP для групування робочих областей, які логічно належать разом. Оскільки окремі елементи в структурі можуть бути будь-якого типу, а самі можуть бути структурами або внутрішніми таблицями, можливі варіанти використання структур дуже різноманітні. Наприклад, ви можете використовувати структуру з елементарними типами даних для відображення рядків із таблиці бази даних у програмі. Ви також можете використовувати структури, що містять агреговані елементи, щоб включити всі атрибути екрана або елемента керування в один об’єкт даних.

* Internal Tables

Внутрішні таблиці складаються з ряду рядків, які мають однаковий тип даних.

**12. Data types used for domains creation.** (<https://www.ibm.com/docs/en/ida/9.1.1?topic=types-domain-models>)

A domain data type represents **an abstract data type that can be restricted by adding constraints**. Domain data types are based on base data types. For example, you can define domain data types for commonly used definitions such as social security number, sex, height, or marital status.

Тип даних домену являє собою абстрактний тип даних, який можна обмежити шляхом додавання обмежень. Типи даних домену базуються на базових типах даних. Наприклад, ви можете визначити типи даних домену для загальновживаних визначень, таких як номер соціального страхування, стать, зріст або сімейний стан.

**13. Domains and fixed values**

Домен:

Домен – це окремий об’єкт словника, який визначає технічні та семантичні атрибути елементарних типів даних. Елементи даних можна визначити з посиланням на домен і успадкувати його атрибути. Домени можна використовувати лише в елементах даних.

Фіксоване значення:

Фіксовані значення використовуються для додаткового обмеження діапазону значень домену. Якщо ви визначаєте фіксовані значення для домену, вони використовуються в шаблонах екрана перевірки введення. Ви можете визначити інтервали фіксованих значень, ввівши верхню та нижню межі, або вказавши окремі значення. За потреби можна комбінувати діапазони значень та окремі значення. Ви можете ввести пояснювальний текст для кожного окремого значення або інтервалу. Цей текст відображається у довідці введення.

Визначити фіксовані значення можна лише для доменів типів даних CHAR, NUMC, DEC, INT1, INT2 та INT4.

**14. Data elements (workbook 2)**

Елемент даних описує простий тип або посилальний тип.

Простий тип визначений вбудованим типом даних, довжиною і, можливо, кількістю десяткових знаків. Ці атрибути типу можуть бути визначені безпосередньо в елементі даних або скопійовані з домену.

Посилальний тип визначає типи опорних змінних у програмах ABAP.

Інформація про значення поля таблиці або компонента структури та інформація про редагування відповідного поле екрана може бути приствоєною елементу даних. Ця інформація автоматично доступна для всіх полів екрана, які посилаються на елемент даних.

**Створення data elements:**

1. Транзакція SE11

На початковому екрані словника ABAP виберіть тип об’єкта data element, введіть назву елемента даних та виберіть Створити. З’явиться діалогове вікно.

2. Виберіть data element і виберіть Продовжити.

3. Введіть короткий пояснювальний текст.

4. На вкладці Тип даних вкажіть тип даних, довжину та, можливо, число

десяткових знаків елемента даних. Ви можете визначити ці атрибути, вказавши назву домену або за прямим записом типу.

Якщо елемент даних повинен мати атрибути типу домену, потрібно лише вибрати Домен і ввести ім’я домену у відповідне поле.

5. На вкладці «Field label» ви можете зберігати текстову інформацію для елемента даних.

6. Збережіть елемент даних. Призначте його своєму пакету розробки.

7. Активуйте елемент даних.

**15. Structures**

**(https://www.tutorialspoint.com/sap\_abap/sap\_abap\_structures.htm)**

Структура — це об'єкт даних, який складається з компонентів будь-якого типу даних, що зберігаються один за одним в пам'яті. Структури корисні для виведення полів екрану, і для маніпулювання даними, які мають послідовний формат, визначений дискретною кількістю полів. Структура може мати тільки один запис в один раз, але таблиця може мати багато записів.

(Приклад створення)

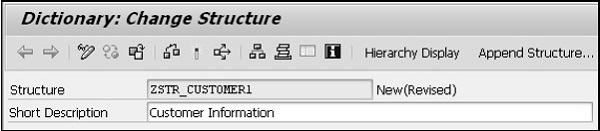
Creating a Structure

**Step 1** − Go to transaction SE11.

**Step 2** − Click on the ‘Data type’ option on the screen. Enter the name 'ZSTR\_CUSTOMER1' and click on Create button.

**Step 3** − Select the option 'Structure' in the next screen and press Enter. You can see 'Maintain / Change Structure' wizard.

**Step 4** − Enter the Short Description as shown in the following snapshot.

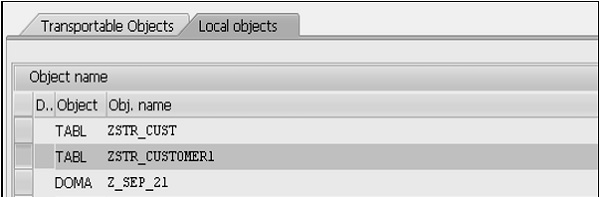


**Step 5** − Enter the Component (Field Name) and Component Type (Data Element).

**Note** &minu; Here the component names start with Z as per the SAP recommendation. Let's use data elements that we have already created in the database table.

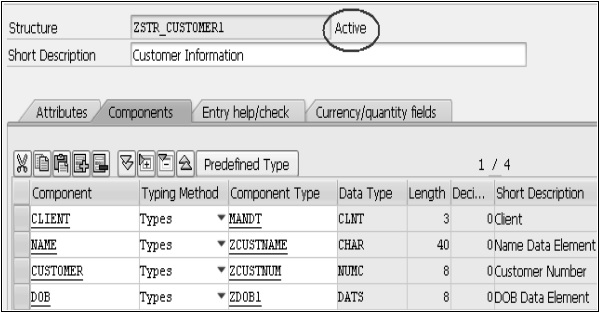
**Step 6** − You need to Save, Check and Activate after providing all the components and component types.

The following screen appears −



**Step 7** − As this 'ZSTR\_CUSTOMER1' is highlighted, click the green tick button. This window disappears and the status bar will display the message ‘Active’.

The structure is now activated as shown in the following snapshot −



**16. Table types and their types (standard, sorted, hashed)**

<https://gocoding.org/table-type-in-sap-abap/>

<https://www.dan852.com/sap-abap-types-internal-table-declaration/>

**Тип таблиці в SAP ABAP** - це глобальний багаторазовий об'єкт, який використовується для визначення структури і функціональних атрибутів внутрішньої таблиці в ABAP.

АВАР знає три типи таблиць **STANDARD, SORTED і HASHED**. Кожен з цих типів таблиць відрізняється тим, яким чином вони оголошуються, доступні і змінюються під час виконання.

Найпростіший тип - **STANDARD TABLE**. Це працює в основному як масив в інших мовах програмування і може бути використано для невеликого набору даних.

**Standard Table**

* Unsorted but can be sorted with SORT
* Access via table index and table key

DATA: sales\_order\_items TYPE STANDARD TABLE OF ty\_sales\_order\_item WITH DEFAULT KEY.

Якщо дані повинні зберігатися в певному порядку, рекомендується використовувати **SORTED TABLE**. Під час виконання гарантується, що всі записи таблиці сортуються відповідно до вказаного ключа таблиці. Майте на увазі, що додавання нових записів до заяви APPEND неможливе.

**Sorted Table**

* Sorted according to the defined table key
* Access via table index and table key
* Fast key access thanks to binary search
* Primary Key: posnr
* Secondary Key: flag

DATA: sales\_order\_items TYPE SORTED TABLE OF string WITH UNIQUE KEY table\_line.

Для великих наборів даних, доступ до яких буде здійснюватися за допомогою клавіші таблиці, слід використовувати **HASHED TABLE**. Завдяки хешованому ключу записи можна отримати швидко і ефективно.

**Hashed Table**

* Access only via a unique key, no index
* Each key value may only appear once, otherwise an exception is raised
* Fast if all key fields are included

DATA: sales\_order\_items TYPE HASHED TABLE OF ty\_sales\_order\_item WITH UNIQUE DEFAULT KEY.

**17. Keys in table types**

Стандартний ключ: Структура первинного ключа залежить від категорії типу рядка. У структурованому типі рядка стандартний ключ складається з усіх символьних компонентів рядка таблиці. У елементарному типі рядка або типі посилання як тип рядка стандартний ключ складається з усього рядка таблиці. Якщо тип рядка є типом таблиці, стандартний ключ порожній. Примітка Порожній ключ дозволяється лише для режиму доступу Стандартна таблиця. Тип лінії: Первинний ключ складається з усіх полів типу рядка. Ключові компоненти: Первинний ключ визначається явно шляхом вибору компонентів (полів) типу рядка. Це можливо лише в тому випадку, якщо ви вибрали структуру, таблицю або подання як тип рядка. Ключ не вказано: Первинний ключ не вказано. Це визначає загальний тип таблиці.

1. **Lock Objects and related functions**

(https://help-sap-com.translate.goog/doc/saphelp\_nw75/7.5.5/EN-US/cf/21eea5446011d189700000e8322d00/content.htm?no\_cache=true&\_x\_tr\_sl=en&\_x\_tr\_tl=uk&\_x\_tr\_hl=ru&\_x\_tr\_pto=wapp)

Блокування об'єктів

Визначення

Система SAP синхронізує одночасний доступ кількох користувачів до одних і тих самих записів даних за допомогою [механізму блокування](https://help-sap-com.translate.goog/doc/saphelp_nw75/7.5.5/EN-US/cf/21eed9446011d189700000e8322d00/content.htm?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp) . Коли інтерактивні транзакції запрограмовані, система встановлює та знімає блокування, викликаючи функціональні модулі (див. [Функціональні модулі для запитів на блокування](https://help-sap-com.translate.goog/doc/saphelp_nw75/7.5.5/EN-US/cf/21eebf446011d189700000e8322d00/content.htm?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp) ). Ці функціональні модулі автоматично генеруються на основі визначення об’єктів блокування в словнику ABAP.

Структура

В об’єкті блокування необхідно визначити таблиці, в яких мають бути заблоковані записи даних за допомогою запиту на блокування, і ключові поля цих таблиць. Коли ви вибираєте таблиці, спочатку потрібно вибрати основну таблицю. Ви також можете додати додаткові таблиці (вторинні таблиці), використовуючи зв’язки із зовнішніми ключами (див. [Умови для зовнішніх ключів](https://help-sap-com.translate.goog/doc/saphelp_nw75/7.5.5/EN-US/cf/21eecc446011d189700000e8322d00/content.htm?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp) ). На наступному малюнку показано первинну таблицю та її зв’язки з двома іншими таблицями (вторинними таблицями).

**Аргументи блокування**

Аргумент блокування таблиці в об’єкті блокування складається з ключових полів таблиці.

Поля аргументів блокування об'єкта блокування використовуються як вхідні параметри у функціональних модулях для встановлення та зняття блокувань, створених із визначення об'єкта блокування. Коли ці функціональні модулі викликаються, рядки таблиці, які потрібно заблокувати або розблокувати, визначаються визначенням певних значень у цих полях. Ці значення також можуть бути загальними. Поля аргументу блокування визначають, яка підмножина рядків таблиці має бути заблокована.

Найпростіший випадок об’єкта блокування складається саме з однієї таблиці, а аргумент блокування таблиці є первинним ключем цієї таблиці. Ви також можете включити кілька таблиць в об’єкт блокування. Запит на блокування може блокувати не тільки запис таблиці, а й цілий логічний об’єкт. Таким логічним об’єктом може бути, наприклад, документ, що містить запис у таблиці заголовка та N записів у таблиці позицій. На малюнку нижче показано заблоковані записи в прикладі з трьома таблицями.

Примітка

* Блокування також можна встановити з програм в інших системах з відповідними інтерфейсами, якщо об’єкт блокування було визначено з авторизацією RFC.
* Для кожної таблиці в об’єкті блокування можна призначити [режим блокування.](https://help-sap-com.translate.goog/doc/saphelp_nw75/7.5.5/EN-US/cf/21eeb2446011d189700000e8322d00/content.htm?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp)Цей режим визначає, як інші користувачі можуть отримати доступ до заблокованого запису таблиці.

**19. Search Helps.**

**Довідки пошуку** (**Search Helps**) – це об’єкти, які можна використовувати, щоб призначити довідку введення (F4 Help) для полів екрана. Ви можете зробити це, створивши довідку пошуку в словнику ABAP і прикріпивши її до відповідного поля екрана.

**Використання**

Довідка введення (F4 Help) є стандартною функцією системи SAP. Ви можете відобразити список усіх можливих значень введення для екранного поля за допомогою довідки введення. Ви можете покращити можливі вхідні значення за допомогою додаткової інформації. Це має сенс, особливо коли поле вимагає введення формального ключа.

Щоб детально зрозуміти функціональні можливості пошуку, ви повинні бути знайомі зі стандартним процесом довідки введення.

**Довідковий процес стандартного введення (Standard Input Help Process)**

Ви можете викликати довідку щодо введення за допомогою таких кроків (деякі кроки можна пропустити, залежно від визначення довідки введення):

* Ви запускаєте довідку щодо введення, щоб відобразити можливі значення введення для поля (поля пошуку) у шаблоні екрана.
* Система пропонує вам ряд можливих шляхів пошуку. Ви повинні вибрати один із цих шляхів пошуку. Кожен шлях пошуку пропонує ряд обмежень для обмеження кількості можливих вхідних значень. Ці значення пропонуються в діалоговому вікні для обмеження значення, коли вибрано шлях пошуку.
* Введіть обмеження, якщо потрібно, а потім почніть пошук.

**Функція довідки пошуку (Function of a Search Help)**

Цей стандартний процес можна повністю визначити, створивши довідку пошуку в словнику ABAP. Цю довідку пошуку потрібно призначити лише тим полям екрана, у яких вона має бути доступна.

Існує два типи довідок пошуку:

* **Елементарні довідки пошуку (Elementary search helps)**

Вони описують шлях пошуку. Довідка елементарного пошуку повинна визначати наступне:

* Звідки потрібно прочитати дані списку звернень (метод вибору);
* Як реалізується обмін значеннями між шаблоном екрана та методом вибору (інтерфейс довідки пошуку);
* Як має бути визначена онлайн-довідка для введення (онлайнова поведінка довідки пошуку).
* **Колективні довідки (Collective search helps)**

Колективна довідка пошуку допомагає об’єднати кілька елементарних пошукових допоміжних засобів. Може запропонувати кілька альтернативних шляхів пошуку.

(<https://help.sap.com/SAPhelp_nw73/helpdata/en/cf/21ee2b446011d189700000e8322d00/frameset.htm>)

**20. Tables and table types. Primary keys. Main table settings**

**Таблиці та їх типи. Первинні ключі. Основні налаштування таблиць**

Стандартна таблиця,

Відсортована таблиця,

Індексована таблиця,

Хеш-таблиця,

Таблиця без типу (Універсальний тип)

**Первинний ключ** — атрибут або набір атрибутів, який однозначно ідентифікує кортеж даного [відношення](https://uk.wikipedia.org/wiki/%D0%92%D1%96%D0%B4%D0%BD%D0%BE%D1%88%D0%B5%D0%BD%D0%BD%D1%8F_(%D1%80%D0%B5%D0%BB%D1%8F%D1%86%D1%96%D0%B9%D0%BD%D0%B0_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C))

1. **Dependent tables and foreign keys**

(Документація SAP: <https://help.sap.com/saphelp_nw73/helpdata/en/44/e3a1e04c836c2fe10000000a114a6b/content.htm?no_cache=true>

)

Ви можете визначити залежність між двома таблицями, яка впливає на таблиці під час внесення змін до рядків. Це відношення називається посилальним обмеженням (referential constraint) або залежністю від зовнішнього ключа (foreign key dependency).

*Простий зовнішній ключ (Simple Foreign Key)*

Ви можете визначити залежності зовнішнього ключа, які посилаються лише на один стовпець первинного ключа в таблиці.

*Зовнішній ключ у кількох стовпцях (Foreign Key in Several Columns)*

Ви можете визначити зовнішній ключ для кількох стовпців первинного ключа.

**22. Table indexes** (<https://help.sap.com/doc/abapdocu_752_index_htm/7.52/en-US/abenddic_database_tables_index.htm>)

Індекс у таблиці бази даних допомагає прискорити вибір рядків. Індекс — це відсортована копія вибраних полів таблиці бази даних. Додаткове поле містить покажчик на фактичні рядки таблиці. Сортування забезпечує швидший доступ до рядків у таблиці, наприклад, у двійковому пошуку. Таблиця бази даних має принаймні один первинний індекс, визначений її ключовими полями. Він також може мати один або кілька додаткових вторинних індексів.

*Первинний індекс*

Первинний індекс — це унікальний індекс, створений із ключових полів первинного ключа. Він завжди створюється автоматично в AS ABAP. Для кожної комбінації індексних полів у таблиці існує максимум один запис. Якщо первинний індекс не може бути використаний для ідентифікації набору результатів, наприклад, через те, що жодне поле з основного індексу не було вибрано, таблиця сканується повністю або робиться спроба використати відповідний вторинний індекс (якщо він існує).

*Вторинні індекси*

Поряд із первинним індексом, визначеним за допомогою первинного ключа, для таблиці бази даних можна створити як унікальні, так і неунікальні вторинні індекси. Створення вторинних індексів зазвичай покращує продуктивність читання бази даних, що оцінює індекси бази даних.

Вторинні індекси таблиці бази даних складаються з серії полів таблиці та визначаються буквено-цифровим ідентифікатором індексу, що містить максимум три символи (літери або цифри). Ідентифікатор 0 зарезервовано для первинного індексу. Поля таблиці з вбудованими типами даних STRING і RAWSTRING не повинні бути полями індексу. Рекомендується, щоб поля таблиці з типом даних FLTP не були полями індексу.

*Повнотекстовий покажчик*

База даних SAP HANA підтримує повнотекстовий індекс як індекс вторинної таблиці. Повнотекстовий індекс створюється як додатковий невидимий стовпець у базі даних. Вміст стовпця, створеного для повнотекстового індексу, зберігається в цьому додатковому стовпці з відповідним форматуванням і оцінюється при зверненні до відповідних даних.

**23. Views and their types**

Дані про об'єкт програми можна розподілити по декількох таблицях. Визначаючи подання, ви можете визначити представлення, залежне від програми, яке поєднує ці дані. Структура такого представлення визначається шляхом вказівки таблиць і полів, які використовуються в поданні. Поля, які не є обов’язковими, можна приховати, таким чином мінімізуючи інтерфейси. Подання можна використовувати в програмах ABAP для вибору даних.

Підтримуються чотири різних типи перегляду. Вони відрізняються способом реалізації представлення та методами, дозволеними для доступу до даних представлення:

* Database views реалізовані з еквівалентним представленням бази даних.
* Projection views використовуються для приховування полів таблиці (тільки проекція).
* Help views можна використовувати як метод вибору в довідках пошуку.
* Maintenance views дозволяють підтримувати дані, розподілені в кількох таблицях для одного об’єкта програми одночасно.

**24. Ways of filling table with data (SE16, Maintenance dialog, View Cluster, SQL statements) (workbook 3, документація на сайті сапу** [**https://help.sap.com/docs/HANA\_SERVICE\_CF/7c78579ce9b14a669c1f3295b0d8ca16/20f7f70975191014a76da70c9181720e.html?locale=en-US**](https://help.sap.com/docs/HANA_SERVICE_CF/7c78579ce9b14a669c1f3295b0d8ca16/20f7f70975191014a76da70c9181720e.html?locale=en-US)**)**

**SE16:** ввести назву транзакції SE16, в полі заповнити назву таблички і натиснути зверху кнопку Create Entries (F5), заповнити пропоновані поля.

**Maintenance dialog:** SE54, ввести назву таблиці та натиснути кнопку Edit view cluster. Далі натиснути кнопку Test. У новому вікні обрати кнопку New Entries та заповнити поля.

**SQL:** INSERT INTO *<table\_name>* [ PARTITION *<num>* ]

[ [ AS ] *<table\_alias>* ]

[ ( *<column\_list\_clause>* ) ]

{ { *<value\_list\_clause>* | [ *<overriding\_clause>* ] *<subquery>* } | *<with\_clause>* }

[ *<hint\_clause>* ]

**25. Customizing tables. SPRO and SIMGH transactions**

**(https://help.sap.com/docs/ABAP\_PLATFORM\_NEW/4a368c163b08418890a406d413933ba7/4d7e6d494e0f0d77e10000000a42189e.html?locale=en-US&version=202110.000)**

Визначення  
Зміна даних — це дані, створені клієнтами, коли вони налаштовують свої системи.  
Дані настроювання в клієнті клієнта не перезаписуються за допомогою імпорту мови або оновлення. Дані імпортуються лише в клієнт 000. Це тому, що настроювання даних є відповідальністю клієнта і не може бути перезаписане даними SAP. Це стосується і, без обмежень, перекладів. Проте, клієнти можуть час від часу захотіти оновити свої вибіркові дані та типові параметри за допомогою останніх перекладів від SAP.  
Примітка  
Копіювати клієнтів з клієнта 000 тільки після того, як всі необхідні мови були імпортовані в клієнт 000.  
Ви маєте наступні параметри для оновлення мови перекладу для налаштування параметрів існуючого клієнта:  
*Використання інструментів*  
*Ручний переклад*  
  
**Використання інструментів**  
Станом на Basis Release 3.1H, SAP надає наступні інструменти для оновлення Зберігаючих текстів у клієнтів, відмінних від 000:  
  
RREFILL  
RSREFILL порівнює Customizing тексти в логоні клієнта з SAP-пропозиціями на довідковою мовою в клієнті 000. Якщо ці тексти ідентичні, то їх переклади на цільовій мові копіюються з клієнта 000 на клієнта-логона. Дані вставляються тільки; існуючі дані не перезаписуються.  
  
Обслуговування клієнтів  
Ця функція працює так само, як і RSREFILL, але не робить порівняння з опорною мовою. Це означає, що більше текстів можна скопіювати з клієнта 000 до цільового клієнта. Як і в RSREFILL, вставляється Customizing data. Дані не видаляються або перезаписуються.  
RSREFILL і клієнтське обслуговування можуть використовуватися в поєднанні з мовними доповнюванням з транзакцією SMLT. Також функцію обслуговування клієнтів можна запустити в транзакції SMLT, обравши Language Special actions Client Maintenance.  
Поєднання RSREFILL або функції обслуговування клієнтів з програмою підтримки мови має наступні наслідки для кожної конкретної клієнтської таблиці настроювання:  
1. Будь-які попередні доповнення скидаються.

1. Виконується RREFILL або функція обслуговування клієнтів.
2. Таблиця доповнена.

Це дозволяє уникнути будь-яких проблем, викликаних порядком, в якому запускається RSREFILL, функція обслуговування клієнтів і програма підтримки мови.

**Ручний переклад**

Ручний переклад зв'язок між середовищем перекладу та системою зміни та транспорту дозволяє створювати перекладені настройки текстів у певній системі, записувати їх у транспортних запитах та поширювати у системному ландшафті. Перекласти решту Зберігаючих текстів у транзакції SE63. Запишіть переклади в транспортних запитах, а потім використайте систему управління транспортом (TMS), щоб розподілити ці запити в системному ландшафті.

**SPRO**

SAP SPRO означає SAP Reference Project Object, SPRO — код транзакцій, який використовується для налаштування всіх модулів SAP R/3 відповідно до вимог клієнта.

Введіть код операції SPRO в командному полі SAP і натисніть клавішу ENTER для виконання процесу впровадження SAP IMG.

SAP Transaction Code SPRO

У наступному вікні налаштування: Виконати проект, доступні три параметри

SAP Reference IMG

IMG Information

Project Analysis.

За допомогою вікна виконання проекту ви можете керувати існуючими проектами і створювати нові проекти відповідно до вимог клієнта.

**SIMGH**

Як відомо, він використовується в компоненті SAP BC-CUS (Customizing in Basis), який надходить за модулем BC (BASIS).SIMGH є кодом транзакцій, що використовується для обслуговування структури IMG в SAP.

**26. FMs and their types**

<https://help.sap.com/docs/SAP_NETWEAVER_731_BW_ABAP/cfae740a0a21455dbe6e510c2d86e36a/9fdb988735c111d1829f0000e829fbfe.html?version=7.31.22&locale=en-US>

<https://www.dan852.com/sap-abap-function-modules/>

**Funciton modules** - це процедури, які визначаються тільки в спеціальних програмах АВАР, так званих функціональних групах, але можуть бути викликані з усіх програм АВАР. Функціональні групи діють як контейнери для додатків функцій, які логічно належать один одному. Функціональні модулі дозволяють інкапсулювати і повторно використовувати глобальні функції в системі SAP. Вони управляються в центральній функціональній бібліотеці. Система SAP містить кілька попередньо визначених модулів функцій, які можна викликати з будь-якої програми ABAP. Функціональні модулі також відіграють важливу роль під час оновлення та у взаємодії між різними системами SAP, або між системами SAP та віддаленими системами за допомогою віддаленого зв'язку.

**Types of the FMs**

**Regular Function Module**

Regular Function Module - це дефолтний параметр. Цей вид Function Module виконується негайно і синхронно на вашій поточній системі SAP.

CALL FUNCTION func { parameter\_list | parameter\_tables }.

**Remote-Enabled Function Module**

Віддалені функціональні модулі можуть викликатися іншими системами SAP та іншими системами, що використовують протокол RFC.

Наприклад, можна визначити віддалений функціональний додаток у системі EWM, який потім викликається вашою системою ERP для отримання додаткової інформації про рівні запасів або дані про постачання.

CALL FUNCTION func DESTINATION dest parameter\_list.

Ця інструкція синхронно викликає цільову систему, вказану в dest. Доступні призначення RFC налаштовуються за допомогою транзакції SM59. Використовуйте унікальну назву призначення для параметра dest.

Програма продовжує виконання після завершення RFC. Залежно від підключення, це може бути проблемою продуктивності.

**Update Function Module**

Update Function Module не виконуються негайно. Вони заплановані для виконання в процесі роботи оновлення. Цей процес оновлення запускається інструкцією COMMIT WORK. Інструкція ROLLBACK WORK вилучає всі реєстрації додатка функції оновлення.

CALL FUNCTION update\_function IN UPDATE TASK

[EXPORTING p1 = a1 p2 = a2 ...]

[TABLES t1 = itab1 t2 = itab2 ...].

Використання функції CALL.. IN UPDATE TASK - це спосіб об'єднання змін бази даних в єдину базу даних LUW. Це також вигідно для продуктивності, оскільки оновлення відбувається асинхронно.

**27. Reports and their types**

Існує 2 види звітів.

Вони є: Інтерактивний звіт

Класичні звіти

У класичних звітах ми бачимо вихідні дані в одному списку, тоді як в інтерактивних звітах ми бачимо результат у кількох списках.

У ABAP всього 7 типів звітів.

Вони є:

Classical

Interactive

Logical Database

ABAP query

ALV Reports (ALV stands for ABAP List Viewer)

Report Writer/Report Painter

Views (There are different types of views also)

1. **Variables definition**

(https://www-tutorialspoint-com.translate.goog/sap\_abap/sap\_abap\_variables.htm?\_x\_tr\_sl=en&\_x\_tr\_tl=uk&\_x\_tr\_hl=ru&\_x\_tr\_pto=wapp#:~:text=Advertisements,the%20help%20of%20ABAP%20statements.)

Змінні — це іменовані об’єкти даних, які використовуються для зберігання значень у межах відведеної для програми області пам’яті. Як видно з назви, користувачі можуть змінювати вміст змінних за допомогою операторів ABAP. Кожна змінна в ABAP має певний тип, який визначає розмір і розташування пам'яті змінної; діапазон значень, які можуть зберігатися в цій пам'яті; і набір операцій, які можна застосувати до змінної.

Ви повинні оголосити всі змінні, перш ніж їх можна буде використовувати. Основною формою оголошення змінної є −

DATA <f> TYPE <type> VALUE <val>.

Тут <f> вказує ім'я змінної. Ім’я змінної може містити до 30 символів. <type> визначає тип змінної. Будь-який тип даних з повністю визначеними технічними атрибутами відомий як <type>. <val> визначає початкове значення змінної <f>. Якщо ви визначаєте елементарну змінну фіксованої довжини, оператор DATA автоматично заповнює значення змінної початковим значенням, специфічним для типу. Інші можливі значення для <val> можуть бути літералом, константою або явним реченням, наприклад Is INITIAL.

Нижче наведені дійсні приклади оголошення змінних.

DATA d1(2) TYPE C.

DATA d2 LIKE d1.

DATA minimum\_value TYPE I VALUE 10.

У наведеному вище фрагменті коду d1 — це змінна типу C, d2 — змінна типу d1, а мінімальне\_значення — це змінна цілого типу ABAP I.

У цьому розділі пояснюються різні типи змінних, доступні в ABAP. У ABAP є три типи змінних −

* Статичні змінні
* Довідкові змінні
* Системні змінні

## Статичні змінні

* Статичні змінні оголошуються в підпрограмах, функціональних модулях і статичних методах.
* Термін життя пов'язаний з контекстом декларації.
* За допомогою оператора 'CLASS-DATA' ви можете оголошувати змінні всередині класів.
* Оператор 'PARAMETERS' можна використовувати для оголошення елементарних об'єктів даних, які пов'язані з полями введення на екрані вибору.
* Ви також можете оголосити внутрішні таблиці, які пов’язані з полями введення, на екрані вибору за допомогою оператора SELECT-OPTIONS.

Нижче наведено умовні угоди, які використовуються під час іменування змінної −

* Ви не можете використовувати спеціальні символи, такі як "t" і "," для іменування змінних.
* Ім'я попередньо визначених об'єктів даних не можна змінити.
* Ім'я змінної не може збігатися з будь-яким ключовим словом або реченням ABAP.
* Ім'я змінних повинно передавати значення змінної без необхідності додаткових коментарів.
* Дефіси зарезервовано для позначення компонентів структур. Тому ви повинні уникати дефісів в іменах змінних.
* Символ підкреслення можна використовувати для розділення складних слів.

Ця програма показує, як оголосити змінну за допомогою оператора PARAMETERS −

REPORT ZTest123\_01.

PARAMETERS: NAME(10) TYPE C,

CLASS TYPE I,

SCORE TYPE P DECIMALS 2,

CONNECT TYPE MARA-MATNR.

Тут NAME представляє параметр із 10 символів, CLASS визначає параметр цілого типу з розміром за замовчуванням в байтах, SCORE представляє параметр упакованого типу зі значеннями до двох знаків після коми, а CONNECT відноситься до типу MARA-MATNF словника ABAP .

Наведений вище код дає наступний вихід −

## Довідкові змінні

Синтаксис оголошення посилання-змінних -

DATA <ref> TYPE REF TO <type> VALUE IS INITIAL.

* Додавання REF TO оголошує посилання-змінну ref.
* Специфікація після REF TO визначає статичний тип опорної змінної.
* Статичний тип обмежує набір об'єктів, на які може посилатися <ref>.
* Динамічний тип опорної змінної — це тип даних або клас, до якого вона зараз посилається.
* Статичний тип завжди більш загальний або такий же, як і динамічний тип.
* Додавання TYPE використовується для створення зв’язаного типу посилання та як початкове значення, і лише IS INITIAL можна вказати після додавання VALUE.

### Приклад

CLASS C1 DEFINITION.

PUBLIC SECTION.

DATA Bl TYPE I VALUE 1.

ENDCLASS. DATA: Oref TYPE REF TO C1 ,

Dref1 LIKE REF TO Oref,

Dref2 TYPE REF TO I .

CREATE OBJECT Oref.

GET REFERENCE OF Oref INTO Dref1.

CREATE DATA Dref2.

Dref2→\* = Dref1→\*→Bl.

* У наведеному вище фрагменті коду оголошуються посилання на об’єкт Oref і дві змінні посилання на дані Dref1 і Dref2.
* Обидві опорні змінні даних повністю введені та можуть бути розіменовані за допомогою оператора розіменування →\* на позиціях операнда.

## Системні змінні

* Системні змінні ABAP доступні з усіх програм ABAP.
* Ці поля фактично заповнені середовищем виконання.
* Значення в цих полях вказують на стан системи в будь-який момент часу.
* Повний список системних змінних можна знайти в таблиці SYST в SAP.
* До окремих полів структури SYST можна отримати доступ за допомогою «SYST-» або «SY-».

### Приклад

REPORT Z\_Test123\_01.

WRITE:/'SY-ABCDE', SY-ABCDE,

/'SY-DATUM', SY-DATUM,

/'SY-DBSYS', SY-DBSYS,

/'SY-HOST ', SY-HOST,

/'SY-LANGU', SY-LANGU,

/'SY-MANDT', SY-MANDT,

/'SY-OPSYS', SY-OPSYS,

/'SY-SAPRL', SY-SAPRL,

/'SY-SYSID', SY-SYSID,

/'SY-TCODE', SY-TCODE,

/'SY-UNAME', SY-UNAME,

/'SY-UZEIT', SY-UZEIT.

Наведений вище код дає наступний вихід −

SY-ABCDE ABCDEFGHIJKLMNOPQRSTUVWXYZ

SY-DATUM 12.09.2015

SY-DBSYS ORACLE

SY-HOST sapserver

SY-LANGU EN

SY-MANDT 800

SY-OPSYS Windows NT

SY-SAPRL 700

SY-SYSID DMO

SY-TCODE SE38

SY-UNAME SAPUSER

SY-UZEIT 14:25:48

**29. Fieled-symbols. Assignment, unassignment, setting of values, reading of values.**

**Символи полів (Field symbols)** – це заповнювачі або символічні імена для існуючих об'єктів даних (або частин існуючих об'єктів даних), оголошених оператором FIELD-SYMBOLS або оператором оголошення FIELD-SYMBOL. Область пам'яті призначається символу поля під час виконання програми. Символ поля не резервує фізичний простір в області даних програми, як у випадку з об’єктом даних. Замість цього він працює як динамічне ім'я області пам'яті, де розташований конкретний об'єкт даних або частина об'єкта. Символ поля може використовуватися замість об’єктів даних у позиціях операнда операторів. Коли виконується оператор цього типу, область пам’яті повинна бути призначена для символу поля за допомогою оператора ASSIGN або доповнення ASSIGNING при обробці внутрішніх таблиць.

Символи полів можуть вказувати майже на будь-який об'єкт даних. Символ поля є еквівалентом змінної посилання на дані, розіменованої за допомогою ->\*. На відміну від опорних змінних даних, семантика значень використовується для доступу до символу поля. Доступ до символу поля подібний до доступу до названого об’єкта даних (або частини об’єкта). Однак для опорних змінних даних застосовується семантика посилання.

Символи полів реалізовані як для загальної роботи з об'єктами даних, так і для явного приведення об'єктів даних (можливе лише використання символів полів).

Іноді може бути корисно скинути символ поля. Це можна зробити за допомогою UNASSIGN.

(<https://help.sap.com/doc/abapdocu_751_index_htm/7.51/en-us/abenabap_field_symbols.htm#:~:text=Field%20symbols%20are%20placeholders%20or,field%20symbol%20at%20program%20runtime>.

<https://help.sap.com/doc/saphelp_scm700_ehp01/7.0.1/en-US/fc/eb3860358411d1829f0000e829fbfe/frameset.htm>)

**30. Parameters and select-options**

Параметри та опції вибору

Параметри — це компоненти екрана вибору, яким у програмі ABAP призначається глобальний елементарний об’єкт даних і поле введення на екрані вибору. Ім’я параметра може містити не більше восьми символів.

Select-Options — це оператор, який використовується для визначення двох полів введення, щоб користувачі могли вводити діапазон значень

1. **Selection Screens and their events:**

**INITIALIZATION, AT SELECTION-SCREEN OUTPUT, AT SELECTION-SCREEN, START OF SELECTION, etc.**

( <https://help.sap.com/SAPhelp_nw73/helpdata/en/4a/43c40d5a503f04e10000000a421937/content.htm?no_cache=true> )

*Екрани вибору (Selection screens)* — це спеціальні екрани(screens), які визначаються за допомогою операторів ABAP. Оскільки програмісти не мають доступу до логіки потоку екранів вибору, вони не можуть визначити діалогові модулі для екранів вибору. Середовище виконання ABAP повністю контролює процес обробки екранів вибору. Щоб дозволити програмістам змінювати екран вибору перед його викликом (PBO) і реагувати на дії користувача на екрані вибору (PAI), середовище виконання ABAP генерує ряд спеціальних подій екрана вибору до відображення екрана вибору та після того, як користувач виконав дії на екрані вибору. Програмісти можуть визначити блоки подій у програмі, щоб реагувати на ці події.

Основною формою подій екрана вибору є AT SELECTION-SCREEN подія. Ця подія відбувається після того, як середовище виконання передало всі вхідні дані з екрана вибору до програми ABAP. Інші події екрана вибору дозволяють програмістам змінювати екран вибору перед його відправкою і, зокрема, перевіряти введення користувача.

Обробка екрана вибору починається після INITIALIZATION події з AT SELECTION SCREEN OUTPUT. Екран вибору потім надсилається на екран. Дії користувача на екрані вибору призводять до інших подій, які або використовуються для довідки по полях або можливим записам, або які запускають обробку PAI на екрані вибору. Під час обробки PAI повідомлення про помилки у відповідних блоках подій дозволяють користувачам повернутися до екрана вибору. Лише якщо AT SELECTION-SCREEN подія завершується належним чином, тобто не через повідомлення про помилку, інші події виконуваної програми запускаються, починаючи з START-OF-SELECTION.

**32. Custom Screens and main events: PBO, PBI** (<https://drive.google.com/file/d/1h3SvRwKEOIQznlNnlS7qlIZ4tCcVtFqL/view?usp=sharing> ст.229 в файлі, <https://www.stechies.com/what-is-pbo-and-pai-events/>)

Process before output (PBO, обработка до вывода) — событие, которое обрабатывается до отображения экрана и обычно используегся для инициализации экранных полей.

Process after input (PAI. обработка после ввода) — событие, которое обрабатывается при выборе функции меню, при нажатии экранной кнопки или функциональной клавиши.

PBO (Process Before Output) визначає логіку потоку або виконує модулі перед відображенням екрана. Ця подія використовується для виведення на екран полів програми. Користувач може створити статус GUI в PBO.

PAI (Process After Input) визначає логіку потоку, що виконує модулі після відображення екрана та після отримання введених даних від Користувача. Ця подія використовується для написання коду для дій Користувача на екрані. У PAI дані переміщуються від екрана до програми.

1. **ABAP statement ‘READ TABLE … WITH KEY’**

Визначення ключа таблиці як ключа пошуку. Можна вказати або первинний ключ таблиці, або додатковий ключ таблиці. Значення можуть бути оголошені або неявно в робочій області wa позаду FROM або шляхом явного перерахування компонентів ключа таблиці за TABLE KEY .

Коли використовується первинний ключ таблиці, доступ до категорій таблиці здійснюється наступним чином:

* Пошук у стандартних таблицях здійснюється лінійно.
* Для відсортованих таблиць використовується двійкове сканування.
* Для хешованих таблиць використовується хеш-алгоритм .

Коли використовується вторинний ключ таблиці, у випадку відсортованого ключа використовується двійкове сканування, а у випадку хеш-ключа використовується алгоритм хешування .

Коли рядок знайдено, системне поле sy-tabix встановлюється відповідно до використовуваного ключа таблиці:

* Для відсортованих ключів встановлюється кількість рядків, знайдених у пов’язаному індексі таблиці.
* Для хеш-ключів встановлюється значення 0.

Якщо рядок не знайдено, а у випадку відсортованих ключів, sy-tabix встановлюється на номер рядка запису в асоційованому індексі таблиці, перед яким рядок буде вставлено за допомогою INSERT ... INDEX ... , щоб зберегти сорт.

**34. ABAP Statements 'LOOP AT... INTO' and 'LOOP AT... ASSIGNING'**

LOOP AT … INTO – призначена для читання більше ніж одного запису таблиці (internal table).

Синтаксис: **LOOP AT itab INTO wa.**

itab – internal table, wa – working area.

Приклад (виводить на екран поля STRUC-NAME, STRUC-SALDO з ITAB):

LOOP AT ITAB INTO STRUC.

WRITE: / STRUC-NAME, STRUC-SALDO.

ENDLOOP.

LOOP AT... ASSIGNING – призначена для роботи з даними внутрішньої таблиці безпосередньо, тобто дії всередині циклу виконуються над таблицею.

Приклад:

LOOP AT ITAB ASSIGNING <FS>.

WRITE: / <FS>-COL1, <FS>-COL2.

ENDLOOP.

**35. ABAP Statements 'INSERT ... INTO TABLE ... ' and 'INSERT LINES OF ... INTO TABLE ... '**

**(https://www.saphub.com/abap-tutorial/inserting-lines-into-abap-internal-tables/)**

Ми можемо вставити один або декілька рядків до внутрішніх таблиць ABAP за допомогою ОПЕРАТОРА INSERT. Щоб вставити одиничний рядок, спочатку розташуйте значення, які ми хочемо вставити в робочу область і скористайтеся твердженням INSERT, щоб вставити значення в робочу область у внутрішню таблицю.

Синтаксис для додавання рядка до внутрішньої таблиці

INSERT <work area> INTO TABLE <internal table>.

OR

INSERT <work area> INTO <internal table> INDEX <index>.

Ми також можемо вставити декілька рядків до внутрішньої таблиці з одним ТВЕРДЖЕННЯМ INSERT тобто ми можемо вставити рядки однієї внутрішньої таблиці до іншої внутрішньої таблиці.

Синтаксис для додавання декількох рядків до внутрішньої таблиці

INSERT LINES OF <itab1> [FROM <index 1>] [TO <index 2>] INTO TABLE <itab2>.

OR

INSERT LINES OF <itab1> [FROM <index 1>] [TO <index 2>] INTO

<itab2> INDEX <index>.

**36. ABAP Statement 'APPEND ... TO...'**

<https://www.tutorialscampus.com/sap-abap/append-statement.htm>

Оператор APPEND додає один рядок/рядок до існуючої внутрішньої таблиці. Оператор APPEND використовує робочу область для додавання інформації про рядок/рядок до таблиці. APPEND вставляє дані після останнього рядка внутрішньої таблиці. Робоча область може бути або рядком заголовка, або будь-яким іншим рядком з тією ж структурою, що і внутрішня таблиця.

APPEND [work-area /INITIAL LINE] TO itable .

* wrok-area - Визначає робочу область.
* itable - Вказує ім'я внутрішньої таблиці.

**37. Щоб переміщувати значення між компонентами структур, використовуйте оператор**

MOVE-CORRESPONDING <struct1> TO <struct2>.

Цей оператор переміщує вміст компонентів структури <struct1> до компонентів <struct2>, які мають ідентичні імена.

Коли він виконується, він розбивається на набір операторів MOVE, по одному для кожної пари полів з ідентичними іменами, таким чином:

MOVE STRUCT1-<ci> TO STRUCT2-<c i>.

**38. System variables (sy-subrc, sy-tabix, sy-uname, sy-batch, sy-datum, sy-timlo, etc.)**

(https://sapcodes.com/2015/11/17/abap-system-variables/)

#### **SY-CPAGE – Holds Current Page Number**

#### **SY-CPROG – Contains Program Name**

#### **SY-CUCOL – Cursor Position ( Column)**

#### **SY-CUROW – Cursor Position (Line)**

#### **SY-DATLO  –  Local Date for User**

#### **SY-DATUM  – System Date**

#### **SY-DATUD – Global Date Related To UTC ( GMT )**

#### **SY-DBCNT – Number of entries read by DB Operation**

#### **SY-DBNAM- Logical DB for ABAP/4 Program**

#### **SY-DBSYS –  System DB System**

#### **SY-DYNNR- Number of Current Screen**

#### **SY-INDEX –  Number Of Loop passes**

#### **SY-LANGU- SAP Log on Language Key**

#### **SY-LILLI  – Number Of Current List Line**

#### **SY-LINCT – Number Of List Lines**

#### **SY-LINNO – Current Line for List Creation**

#### **SY-LINSZ – Line Size of List**

#### **SY-LISEL – Selected Line For Interactive List**

#### **SY-LSIND – Number Of Secondary List**

#### **SY-MANDT – Client Number For SAP Log On**

#### **21.SY-MSGID  – Message ID**

#### **SY-MSGTY – Message Type**

#### **SY-MSGNO –  Message Number**

#### **SY-MSGV1 – Message Variable1**

#### **SY-MSGV2 – Message Variable2**

#### **SY-MSGV3 – Message Variable3**

#### **SY-MSGV4 – Message Variable4**

#### **SY-OPSYS – Operating System**

#### **SY-PAART –  Print Format**

#### **SY-PAGCT – Page Size of List in Report Program**

#### **SY-PAGNO – Current Page in the List**

#### **SY-PDEST – Print Output Device**

#### **SY-PFKEY – Run time Current F key Status**

#### **SY-REPID – Report Name in ABAP**

#### **SY-SAPRL –  SAP Release**

#### **SY-SUBRC – Return Value after specific ABAP Statement**

#### **SY-SYSID – SAP system ID**

#### **SY-TABIX  –  Current line of Internal Table**

#### **SY-TCODE – Current Transaction Code**

#### **SY-TIMLO – Local Time Of User**

#### **SY-TITLE  – Title of ABAP Program**

#### **SY-TMAXL – Maximum entries in a Internal Table**

#### **SY-TNAME – Name Of Internal Table after first Access**

#### **SY-UCOMM -Interact : Command field Function entry / Function code**

#### **SY-ULINE  – Underline**

#### **SY-UNAME – Session : SAP user from Log On**

#### **SY-UZEIT  – System Time**

#### **SY-VLINE  – Vertical Bar**

**39. Basic rules of work with exceptions.**

**(**Я ХУЙ ЗНАЄ, ЩО САМЕ МАЄТЬСЯ НА УВАЗІ, ТОМУ НИЖЧЕ **ДВА** ВАРІАНТИ, ОБИРАЙТЕ САМІ**)**

**Варіант 1**

З міркувань сумісності є два варіанти визначення окремих оброблюваних винятків у ABAP:

* **Класичні винятки (Classic exceptions)**

Ці винятки можуть бути оголошені лише в інтерфейсах методів або функціональних модулів за допомогою EXCEPTIONS і можуть бути викликані в рамках такої процедури за допомогою операторів RAISE або MESSAGE RAISING. Викликаючий процедуру може використовувати додавання EXCEPTIONS операторів meth( ... ) або CALL FUNCTION, щоб призначити коди повернення для системного поля sy-subrc виняткам, які викликається, і оцінити їх після виклику.

* **Винятки на основі класу (Class-Based Exceptions)**

Ці винятки визначаються класами винятків, з яких може бути створений об’єкт винятку, коли виникає виняток (якщо обробник використовує додавання INTO у CATCH). Виняток на основі класу може або скасувати поточний контекст, або дозволити створити резюме. Винятки виникають за допомогою оператора RAISE EXCEPTION і обробляються за допомогою CATCH у структурі керування TRY. Винятки, засновані на класах, можуть бути викликані будь-якими процедурами і можуть поширюватися будь-якими процедурами.

Співіснування двох концепцій винятків регулюється таким чином:

* Класичні винятки та винятки на основі класів не можуть бути оголошені разом в інтерфейсі процедури. У блоці обробки можуть бути викликані лише класичні або лише класові винятки.
* З міркувань сумісності в рамках блоку обробки винятки на основі класів можна обробляти та оцінювати повернуті значення функціональних модулів і методів за допомогою класичних винятків.

**Правила**:

Використовуйте винятки на основі класів.

Only raise class-based exceptions in new procedures if it is possible to dispense with classic exceptions from the technical point of view.

(<https://help.sap.com/doc/abapdocu_751_index_htm/7.51/en-us/abenclass_exception_guidl.htm>)

**Варіант 2**

**Обробка винятків.** Якщо виникають помилки, створюються робочі елементи, пов’язані із завданнями SAP. Ці робочі елементи відображаються на робочому місці для бізнесу вибраного одержувача. Завдання визначаються кодами процесу.

**Обробка за допомогою SAP tasks**

Завдання для інтерфейсу IDoc поділяються на вхідну та вихідну обробку. Під час вхідної обробки можуть виникнути формальні помилки, наприклад, якщо профіль партнера для контрольного запису не знайдено. Ви також можете активувати перевірку синтаксису для IDocs у профілях партнера. Помилка, яка виникає під час такої перевірки синтаксису, призначається спеціальному стандартному завданню (див. також Обробка винятків: процедура).

Обробка винятків в інтерфейсі IDoc покращена у вхідній обробці за допомогою обробки винятків у програмі, яка може відповідати на статус «51» (документ програми не опубліковано).

Одержувач робочого елемента може потім виправити помилку і знову запустити перетворення отриманих IDoc або позначити IDoc для видалення.

(<https://help.sap.com/docs/SAP_NETWEAVER_AS_ABAP_751_IP/8f3819b0c24149b5959ab31070b64058/4b42b79961384cb9e10000000a42189b.html?version=7.51.12&locale=en-US>**)**

**40. BAPIs and related ABAP syntax and statements**

BAPI та супровідний ABAP синтаксис

BAPI реалізовані як функціональні модулі, які викликають внутрішній код SAP. Залежно від того, який набір BAPI використовується, вони можуть викликати моделі бізнес-об’єктів, визначені за допомогою Business Objects Processing Framework (BOPF), або застарілі моделі, визначені за допомогою програм, таблиць і функціональних модулів.

1. **Classes and interfaces**

( <https://help.sap.com/doc/abapdocu_752_index_htm/7.52/en-US/abenclass_interface_definition.htm> )

Класи та інтерфейси в об'єктах ABAP можуть бути оголошені глобально або локально.

* Глобальні класи та інтерфейси зберігаються в центральній [бібліотеці](javascript:call_link('abenclass_library_glosry.htm')) класів [репозиторію](javascript:call_link('abenrepository_glosry.htm')) . З технічної точки зору глобальні класи та інтерфейси визначаються в [пулах класів](javascript:call_link('abenclass_pool_glosry.htm')) або пулах [інтерфейсів](javascript:call_link('abeninterface_pool_glosry.htm')) . Усі програми ABAP в [AS ABAP](javascript:call_link('abensap_nw_abap_glosry.htm')) можуть отримати доступ до цих глобальних класів та інтерфейсів. Доступом керує перевірка пакета. Глобальні класи та інтерфейси зберігаються в просторі імен разом із [типами даних](javascript:call_link('abendata_type_glosry.htm')) словника [ABAP](javascript:call_link('abenabap_dictionary_glosry.htm')) .
* Локальні класи та інтерфейси можна визначити в усіх програмах, крім [пулів інтерфейсів](javascript:call_link('abeninterface_pool_glosry.htm')) і [груп типів](javascript:call_link('abentype_group_1_glosry.htm')) . Їх можна використовувати статично лише у програмі визначення. Динамічний доступ з інших програм можливий, але не бажаний. Коли клас використовується в програмі ABAP, система спочатку шукає локальний клас із зазначеним іменем. Якщо вона не знаходить його, то шукає глобальний клас. В іншому випадку немає різниці між використанням глобальних і локальних класів або інтерфейсів.

У локальних класах, які використовуються лише в одній програмі, зазвичай достатньо визначити загальнодоступний інтерфейс (тобто видимі зовні компоненти), щоб він вписувався в цю програму. І навпаки, кожен глобальний клас доступний у всій системі, а це означає, що його відкритий інтерфейс можна вказати лише з посиланням на типи даних, які самі є видимими в системі.

Синтаксис визначення класів та інтерфейсів по суті однаковий для локальних і глобальних класів та інтерфейсів. Єдина відмінність полягає в додаванні **PUBLIC** , яке розрізняє глобальні класи та інтерфейси та локальні оголошення.

Класи складаються з вихідного коду ABAP , укладеного в оператори ABAP CLASS-ENDCLASS . Повне визначення класу складається з частини оголошення і, якщо потрібно, частини реалізації.

Частина оголошення класу з назвою class складається з блоку оператора:

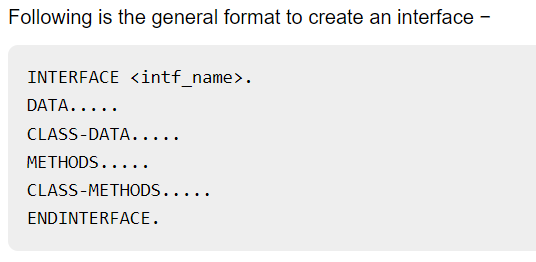
[**CLASS class DEFINITION.**](javascript:call_link('abapclass.htm'))  
...  
[**ENDCLASS.**](javascript:call_link('abapendclass.htm'))

Визначення інтерфейсу intf міститься в таких заявах:

[**INTERFACE intf.**](javascript:call_link('abapinterface.htm'))  
...  
[**ENDINTERFACE.**](javascript:call_link('abapendinterface.htm'))

**42. Implementation of interfaces(**[**https://www.tutorialscampus.com/sap-abap/interfaces.htm**](https://www.tutorialscampus.com/sap-abap/interfaces.htm))

Інтерфейси використовуються, коли два подібних класи мають метод з однаковою назвою, але функціональні можливості відрізняються один від одного. Інтерфейси можуть виглядати схожими на класи, але функції, визначені в інтерфейсі, реалізовані в класі, щоб розширити область дії цього класу. Інтерфейси разом із ознакою успадкування забезпечують основу для поліморфізму. Це тому, що метод, визначений в інтерфейсі, може вести себе по-різному в різних класах.



У цьому синтаксисі <intf\_name> представляє назву інтерфейсу. Операції DATA і CLASSDATA можна використовувати для визначення екземплярів і статичних атрибутів інтерфейсу відповідно. Оператори METHODS і CLASS-METHODS можна використовувати для визначення екземплярів і статичних методів інтерфейсу відповідно. Оскільки визначення інтерфейсу не включає клас реалізації, немає необхідності додавати речення DEFINITION в оголошення інтерфейсу.

Усі методи інтерфейсу є абстрактними. Вони повністю оголошені, включаючи їхній інтерфейс параметрів, але не реалізовані в інтерфейсі. Усі класи, які хочуть використовувати інтерфейс, повинні реалізувати всі методи інтерфейсу. В іншому випадку клас стає абстрактним.

У цьому синтаксисі <intf\_name> представляє назву інтерфейсу. Зауважте, що цей синтаксис необхідно використовувати в загальнодоступному (public) розділі класу.

**43. Calling of methods and attributes**

*Виклик метода:* CALL METHOD [method] …

Спосіб, яким ви звертаєтесь до методу, залежить від самого методу та від того, звідки ви його викликаєте. У частині реалізації класу ви можете викликати методи того самого класу безпосередньо, використовуючи їх ім’я.

CALL METHOD [method]...

Поза класом видимість методу залежить від того, чи можна його взагалі викликати. Методи видимого екземпляра можна викликати поза класом за допомогою

CALL METHOD ref->[method]...

де ref є опорною змінною, значення якої вказує на екземпляр класу. Методи видимого екземпляра можна викликати поза класом за допомогою

CALL METHOD =>[method]...

*Виклик атрибута* відбувається звичайним прописом його назви у коді.

**44. Access to methods and attributes (private, protected, public)(документація сапу на сайті)**

Public Section - Усі компоненти, оголошені в загальнодоступному розділі, доступні всім користувачам класу, а також методам класу та будь-яким класам, які від нього успадковують.

Protected Section - Усі компоненти, оголошені в захищеному розділі, доступні для всіх методів класу та класів, які від нього успадковують.

Private Section - Компоненти, які ви оголошуєте в приватному розділі, видимі лише в методах того самого класу.

**45. Static and instance methods and attributes. Call of them**

**(**<https://help.sap.com/saphelp_nw73/helpdata/en/4a/0c247779281b40e10000000a42189c/content.htm?no_cache=true>

**https://help.sap.com/doc/abapdocu\_752\_index\_htm/7.52/en-us/abenstatic\_method\_glosry.htm)**

Метод класу, який може бути використаний незалежно від екземпляра класу. Оголошується за топомогою CLASS METHODS. Статичні методи можуть отримати доступ тільки до статичних атрибутів і статичних подій власного класу.

**Syntax**

CLASS-METHODS meth *[*[DEFAULT IGNORE*|*FAIL](javascript:call_link('abapmethods_default.htm'))*]*  
  *[*IMPORTING [parameters](javascript:call_link('abapmethods_parameters.htm')) *[*PREFERRED PARAMETER p*]]*  
  *[*EXPORTING [parameters](javascript:call_link('abapmethods_parameters.htm'))*]*  
  *[*CHANGING  [parameters](javascript:call_link('abapmethods_parameters.htm'))*]*  
  *[{*RAISING exc1*|*RESUMABLE(exc1) exc2*|*RESUMABLE(exc2) ...*}*  
  *|{*EXCEPTIONS exc1 exc2 ...*}]*.

Атрибути є об'єктами даних класу. Також можливе оголошення об'єктів даних класу. Декларація атрибутів і типів даних відбувається з використанням загальнИХ твердження, які також можливі в інших контекстах:

* [**TYPES**](javascript:call_link('abaptypes.htm'))
* [**DATA**](javascript:call_link('abapdata.htm'))
* [**CONSTANTS**](javascript:call_link('abapconstants.htm'))

спеціальне твердження, яке можливе лише в класах:

* [**CLASS-DATA**](javascript:call_link('abapclass-data.htm'))

**Можна назвати статичним методом глобального класу об'єктів ABAP наступним чином:**

<tt:call-method class="class" [s-|d-]name="meth"

[writer = "writer\_para"]

[reader = "reader\_para"]>

[<tt:with-parameter [s-|d-]name="para1" [ref="node1"|val="val1"|var="var1"] />

<tt:with-parameter [s-|d-]name="para2" [ref="node2"|val="val2"|var="var2"] />

...]

</tt:call-method>

**46. Constructor and class-constructor**

<https://help.sap.com/doc/abapdocu_752_index_htm/7.52/en-us/abenconstructor.htm>

**Constructors**  - це спеціальні методи, які створюють визначений початковий стан для об'єктів і класів. Стан об'єкта визначається його екземплярами атрибутів і статичними атрибутами. Вміст можна присвоїти атрибутам за допомогою додавання VALUE оператора DATA. Конструктори необхідні, коли початковий стан об'єкта повинен бути визначений динамічно.

Існує два типи конструкторів: конструктори екземплярів і статичний конструктор.

Кожен клас має один **конструктор екземплярів**. Це заздалегідь визначений екземпляр методу класу, який називається **Constructor.** Щоб використовувати конструктор екземплярів, конструктор методу повинен бути оголошений в розділі видимості класу за допомогою оператора METHOD, і реалізований в частині реалізації. У глобальних класах конструктор екземплярів може бути оголошений тільки в області публічної видимості з технічних причин. У локальних класах розділ видимості, в якому конструктор екземплярів може бути оголошений, повинен бути більш загальним або рівним нестабільності, визначеної додаванням CREATE оператора CLASS. Тут рекомендується найбільш спеціалізована зона. Якщо він не оголошений явно, конструктор екземплярів є неявним методом, який успадковує і отримує доступ до інтерфейсу від конструктора екземплярів в суперкласі.

Кожен клас має єдиний статичний конструктор. Це заздалегідь визначений публічний статичний метод класу **class-constructor**. Для використання статичного конструктора статичний метод class\_constructor повинен бути оголошений в загальнодоступному розділі декларативної частини класу за допомогою оператора CLASS-METHODS і реалізований в частині реалізації. Статичний конструктор не має параметрів інтерфейсу і не може підняти винятки. Якщо не оголошено явно, статичний конструктор є просто порожнім методом.

Статичний конструктор викликається один раз для кожного класу і внутрішнього сеансу. Статичний конструктор класу викликається автоматично перед першим доступом до класу. Клас доступний, коли створюється екземпляр класу або використовується компонент класу (за винятком типів і констант). Не обов'язково виконувати статичний конструктор для доступу до типу або константи класу.

**47. Events and their listeners.**

Event– це набір результатів, визначених у класі, для запуску обробників подій в інших класах. Коли событие ініціюється, ми можемо викликати будь-яку кількість методів обробки подій. Зв'язок між триггером і його методом-обробником фактично визначається динамічно під час виконання. При звичайному виклику метод викликає програму визначає, який метод об'єкта або клас необхідно викликати. Оскільки метод фіксованого обробника не реєструється для кожного події, у випадку обробки події метод обробника визначає суспільство, яке повинно бути ініціовано. Event класу може викликати інструкції метод-обробник подій того ж класу за допомогою RAISE EVENT. Для подій метод обробки події може бути визначений в тому ж або іншому класі за допомогою пропозицій FOR EVENT, як показано в наступному синтаксисі:

FOR EVENT <event\_name> OF <class\_name>.

Подібне методам класу, супроводження може мати параметри інтерфейсу, але воно має тільки вихідні параметри. Вихідні параметри передаються в метод обробки подій за допомогою оператора RAISE EVENT, який отримує їх у якості вхідних параметрів.

Робота динамічного зв'язку з методом-обробником у програмі за допомогою оператора SET HANDLER. Коли буття ініціюється, передбачається, що відповідні методи-обробники подій виконуються у всіх класах обробки.

**48. Fabric pattern**

(https://abap--blog-ru.translate.goog/osnovy-abap/oop/fabrichnyj-metod/?\_x\_tr\_sl=ru&\_x\_tr\_tl=uk&\_x\_tr\_hl=ru&\_x\_tr\_pto=wapp)

#### Призначення

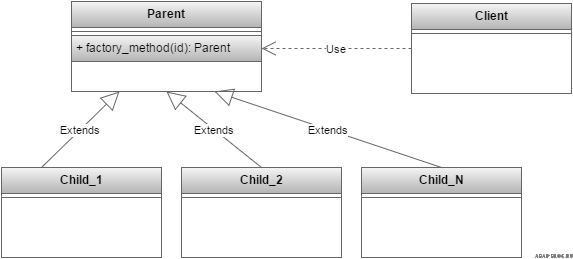
Паттерн вирішує завдання визначення деякого інтерфейсу створення класу, конкретну реалізацію якого визначають дочірні класи, що реалізують даний інтерфейс. На відміну від абстрактної фабрики, фабричний метод оперує створенням єдиного об'єкта, а чи не сімейства.

Цей патерн має кілька версій реалізації, ми розглянемо реалізацію через статичний метод у базовому класі. У такій реалізації статичний метод надходять деякі аргументи, на основі яких приймається рішення про створення того чи іншого спадкоємця базового класу.

Переваги:

* Позбавляє необхідності надмірного зв'язування клієнтського коду з дочірніми класами, тобто. місце, де створюються спадкоємці явно визначено у статичному методі базового класу, а чи не приховано десь у клієнта. Клієнт, як і у випадку з абстрактною фабрикою, не зав'язаний на конкретні реалізації, а оперує абстракцією.

#### Діаграма



Фабричний метод

**49. MVC pattern.**

**MVC (The Model View Controller) pattern** містить чітке розмежування між керуванням обробкою, моделлю даних та відображенням даних в інтерфейсі. Ці три області формально відрізняються один від одного трьома об’єктами: Модель, Вид і Контролер. В результаті ви можете легко розділити веб-додатки на логічні блоки.

Модель використовується як прикладний об'єкт адміністрування даних програми. Він відповідає на запити інформації про свій статус, які зазвичай надходять з представлення, а також на заяви про зміни статусу, які зазвичай надсилає контролер. Таким чином, лише модель використовується для внутрішньої обробки даних без посилання на програму та її інтерфейс користувача. Для моделі можуть бути різні види, які можна реалізувати за допомогою різних сторінок перегляду.

Подання обробляє графічний і текстовий вихід в інтерфейсі і, отже, представляє вхідні та вихідні дані в кожному елементі інтерфейсу, наприклад, кнопки, меню, діалогові вікна тощо. Погляд піклується про візуалізацію. Для візуалізації стану подання запитує модель, або модель інформує подання про можливі зміни статусу.

Контролер інтерпретує та відстежує дані, які вводяться користувачем за допомогою миші та клавіатури, змушуючи модель або вигляд пізніше змінюватися, якщо необхідно. Вхідні дані пересилаються і ініціюються зміни в даних моделі. Контролер використовує методи моделі для зміни внутрішнього стану, а потім повідомляє про це представлення. Таким чином контролер визначає реакції на введення користувача та контролює обробку.

Подання і контролер разом утворюють інтерфейс користувача.

Оскільки модель не розпізнає ні представлення, ні контролер, внутрішня обробка даних відокремлена від інтерфейсу користувача. Оскільки модель не розпізнає ні представлення, ні контролер, внутрішня обробка даних відокремлена від інтерфейсу користувача. Однак у вас також є можливість відображати дані в різних форматах; ви можете відобразити результати виборів у вигляді таблиці, гістограми або кругової діаграми.

(<https://help.sap.com/docs/SAP_NETWEAVER_AS_ABAP_752/0884fb55479542b38fb676e8f5b250a1/859ff12f-61bd-4085-9ccb-051cc3e816ba.html?locale=en-US&version=7.52.1>)

1. **Enhancement spot**

(

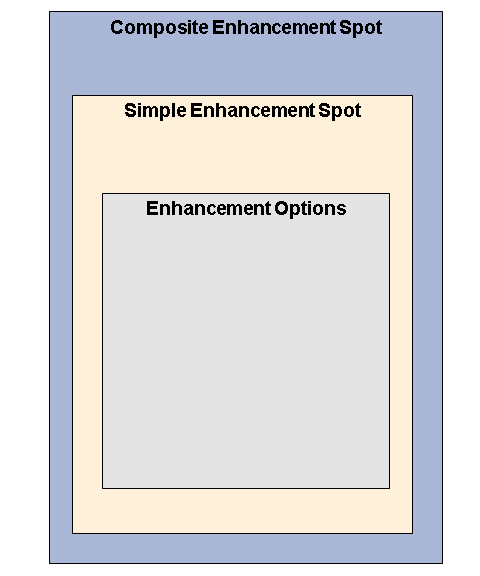
<https://help.sap.com/doc/saphelp_nw75/7.5.5/en-US/91/f1e540f8648431e10000000a1550b0/content.htm?no_cache=true> )

Точки покращення(Enhancement Spots) — це контейнери для явних параметрів покращення, що містять інформацію про позиції, на яких були створені параметри покращення. Визначення елемента точки покращення та відповідні виклики елементів плями покращення складають визначення явного параметра покращення (explicit enhancement option). Наприклад, під час редагування програми ABAP за допомогою редактора ABAP ви можете визначити явні параметри покращення у формі оператора ENHANCEMENT-POINT, який також представляє визначення елемента та виклик елемента.

Точки покращення поділяються на прості та складні(композиційні).

*Прості та складені точки покращення* – це об’єкти сховища, які утворюють деревоподібну структуру, де листя та гілки представляють прості та складені плями покращення відповідно. Проста точка покращення (simple enhancement spot) завжди призначається лише одній технології покращення (покращення вихідного коду ABAP або BAdI).

*Складні точки покращення* (Composite enhancement spots) використовуються для семантичного групування простих плям покращення. Складна точка покращення містить або одну або кілька простих точок покращення та/або одну чи більше складних точок покращення відповідного типу, як описано на малюнку нижче.

**

**50. Singleton pattern**

Singleton паттерн

Шаблон проектування singleton забезпечує створення лише одного екземпляра класу. Клас містить власний конструктор, і в цьому методі розміщена логіка, що забезпечує створення лише одного екземпляра.

**52. Enhancement option (**[**https://help.sap.com/doc/saphelp\_nw75/7.5.5/en-US/fb/e3d8403e37762ae10000000a155106/content.htm?no\_cache=true**](https://help.sap.com/doc/saphelp_nw75/7.5.5/en-US/fb/e3d8403e37762ae10000000a155106/content.htm?no_cache=true)**)**

Enhancement option (параметри покращення) – це позиції в об’єктах сховища, де можна вносити покращення. Ці параметри або явно визначаються (розробником), або надаються фреймворком.

Explicit enhancement options (Явні параметри покращення)

Явні параметри покращення визначаються розробником у центральній початковій системі. Удосконалення внесені в наступні системи. Наразі явні параметри покращення можуть бути визначені за допомогою:

* Явне позначення точок або розділів вихідного коду в програмах ABAP. Для покращення їх можна розширити або замінити плагінами вихідного коду.
* Включення бізнес-надбудов (BAdI) у програми ABAP. Ці програми потім доповнюються об’єктними плагінами із попередньо визначеними інтерфейсами.

Явні параметри покращення керуються точками покращення та розширюються за допомогою реалізацій покращення.

Неявні параметри покращення надаються фреймворком. Вони завжди існують і не вимагають посилення плям. Вони також покращуються реалізацією покращення. Неявні варіанти покращення:

* Конкретні параметри в програмах ABAP, наприклад, кінець програми, які можна розширити за допомогою плагінів вихідного коду. На початку та в кінці кожної форми, функціонального модуля, методу (глобального чи локального класу) і в кінці звіту чи включення та кожної локальної структури є точки неявного покращення. Що стосується методів локальних класів, то можна підвищити параметри всіх типів. Існують також неявні варіанти додати щось у визначальний розділ глобального класу або додати весь розділ (наприклад, захищений розділ), якщо він не існує.
* Інтерфейси параметрів функціональних модулів, які можна розширити за допомогою нових додаткових параметрів.
* Нові методи, атрибути та події, які можна додати до глобальних класів. Кожен метод глобального класу пропонує вихід до, після та перезапису, а також параметри для додавання нових необов’язкових параметрів.
* Покращивши об’єкт Web Dynpro, у вас з’явиться багато різних неявних опцій покращення, які пропонує фреймворк.

**53. Enhancement point**

Визначає позицію в програмі ABAP як опцію покращення, в яку можна вставити один або кілька плагінів вихідного коду.

**54. Enhancement section (документація сапу)**

Оператори enhancement перетворюють розділ поточної програми на явну точку покращення, в результаті чого розділ може бути замінений покращенням вихідного коду. Коли програма генерується, цей розділ програми замінює рівно один модуль вихідного коду призначеної реалізації покращення, комутатор якого має стан очікування або увімкнено. Якщо до плагіна вихідного коду не призначено жодного перемикача, він обробляється так, ніби комутатор має стан увімкненого.

Синтаксис: ENHANCEMENT-SECTION enh\_id SPOTS spot1 spot2 ...

[STATIC]

[INCLUDE BOUND].

...

END-ENHANCEMENT-SECTION.

**55. Enhancement implementation**

**(https://help.sap.com/doc/saphelp\_nw75/7.5.5/en-US/83/43e040e136742ae10000000a155106/content.htm?no\_cache=true)**

Елемент реалізації покращення містить фактичне покращення - наприклад, компонент plug-in вихідного коду містить вихідний код, який потрібно додати. Однак елемент реалізації покращення не містить інформації про те, на яких позиціях об'єкта сховища було створено параметри покращення.

Ви реалізуєте варіант покращення, присвоївши йому елемент реалізації покращення. Призначення реалізації покращення до точки покращення або розділу визначає елемент реалізації покращення. Якщо реалізувати баді, спочатку реалізація (контейнер) повинна бути призначена до місця, яке містить Баді, тоді реалізація Баді може бути створена в контейнері, який призначений до місця, що містить визначення Баді.

*Enhancement implementation*

Прості реалізації покращення (Enhancement implementation)— об'єкти репозиторію, які слугують контейнерами для елементів реалізації покращення. Проста реалізація покращення може містити один або кілька елементів реалізації посилення, які присвоюються визначенню елементів посилення місця посилення.

Реалізації композитного вдосконалення використовуються для семантичного групування простих реалізацій посилення. Реалізація композитного вдосконалення містить або одну або кілька простих реалізацій покращення та/або одну або декілька реалізацій композитного покращення відповідного типу.

**56. Enhancements of class methods (pre-exit, post-exit, overwrite-exit)**

<https://wiki.scn.sap.com/wiki/display/ABAP/Enhancement+Framework+-+Class+Enhancements+-+Pre-exit,+Post-exit+and+Overwrite-exit+methods+-+Concept+and+Simple+Scenarios>

<https://www.stechies.com/enhance-standard-class/>

Існує три типи виходу:

**Pre-Method Exit**: Цей вихід виконує користувацьку логіку перед виконанням стандартного методу.

**Post-Method Exit:** Цей вихід виконує користувацьку логіку після виконання стандартного методу.

**Overwrite-Method Exit:** Цей вихід повністю замінює стандартну логіку.

**Pre-exit methods -** це розширені методи, які можуть бути викликані під час виконання, перш ніж початковий код методу повинен бути виконаний або спрацьований.Попередній вихід може бути використаний для певних перевірок або передачі деяких даних від попереднього виходу до оригінального методу.

**Post-exit methods -** це розширені методи, які можуть бути викликані під час виконання після того, як оригінальний код методу був виконаний або спрацьований. **Post-exit** може бути використаний для додавання або поліпшення даних, що надходять з оригінального методу і задовольнити користувальницькі функціональні можливості бізнесу.

**Overwrite-exit methods -** це розширені методи, які виконують роль заміни під час запуску оригінального методу.

**57. BADI implementation**

BADI – це техніка вдосконалення, яка дозволяє програмісту SAP, користувачеві або певній галузі додати додатковий код до існуючої програми в системі SAP. Ми можемо використовувати стандартну або налаштовану логіку для покращення системи SAP. BADI необхідно спочатку визначити, а потім запровадити для покращення програми SAP.

<https://www.tutorialspoint.com/sap_abap/sap_abap_business_add_ins.htm#:~:text=A%20BADI%20is%20an%20enhancement,implemented%20to%20enhance%20SAP%20application>.

**58. BADI implementation**

(https://help-sap-com.translate.goog/doc/saphelp\_nw75/7.5.5/en-US/44/f518d884056c30e10000000a114a6b/content.htm?no\_cache=true&\_x\_tr\_sl=en&\_x\_tr\_tl=uk&\_x\_tr\_hl=ru&\_x\_tr\_pto=wapp)

У попередньому прикладі ( [Building Your First BAdI](https://help-sap-com.translate.goog/doc/saphelp_nw75/7.5.5/en-US/44/f5175e19fd2463e10000000a1553f7/content.htm?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp) ) ви визначили BAdI, надали резервний клас, створили екземпляр BAdI та викликали метод BAdI. Ви також надали місце для покращення, яке служитиме контейнером для BAdI.

Як ви вже знаєте, резервний клас вибирається у випадку, якщо реалізація BAdI недоступна. Оскільки ви ще не створили реалізацію BAdI, у вашому прикладі коду використовується реалізація методу резервного класу.

У цьому розділі ви дізнаєтеся, як створити реалізацію BAdI. Як тільки є відповідна реалізація BAdI, методи резервного класу більше не використовуються.

Наразі у вас є такі записи:

* Місце підвищенняz\_es\_calc\_tax, в якому BAdIz\_badi\_calc\_vatживе.
* Інтерфейсz\_if\_calc\_vatодним методомget\_vat().

Інтерфейс BAdI визначає важливу частину ідентифікації BAdI. Він визначає методи BAdI, які можна використовувати.

Створіть контейнер для реалізації

Відповідний тип контейнера для реалізацій BAdI називається реалізацією покращення. Проста реалізація покращення може зберігати багато різних реалізацій BAdI, але з одним обмеженням: проста реалізація покращення однозначно призначається місці покращення. Тобто (проста) реалізація покращення може зберігати лише реалізації BAdI BAdI, які належать до місця, якому призначено реалізацію простого покращення. Тому (проста) реалізація покращення не може зберегти реалізації BAdI, які реалізують BAdI, що належать до різних місць.

Щоб створити контейнер, унікально призначений для місця покращення, до якого належить ваш BAdI, виконайте такі дії:

1. У Навігаторі об’єктів (транзакція SE80) відкрийте місце покращення, яке ви вже створили, і натисніть кнопку Створити реалізацію покращення.
2. У новому діалоговому вікні, що з’явиться, створіть реалізацію покращення:

Відкриється наступне нове вікно:

Що вас тут просять ввести?

Поки що ви створили контейнер для реалізацій BAdI - вдосконалення. Цей контейнер унікально призначений для вашого місця покращення. Після встановлення цього з’єднання ви можете створити реалізацію BAdI для BAdI у місці покращення. Оскільки ви визначили лише один BAdI у місці покращення, у вас немає вибору. Якщо у вас було більше BAdI у вашому місці покращення, це буде точка, де ви виберете, який BAdI ви хочете реалізувати.

1. Введітьz\_bdi\_calc\_vat\_usяк назву реалізації BAdI, підтвердьте та збережіть місце покращення на наступному малюнку, який показує (просту) реалізацію покращення, яка містить реалізацію BAdIz\_bdi\_calc\_vat\_us:

Зовнішній вигляд (простої) реалізації покращення в інструменті дуже схожий на реалізацію місця покращення. Під Enh. На вкладці «Елементи реалізації» є дерево з реалізацією(ями) BAdI, що міститься в правій частині. Ліворуч ви бачите властивості позначеної реалізації BAdI.

Примітка

Виберіть властивість «Реалізація активна» на панелі «Поведінка під час виконання» на малюнку вище. Якщо ви це зробите, текст нижче зміниться на Реалізація. Це має на меті допомогти вам зрозуміти, для чого призначений вибір, який ви щойно зробили. Далі нижче наведено список, який показує властивості визначення BAdI, якому призначено вашу реалізація BAdI.

**59. Call of BADI.**

CALL BADI викликає метод BAdI. Має статичний і динамічний варіанти. В обох варіантах для badi необхідно вказати опорну змінну BAdI.

У статичному варіанті для badi потрібно вказати опорну змінну BAdI статичного типу конкретного класу BAdI, а для meth – метод BAdI відповідного BAdI повинен бути введений безпосередньо.

У динамічному варіанті для badi необхідно вказати опорну змінну BAdI статичного типу абстрактного суперкласу CL\_BADI\_BASE. Для meth\_name необхідно вказати символьний об’єкт даних, який повинен містити ім’я методу BAdI у верхньому регістрі під час виконання оператора.

Що стосується адресації методів BAdI, опорна змінна BAdI поводиться як опорна змінна інтерфейсу зі статичним типом інтерфейсу BAdI, на який впливає. Метод BAdI, оголошений як компонент асоційованого інтерфейсу BAdI, адресується безпосередньо за допомогою його імені. Методи BAdI, оголошені в інтерфейсах компонентів інтерфейсу BAdI, можуть бути адресовані за допомогою селектора компонентів інтерфейсу або будь-яких імен псевдонімів.

Доповнення parameter\_list або parameter\_tables призначають фактичний параметр формальним параметрам методу BAdI або обробляють винятки, не засновані на класах, точно так, як описано у Виклику методу. Винятки, які можуть виникати під час динамічного доступу, також такі ж, як і для CALL METHOD.

Оператор CALL BADI вказує, що вказаний метод викликається у всіх додатках об’єкта, на які посилається об’єкт BAdI, на який посилається badi.

**Синтаксис**:

CALL BADI { badi->meth parameter\_list }

| { badi->(meth\_name) {parameter\_list

| parameter\_tables} }.

(<https://help.sap.com/doc/abapdocu_752_index_htm/7.52/en-us/abapcall_badi.htm>)

**60. BADI fallback class**

BADI fallback-класс

Словникове значення ‘fallback’ — це «альтернативний план, який можна використати в надзвичайній ситуації». Саме для цього і призначений резервний клас. Резервний клас з’являється, коли для BADI не існує активної реалізації.

1. **ALV main functions**

(Документація SAP: <https://help.sap.com/saphelp_nw73/HELPDATA/EN/4e/c38f8788d22b90e10000000a42189d/content.htm?no_cache=true> )

SAP List Viewer (ALV) є інтегрованим елементом **середовища програмування ABAP Objects** . Це дозволяє розробникам додатків швидко реалізувати відображення структурованих наборів даних, надаючи три різні інструменти ALV, по одному для відображення:

* Простих та двовимірних таблиць
* Ієрархічно-послідовних списків
* Деревних конструкцій

SAP List Viewer надає розробникам додатків:

* Уніфікований об’єктно-орієнтований API для всіх інструментів ALV, де це можливо
* Узгоджений API
* Якнайшвидше виявлення помилок під час програмування (наприклад, винятки вказують, коли методи неможливі в конкретних ситуаціях)
* Функції доступності інтегровані в ALV, а це означає, що вам не потрібно надавати ці функції самостійно за допомогою програми

За допомогою ALV ви можете використовувати майже однакові методи програмування незалежно від інструменту ALV для відображення різних списків, таблиць або деревоподібних структур. Методи, параметри або класи змінюються лише там, де специфічні для інструментів функції вимагають спеціальної процедури.

«Елемент керування ALV Grid — це гнучкий інструмент для відображення списків. Інструмент забезпечує загальні операції зі списком як загальні функції і може бути розширений самостійно визначеними параметрами».

Елемент керування ALV Grid використовується для створення неієрархічних, інтерактивних списків і списків сучасного дизайну.

Елемент керування ALV Grid забезпечує типові функції списку, як-от сортування, фільтрацію, підсумовування тощо, а також надає можливість розробити функції користувача там, де це необхідно. Він представляє численні інтерфейси, такі як Excel Inplace і Crystal Reports.

**62. Smartforms and related ABAP statements (**[**https://www.guru99.com/smart-forms.html**](https://www.guru99.com/smart-forms.html)**)**

SAP Smart Forms використовується для створення та підтримки форм для масового друку в SAP Systems. Як вихідний засіб SAP Smart Forms підтримує принтер, факс, електронну пошту або Інтернет (за допомогою згенерованого виводу XML).

Переваги Smartforms

* Вони допомагають адаптувати форми без будь-яких знань з програмування завдяки повністю графічному інтерфейсу користувача
* При активації смарт-форми система автоматично генерує функціональний модуль і час виконання.
* Щоб внести будь-які зміни, ми повинні використовувати Drag & Drop, Cut & Paste. Ці дії не включають написання рядків кодування або використання мови сценаріїв.
* Можна вставляти статичні та динамічні таблиці. До них належать переходи рядків в окремих клітинках таблиці, ініціювання подій для заголовків таблиці та проміжних підсумків, а також сортування даних перед виведенням.
* Розумні форми дозволяють користувачеві включати графіку, яка може відображатися як частина форми або як фонова графіка. Під час роздруківки користувач може придушити фонову графіку в міру необхідності.
* Можлива Веб-публікація за допомогою згенерованого виводу XML

**63. PDF forms and related ABAP statements**

Розробники створюють або змінюють форми на основі PDF за допомогою Form Builder (SFP транзакцій), інтегрованого з ABAP Workbench. Існує два типи форм на основі PDF:

* Інтерактивні форми на основі PDF (тип об’єкта PDFA)
* Друковані форми на основі PDF (тип об'єкта PDFB)

Ви викликаєте форми на основі PDF безпосередньо для перекладу в транзакції SE63, де вони перекладаються в довгому текстовому редакторі.

Стейти:

* Декларативні заяви

Ці оператори визначають типи даних або оголошують об’єкти даних, які використовуються іншими операторами в програмі чи підпрограмі. Зібрані декларативні оператори в програмі або підпрограмі складають її частину оголошення.

Приклади декларативних ключових слів: TYPES, DATA, TABLES

* Заяви про модуляцію

Ці оператори визначають блоки обробки в програмі ABAP.

Ключові слова модульності можна далі розділити на:

Визначення ключових слів

Ви використовуєте оператори, що містять ці ключові слова, щоб визначити підпрограми, функціональні модулі, діалогові модулі та методи. Ви завершуєте ці блоки обробки за допомогою операторів END .

Приклади остаточних ключових слів: METHOD ... ENDMETHOD, FUNCTION ... ENDFUNCTION, MODULE ... ENDMODULE.

Ключові слова подій

Ви використовуєте оператори, що містять ці ключові слова, щоб визначити блоки подій . Немає спеціальних операторів для завершення блоків обробки - вони закінчуються, коли вводиться наступний блок обробки.

Приклади ключових слів події: AT SELECTION SCREEN, START-OF-SELECTION, AT USER-COMMAND

* Контрольні заяви

Ви використовуєте ці оператори, щоб керувати ходом програми ABAP всередині блоку обробки відповідно до певних умов.

Приклади контрольних ключових слів: IF, WHILE, CASE

* Call Statements

Ви використовуєте ці оператори для виклику блоків обробки, які ви вже визначили за допомогою операторів модульності. Блоки, які ви викликаєте, можуть бути в одній програмі ABAP або в іншій програмі.

Приклади ключових слів для виклику: CALL METHOD, CALL TRANSACTION, SUBMIT, LEAVE TO

* Оперативні заяви

Ці ключові слова обробляють дані, які ви визначили за допомогою декларативних операторів.

Приклади операційних ключових слів: MOVE, ADD

**64. Smartforms and PDF forms transactions (workbook 7)**

У транзакції **SMARTFORMS** можна створювати смартформи та стилі для них.

Для створення стилю потрібно обрати опцію Style, ввести назву та натиснути кнопку Create. Далі потрібно створити вузли (Create Node) для основних елементів форми, наприклад Header, Table Cells. Далі можна налаштовувати вигляд цих елементів, встановлювати шрифти та розмір.

Для створення смартформи у транзакції SMARTFORMS обираємо опцію Form, вводимо її назву та клікаємо Create.

FORM ATTRIBUTES: General Attributes і Output options.

FORM INTERFACE:

a. Import — параметри, які необхідно імпортувати до функціонального модуля форми.

b. Export — параметри, які експортуються з функціонального модуля форми.

c. Tables — параметри, які використовуються для передачі внутрішніх таблиць у головну програму.

d. Exceptions — винятки, які створюються у функціональному модулі.

Усі параметри, визначені в інтерфейсі форми, відображатимуться як параметри у формі функціональний модуль.

У GLOBAL DEFINITIONS визначаються глобальні змінні, типи, символи полів тощо.

Транзакція **SFP** створює PDF-форми та інтерфейси.

**65. Web Dynpro**

**(**<https://www.tutorialspoint.com/sap_abap/sap_abap_web_dynpro.htm>**)**

Web DynPro (WD) для ABAP — технологія стандартного інтерфейсу користувача SAP, розроблена SAP AG. Він може бути використаний при розробці веб-застосунків в середовищі SAP ABAP, що використовує засоби та концепції розробки SAP. Він забезпечує інтерфейс веб-користувача на передньому кінці для підключення безпосередньо до серверних систем SAP R/3 для доступу до даних і функцій для звітності.

Web DynPro для ABAP складається з середовища виконання і графічного середовища розробки з конкретними інструментами розробки, інтегрованими в ABAP Workbench (транзакція: SE80).

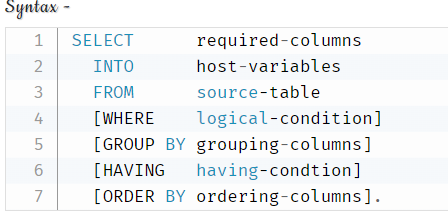
**66. SELECT ...**

<https://www.tutorialscampus.com/sap-abap/open-sql-select.htm>

<https://help.sap.com/doc/abapdocu_750_index_htm/7.50/en-US/abapselect.htm>

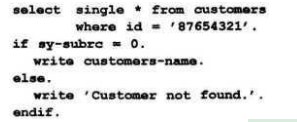
**SELECT -** це інструкція Open SQL для читання даних з однієї або декількох таблиць баз даних, класичних представлень або CDS об'єктів в об'єкти даних.

Оператор SELECT використовується для читання даних з таблиць баз даних. Приходячи до програмування, завжди бажано отримати лише один запис для більш плавної обробки. Якщо оператор SELECT отримує більше одного рядка, перший рядок з набору результатів буде отримано до робочої області. Якщо всі рядки, отримані за допомогою оператора SELECT, потрібні для обробки в програмі ABAP, то бажано використовувати оператор CURSOR.



|  |  |
| --- | --- |
| required-columns | Вказує назви стовпчиків таблиці, з яких дані були вилучені в host-variables. |
| host-variables | Визначає host-variables, в яких дані стовпчика витягуються з required-columns |
| source-table | Вказує database table або view name |
| logical-condition | Визначає умову для критеріїв вибору |
| grouping-columns | Визначає групу стовпчиків для створення одного рядка з декількох рядків з одним значенням |
| having-condtion | Вказує логічну умову для умови GROUP BY |
| ordering-columns | Тут ви можете вказати порядок розташування рядків. Порядок може бути висхідним (ASC) або спадним (DESC). |

**67. SELECT SINGLE ...**



При читанні єдиного елемента таблиці завжди застосовується команда select single і запро- ня, де вказані всі поля первинного ключа. Після її виконання системний код повернення sy- subrc вказує на успішне (значення дорівнює нулю) або неуспішне (значення не дорівнює нулю) завершення запиту до бази даних:

Зверніть увагу, що оператор select single виконується правильно лише в тому випадку, якщо все поля первинного ключа задані як поодинокі значення (тобто. з допомогою операторів = чи eq).

**68. SELECT ... FOR ALL ENTRIES**

(https://help-sap-com.translate.goog/doc/abapdocu\_752\_index\_htm/7.52/en-US/abenwhere\_logexp\_itab.htm?\_x\_tr\_sl=en&\_x\_tr\_tl=uk&\_x\_tr\_hl=ru&\_x\_tr\_pto=wapp)

**ВИБІР - ДЛЯ ВСІХ ЗАПИСІВ**

**Синтаксис**

... ДЛЯ ВСІХ ЗАПИСІВ В @itab, ДЕ ... [**оператор**](javascript:call_link('abenwhere_logexp_compare.htm')?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp) col @itab-comp ...

**Ефект**

Якщо доповнення FOR ALL ENTRIES вказано перед мовним елементом [WHERE](javascript:call_link('abapwhere.htm')?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp) оператора SELECT головного [запиту](javascript:call_link('abenmainquery_glosry.htm')?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp) , компоненти comp внутрішньої таблиці itab , зазначені тут, можна використовувати в [sql\_cond](javascript:call_link('abenwhere_logexp.htm')?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp) з правого боку порівняння [оператора відношення](javascript:call_link('abenwhere_logexp_compare.htm')?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp) в порівняння зі [стовпчиком](javascript:call_link('abenopen_sql_columns.htm')?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp)col . Зазначений компонент comp має бути сумісним із стовпцем col . Внутрішня таблиця itab може мати структурований або елементарний тип рядка. Для елементарного типу рядка [псевдокомпонент](javascript:call_link('abenpseudo_component_glosry.htm')?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp) table\_line необхідно вказати для comp . Ім'я хост-змінної [dbcur](javascript:call_link('abenopen_sql_host_variables.htm')?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp) має мати префікс escape-символу @ .

Весь логічний вираз [sql\_cond](javascript:call_link('abenwhere_logexp.htm')?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp) оцінюється для кожного окремого рядка внутрішньої таблиці itab . Набір результатів оператора SELECT є об’єднанням наборів результатів, створених окремими оцінками. Рядки, які зустрічаються більше одного разу, автоматично видаляються з набору результатів. Тут розглядається повний вміст рядка.

Якщо внутрішня таблиця itab порожня, вся умова WHERE ігнорується. Це означає, що жоден з рядків у таблиці бази даних не пропускається і поміщається в набір результатів (після видалення повторюваних рядків).

Логічний вираз [sql\_cond](javascript:call_link('abenwhere_logexp.htm')?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp) умови WHERE може містити кілька логічних виразів за допомогою І та АБО . Однак, якщо вказано FOR ALL ENTRIES , має бути принаймні одне [порівняння](javascript:call_link('abenwhere_logexp_compare.htm')?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp) зі стовпцем внутрішньої таблиці itab , яке можна вказати статично або динамічно.

При використанні доповнення ДЛЯ ВСІХ ЗАПИСІВ з іншими доповненнями застосовуються такі обмеження:

* Додавання FOR ALL ENTRIES можливе лише перед умовами WHERE в окремому [операторі SELECT](javascript:call_link('abapselect.htm')?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp) або в [основному запиті](javascript:call_link('abenmainquery_glosry.htm')?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp) після [OPEN CURSOR , якщо за допомогою](javascript:call_link('abapopen_cursor.htm')?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp)[WITH](javascript:call_link('abapwith.htm')?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp) не визначено жодних загальних табличних виразів .
* Додаток ДЛЯ ВСІХ ЗАПИСІВ не можна використовувати з додатком [SINGLE](javascript:call_link('abapselect_single.htm')?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp) .
* Додаток FOR ALL ENTRIES не можна поєднувати з [виразами SQL](javascript:call_link('abapsql_expr.htm')?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp) .
* Якщо використовується доповнення FOR ALL ENTRIES , не можна створювати [дескриптори LOB як](javascript:call_link('abenselect_into_lob_handles.htm')?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp)[потоки зчитувачів](javascript:call_link('abenreader_stream_glosry.htm')?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp) або як [локатори](javascript:call_link('abenlocator_glosry.htm')?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp) в [цільовій області](javascript:call_link('abapinto_clause.htm')?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp) окремого [оператора SELECT](javascript:call_link('abapselect.htm')?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp) .
* Додаток ДЛЯ ВСІХ ЗАПИСІВ не можна поєднувати з [UNION](javascript:call_link('abapunion.htm')?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp) .
* Додавання ДЛЯ ВСІХ ЗАПИСІВ не слід використовувати з додаванням [GROUP BY](javascript:call_link('abapgroupby_clause.htm')?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp) . Додавання GROUP BY не діє, якщо використовується ДЛЯ ВСІХ ЗАПИСІВ .
* У операторі SELECT з FOR ALL ENTRIES додавання [ORDER BY](javascript:call_link('abaporderby_clause.htm')?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp) можна використовувати лише з додаванням PRIMARY KEY і лише для доступу до однієї таблиці або представлення. У цьому випадку всі стовпці первинного ключа (крім стовпця клієнта в таблицях, що стосуються клієнта) мають бути в [списку SELECT](javascript:call_link('abapselect_list.htm')?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp) .
* Жодні [вирази шляху](javascript:call_link('abenopen_sql_path.htm')?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp) не можна використовувати в операторі SELECT з FOR ALL ENTRIES .
* Якщо використовується додавання ДЛЯ ВСІХ ЗАПИСІВ , поля бази даних вбудованих типів STRING і RAWSTRING плюс LCHR і LRAW не повинні зустрічатися в [списку SELECT](javascript:call_link('abapselect_list.htm')?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp) . Ці типи даних запобігають видаленню рядків, які зустрічаються в системі бази даних більше одного разу. Ці рядки видаляються лише з набору результатів на сервері додатків. Якщо вказано в списку SELECT , виникає попередження про перевірку синтаксису, яке може бути приховане прагмою.

Внутрішня таблиця itab оцінюється один раз для кожного запиту. Будь-які зміни, внесені до вмісту внутрішньої таблиці в циклі SELECT або WITH , ігноруються логічним виразом.

**Примітки**

* Одну і ту ж внутрішню таблицю можна вказати після FOR ALL ENTRIES і після [INTO](javascript:call_link('abapinto_clause.htm')?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp) . Вміст таблиці оцінюється за допомогою FOR ALL ENTRIES , а потім перезаписується [реченням INTO](javascript:call_link('abapinto_clause.htm')?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp) .
* Порівняння зі стовпцем внутрішньої таблиці можна також виконати за допомогою умови WHERE [підзапиту](javascript:call_link('abensubquery_glosry.htm')?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp) для того самого [джерела даних](javascript:call_link('abapselect_data_source.htm')?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp) .
* Стосовно рядків, що зустрічаються більше одного разу в наборі результатів, додавання ДЛЯ ВСІХ ЗАПИСІВ має той самий ефект, що й у випадку, коли додавання [DISTINCT](javascript:call_link('abapselect_clause.htm')?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp) вказано у визначенні набору вибору. На відміну від DISTINCT , рядки не завжди видаляються з системи бази даних, а іноді спочатку видаляються з набору результатів на сервері додатків. Повторювані рядки потім видаляються з системи бази даних, якщо оператор SELECT можна передати в систему бази даних як один оператор SQL. Тут підтримується додавання DISTINCT . Якщо SELECTІнструкцію необхідно розповсюдити на кілька операторів SQL, перш ніж вона буде передана, або якщо стовпці типів STRING і RAWSTRING плюс LCHR і LRAW вказані в списку [SELECT ,](javascript:call_link('abapselect_list.htm')?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp) рядки агрегуються на сервері програм.
* Якщо повторювані рядки спочатку видаляються із сервера додатків, усі рядки, визначені умовою WHERE (у деяких випадках), передаються до внутрішньої системної таблиці, а потім агрегуються. Максимальний [розмір](javascript:call_link('abenmemory_consumption_2.htm')?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp) цієї системної таблиці обмежено розміром звичайних внутрішніх таблиць. Точніше, системна таблиця завжди потрібна, якщо одночасно використовується одне з доповнень [PACKAGE SIZE](javascript:call_link('abapinto_clause.htm')?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp) або [UP TO , OFFSET](javascript:call_link('abapselect_up_to_offset.htm')?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp) . Потім вони не впливають на кількість рядків, що передаються від сервера бази даних до сервера додатків, але використовуються лише тоді, коли рядки передаються із системної таблиці до фактичної цільової області. Якщо максимальний розмір внутрішньої системної таблиці перевищено, виникає помилка під час виконання.
* Додаток ДЛЯ ВСІХ ЗАПИСІВ обходить [буферизацію](javascript:call_link('abensap_buffering_glosry.htm')?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp) таблиць для таблиць із загальною буферизацією, якщо умова після ДЛЯ ВСІХ ЗАПИСІВ не дозволяє точно вказати єдину загальну область.

У всіх інших випадках використовується буферизація таблиці, а додавання ДЛЯ ВСІХ ЗАПИСІВ може бути ефективнішою альтернативою [виразам об’єднання](javascript:call_link('abenjoin_expression_glosry.htm')?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp) .

* Перед використанням внутрішньої таблиці itab після FOR ALL ENTRIES завжди перевіряйте, чи внутрішня таблиця не є початковою. У початкових внутрішніх таблицях усі рядки зчитуються з бази даних незалежно від будь-яких подальших умов, зазначених після WHERE . Зазвичай це не обов’язкова поведінка.
* Якщо повна умова WHERE ігнорується, оскільки внутрішня таблиця itab порожня, неявна умова WHERE для поточного клієнта або клієнта, зазначеного за допомогою використання [UING CLIENT](javascript:call_link('abapselect_client.htm')?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp) , не впливає (якщо ввімкнено автоматичну [обробку клієнта ).](javascript:call_link('abenopen_sql_client_handling.htm')?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp)Це означає, що всі дані зчитуються лише з поточного клієнта. Якщо автоматична обробка клієнта вимкнена за допомогою [CLIENT SPECIFIED](javascript:call_link('abapselect_client.htm')?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp) , для клієнта не існує неявної умови WHERE . Будь-яка умова WHERE , явно зазначена для стовпця клієнта, ігнорується з умовою повного, якщо внутрішня таблиця itab порожня, а дані з усіх клієнтів зчитуються.
* Якщо в строгому режимі [перевірки синтаксису](javascript:call_link('abenopensql_strict_modes.htm')?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp)FOR ALL ENTRIES вказано разом зі стовпцями типів STRING і RAWSTRING плюс LCHR і LRAW у списку [SELECT ,](javascript:call_link('abapselect_list.htm')?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp) перевірка синтаксису виконується в [строгому режимі з Випуску 7.52](javascript:call_link('abenopensql_strict_mode_752.htm')?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru&_x_tr_pto=wapp) .
* **Example**
* Gets all flight data for a specified departure city. The relevant airlines and flight numbers are first passed to an internal table entry\_tab, which is evaluated in the WHERE condition of the subsequent SELECT statement. This selection could also be carried out in a single SELECT statement by using a join in the [FROM clause](javascript:call_link('abapfrom_clause.htm')). Make sure that the table entry\_tab is not initial before the SELECT statement is executed using FOR ALL ENTRIES.
* DATA city TYPE spfli-cityfrom VALUE 'FRANKFURT'.cl\_demo\_input=>request( CHANGING field = city ).SELECT carrid, connid       FROM spfli       WHERE cityfrom = @( to\_upper( city ) )       INTO TABLE @DATA(entry\_tab).IF entry\_tab IS NOT INITIAL.  SELECT carrid, connid, fldate         FROM sflight         FOR ALL ENTRIES IN @entry\_tab         WHERE carrid = @entry\_tab-carrid AND               connid = @entry\_tab-connid         ORDER BY PRIMARY KEY         INTO TABLE @DATA(result\_tab).  cl\_demo\_output=>display( result\_tab ).ENDIF.
* **Example**
* Uses FOR ALL ENTRIES with an empty internal table. All rows of the database table are respected. The number of read rows is usually, however, smaller in the first SELECT statement than in the second statement. This is because only one column is read and hence more duplicate rows can be removed. The second SELECT statement, on the other hand, moves all rows of the database table to the results set, since their structure covers the full table key.
* DATA carriers TYPE TABLE OF scarr.SELECT carrid, connid       FROM spfli       FOR ALL ENTRIES IN @carriers       WHERE carrid = @carriers-carrid       INTO TABLE @DATA(result1).cl\_demo\_output=>write( result1 ).SELECT carrid       FROM spfli       FOR ALL ENTRIES IN @carriers       WHERE carrid = @carriers-carrid       INTO TABLE @DATA(result2).cl\_demo\_output=>display( result2 ).

**69. SELECT ... INNER JOIN.**

Ключове слово INNER JOIN вибирає записи, які мають відповідні значення в обох таблицях.

**Синтаксис**:

SELECT column\_name(s)  
FROM table1  
INNER JOIN table2ON table1.column\_name = table2.column\_name;

(<https://www.w3schools.com/sql/sql_join_inner.asp>)

**70. COMMIT WORK and ROLLBACK WORK**

COMMIT WORK та ROLLBACK WORK

COMMIT WORK залишає в силі всі зміни, внесені поточною транзакцією, тоді як робота ROLLBACK відновлює їх