

**МІНІСТЕРСТВО ОСВІТИ НАУКИ УКРАЇНИ
НАЦІОНАЛЬНА МЕТАЛУРГІЙНА АКАДЕМІЯ УКРАЇНИ**



**РОБОЧА ПРОГРАМА,
методичні вказівки та індивідуальні завдання
до вивчення дисципліни «Проектування організаційних і технологічних
інформаційних управляючих систем»**

для студентів спеціальності 122 «Комп'ютерні науки»

Дніпро НМетАУ 2019

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНА МЕТАЛУРГІЙНА АКАДЕМІЯ УКРАЇНИ**

**РОБОЧА ПРОГРАМА,
методичні вказівки та індивідуальні завдання
до вивчення дисципліни «Проектування організаційних і
технологічних інформаційних управляючих систем»
для студентів спеціальності 122 «Комп'ютерні науки»
заочної форми навчання**

Дніпро НМетАУ 2019

УДК 621.65

Робоча програма, методичні вказівки та індивідуальні завдання до вивчення дисципліни «Проектування організаційних і технологічних інформаційних управляючих систем» для студентів спеціальності 122 «Комп'ютерні науки» заочної форми навчання / Укл. Н.Л. Дорош, Ю.В. Бабенко. – Дніпропетровськ: НМетАУ, 2019. – 37 с.

Наведена робоча програма дисципліни «Проектування організаційних і технологічних інформаційних управляючих систем» з коротким викладом змісту її розділів і методичними вказівками до вивчення навчального матеріалу, а також завдання до контрольної роботи, на основі якої студент засвоює принципи проектування інформаційних систем, методологію структурного аналізу, об'єктну методологію, мову UML, CASE – додатків Ramus Educational, Rational Rose.

Призначена для студентів спеціальності 122 «Комп'ютерні науки» заочної форми навчання.

Друкується за авторською редакцією.

Укладачі: Н. Л. Дорош, канд. техн. наук, доцент

Ю. В. Бабенко, канд. техн. наук, доцент

Відповідальний за випуск О.І. Михальов, д-р техн. наук, професор

Рецензент В.І. Корсун, д-р техн. наук, професор (НГУ)

Підписано до друку 06.03.2019. Формат 60x84 1/16. Папір типогр. Друк різнограф. Обл.-вид. арк. 2,17. Умов. друк. арк. 2,15. Тираж 100 прим. Замовл. № 7/19.

Національна металургійна академія України.

49600, Дніпро, пр. Гагаріна, 4

ЗМІСТ

ВСТУП.....	4
1 ЗАГАЛЬНІ ВІДОМОСТІ	5
2 РОБОЧА ПРОГРАМА ТА МЕТОДИЧНІ ВКАЗІВКИ	6
2.1 Огляд методологій проектування складних систем	6
2.2 Методологія структурного аналізу.....	7
Використання методології IDEF0. Приклад.	11
Контрольні запитання для самоперевірки.....	19
2.3 Розробка структурної моделі з використанням системи «Ramus Educational».....	20
Створення контекстної діаграми.....	20
Створення діаграми декомпозицій.....	21
Створення діаграми потоків даних (DFD).....	22
2.4 Основи мови моделювання UML	22
Канонічні діаграми мови UML.....	23
Діаграма варіантів використання (use case diagram).....	25
Діаграма класів (class diagram).....	27
Діаграма діяльностей (activity diagram).....	28
Діаграма станів об'єкта(state chart diagram).....	29
Діаграма послідовності (sequence diagrams).....	31
Контрольні запитання для самоперевірки.....	32
3 КОНТРОЛЬНА РОБОТА.....	33
3.1 Завдання	33
3.2 Варіанти контрольних робіт.....	33
ВИСНОВКИ	36
РЕКОМЕНДОВАНА ЛІТЕРАТУРА	37

ВСТУП

Навчальна дисципліна «Проектування організаційних і технологічних інформаційних управляючих систем» присвячена питанням проектування і розробки інформаційних систем.

Мета вивчення дисципліни – вивчення сучасних методологій, методів, інструментальних засобів проектування автоматизованих систем та методики синтезу інформаційних систем (ІС) із використанням системних принципів проектування складних систем.

Сьогодні відповідно до чинного державного освітнього стандарту «Проектування організаційних і технологічних інформаційних управляючих систем» вивчається як самостійна дисципліна студентами спеціальності 122 «Комп'ютерні науки».

Дана дисципліна складається з таких основних тем:

- огляд методологій проектування інформаційних систем;
- методологія структурного аналізу для проектування інформаційних систем;
- методологія об'єктно-орієнтованого проектування. Основи мови моделювання UML;
- використання CASE-засобів в задачах проектування інформаційних систем.

Основні вимоги до виконання контрольної роботи є:

1. до моменту приїзду студента на екзаменаційну сесію він повинен мати конспект дисципліни,
2. виконану і захищену контрольну роботу.

Якщо контрольну роботу не захищено, то студент не допускається до іспиту.

1 ЗАГАЛЬНІ ВІДОМОСТІ

Навчальна дисципліна «Проектування організаційних і технологічних інформаційних управляючих систем» є обов'язковою і входить до циклу професійно-орієнтованих дисциплін за фахом.

Мета вивчення дисципліни – засвоєння студентами теоретичних основ, придбання практичних навичок й освоєння інструментальних засобів проектування інформаційних систем (ІС).

У результаті вивчення дисципліни студент повинен

– знати: сучасні принципи побудови інформаційних, управляючих систем (ІУС); методи оцінки якості проектних рішень (з точки зору соціальних, технічних та економічних критеріїв); задачі структурного та об'єктного підходів проектування інформаційних систем; інструментальні засоби проектування компонентів ІС;

– вміти: використовувати методи збору, обробки та запису інформації в ІС; визначати вимоги до якості і параметрів функціональних підсистем, що входять в ІС; розробляти документацію на технічне завдання, постановку задач проектування системи або її підсистем; використовувати CASE– засоби проектування.

Для успішного складання іспиту з дисципліни необхідно: опрацювати питання контрольної роботи, підготуватись до іспиту згідно питань дисципліни.

Зв'язок з іншими дисциплінами: навички та вміння цієї дисципліни базуються на дисциплінах «Технологія створення програмних продуктів», «Системний аналіз», «Організація баз даних та знань», «Моделювання систем».

Набуті знання та вміння забезпечують необхідну підготовку студентів до виконання дипломної роботи.

2 РОБОЧА ПРОГРАМА ТА МЕТОДИЧНІ ВКАЗІВКИ

Розподіл навчальних годин та форми контролю дисципліни «Проектування організаційних і технологічних інформаційних управляючих систем» для студентів заочної форми навчання спеціальності 122 «Комп'ютерні науки» представлена у таблиці 2.1.

Таблиця 2.1– Розподіл навчальних годин та форми контролю

Усього годин за навчальним планом	150
у тому числі:	
Аудиторні заняття	64
з них:	
– лекції	32
– лабораторні заняття	32
– практичні заняття	
– семінари	
Самостійна робота	86
– виконанні контрольної роботи	
– підготовці до іспиту	12
– вивчення окремих розділів програми, які не увійшли до лекційного курсу	27
Підсумковий контроль (іспит, залік)	іспит

2.1 Огляд методологій проектування складних систем

Програма дисципліни передбачає вивчення *методології структурного аналізу і об'єктної методології* проектування інформаційних систем (ІС).

У Стандарті ISO / ІЕС 2382–1 наведено таке визначення: «*Інформаційна система* – система обробки інформації, що працює спільно з організаційними ресурсами, такими як люди, технічні засоби та фінансові ресурси, які забезпечують і розподіляють інформацію».

Структурна та об'єктна методології проектування ІС відповідають принципам системного аналізу та закладають основу блочно–ієрархічного підходу до проектування. Кожен рівень ієрархії визначається в результаті декомпозиції. У той же час декомпозиція проводиться за різними правилами. У методології

Н.Л. Дорош, Ю.В. Бабенко «Проектування організаційних і технологічних інформаційних управляючих систем»

структурного аналізу декомпозиція проводиться згідно з роботами (функціональна декомпозиція), у об'єктній – по об'єктах (об'єктна декомпозиція).

Методологію структурного аналізу доцільно застосовувати для проектування різних процесів (технологічних та бізнес- процесів, інструкцій та ін.), Тобто де розглядаються дії (роботи).

Об'єктна методологія використовується при проектуванні програмного забезпечення, моделюванні пристроїв у вигляді сукупності об'єктів, в процесі взаємодії яких через передачу повідомлень відбувається виконання необхідних функцій.

2.2 Методологія структурного аналізу

Structured Analysis and Design Technique (SADT) (Технологія структурного аналізу і проектування) – одна з найвідоміших методологій аналізу та проектування систем, яка була розроблена в 70–і роки 20 століття. В даний час ця методологія трансформувалась в методологію *IDEF0 (Integrated DEFinition)*.

Опис системи за допомогою IDEFO називається *функціональною моделлю*. Основним елементом у моделі є *діаграма*. Модель може об'єднувати кілька діаграм в одну ієрархію. Чим глибше діаграма знаходиться в ієрархії, тим більше вона деталізована, тобто тим більш докладно відображає дані або активності системи, або блоку.

Процес моделювання включає збір інформації про досліджувану область, документування отриманої інформації і представлення її у вигляді моделі і уточнення моделі за допомогою ітеративного рецензування. Крім того, цей процес підказує цілком певний шлях виконання узгодженої та достовірної структурної декомпозиції, що є ключовим моментом у кваліфікованому аналізі системи. Методологія IDEFO унікальна в своїй здатності забезпечити як графічну мову, так і процес створення несуперечливої і корисної системи описів.

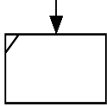
IDEFO є методологією в повному розумінні, тому що вона об'єднує ітеративний процес створення моделі, нотації, мову посилань для діаграм, мову функцій моделей з графічною мовою опису системи, а також рекомендації щодо реалізації аналітичних проектів. Нотації, які керують конфігурацією, гарантують, що нові діаграми будуть коректно вбудовані в ієрархічну структуру моделі. Мова посилань в IDEFO, правила скорочень для посилань, адресованих до окремих частин діаграми, полегшують оформлення зауважень при рецензуванні моделі. Мова функцій дозволяє декларативно визначати правила роботи системи, що часто

Н.Л. Дорош, Ю.В. Бабенко «Проектування організаційних і технологічних інформаційних управляючих систем»

є особливо важливим завершальним кроком в описі системи. Методологія IDEF0 реалізована в програмах *BPWIN*, *RAMUS*.

У таблиці 2.2 наведені основні «будівельні блоки» для діаграм IDEF0.

Таблиця 2.2 - Основні «будівельні блоки» для діаграм IDEF0

№	Назва	Опис елементу IDEF0 діаграми	Графічне представлення
1	2	3	4
1	Модуль поведінки.	Об'єкт служить для опису функцій (процедур, робіт), які виконуються підрозділами / співробітниками підприємства	
2	Стрілка входу (input)	Стрілка описує вхідні документи, інформацію, матеріальні ресурси, необхідні для виконання функції.	
3	Стрілка виходу (output)	Стрілка описує вихідні документи, інформацію, матеріальні ресурси, що є результатом виконання функції.	
4	Стрілка управління (control)	Стрілка описує керуючий вплив, наприклад розпорядження, нормативний документ і т.д. У нотації IDEF0 кожна процедура повинна обов'язково мати не менше однієї стрілки зверху, що відбиває керуючий вплив.	
5	Стрілка механізму (mechanism)	Стрілка знизу описує т.зв. механізми, тобто ресурси, необхідні для виконання процедури, але не змінюють в процесі її виконання свій стан. Приклади: співробітник, верстат.	

Варто відзначити, що у моделі всі роботи і стрілки повинні бути іменовані.

Кількісний аналіз діаграм

Для проведення кількісного аналізу діаграми перерахуємо показники моделі:

- кількість блоків на діаграмі – N;

- рівень декомпозиції діаграми – L;
- збалансованість діаграми – B;
- число стрілок, що з'єднуються з блоком – A.

Даний набір факторів відноситься до кожної діаграми моделі. Далі будуть перераховані рекомендації з бажаним значенням факторів діаграми.

Необхідно прагнути до того, щоб кількість блоків на діаграмах нижніх рівнів була б нижче кількості блоків на батьківських діаграмах, тобто зі збільшенням рівня декомпозиції зменшувався б коефіцієнт $N \setminus L$. Таким чином, спадання цього коефіцієнта говорить про те, що в міру декомпозиції моделі функції повинні спрощуватися, отже, кількість блоків повинна зменшуватися.

Діаграми повинні бути збалансовані. Це означає, що в рамках однієї діаграми не повинно відбуватися ситуації, коли в роботу вхідних стрілок і стрілок управління значно більше, ніж які виходять. Слід зазначити, що дана рекомендація може не виконуватися в моделях, що описують виробничі процеси. Наприклад, при описі процедури складання в блок може входити безліч стрілок, що описують компоненти виробу, а виходити одна стрілка – готовий виріб.

Введемо коефіцієнт збалансованої діаграми:

$$K_b = \left| \frac{\sum_{I=1}^N A_I}{N} - \max_{I=1}^N (A_I) \right|$$

Необхідно прагнути, щоб K_b був мінімальний для діаграми. Крім аналізу графічних елементів діаграми необхідно розглядати найменування блоків. Для оцінки імен складається словник елементарних (тривіальних) функцій модельованої системи. Фактично в даний словник повинні потрапити функції нижнього рівня декомпозиції діаграм. Наприклад, для моделі БД елементарними можуть бути функції "знайти запис", "додати запис в БД", в той час як функція "реєстрація користувача" вимагає подальшого опису.

Після формування словника і складання пакету діаграм системи необхідно розглянути нижній рівень моделі. Якщо на ньому виявляться збігу назвах блоків діаграм і слів зі словника, то це говорить, що достатній рівень декомпозиції досягнутий. Коефіцієнт кількісно відображає даний критерій, можна записати як $L * C$ – добуток рівня моделі на число збігів імен блоків з словами зі словника. Чим нижче рівень моделі (більше L), тим цінніше збіги.

Діаграми потоків даних (Data Flow Diagrams)

Діаграми DFD можна використовувати як доповнення до діаграм IDEF0 для опису документообігу та обробки інформації. Ці діаграми представляють мережу пов'язаних між собою робіт.

DFD описує:

- функції обробки інформації (роботи);
- документи (стрілки, arrow), об'єкти, співробітників або відділи, які беруть участь в обробці інформації;
- зовнішні посилання (external reference), які забезпечують інтерфейс з зовнішніми об'єктами, що знаходяться за межами модельованої системи;
- таблиці для зберігання документів (сховища даних, data store).

Потоки даних є механізмами, що використовуються для моделювання передачі інформації (або фізичних компонентів) з однієї частини системи в іншу. Потоки зображуються на діаграмі іменованими стрілками, орієнтація яких вказує напрямок руху інформації. Стрілки можуть підходити до будь-якої грані прямокутника роботи і можуть бути двонаправленими для опису взаємодії типу "команда-відповідь".

Призначення *процесу* полягає в продукуванні вихідних потоків із вхідних відповідно до дії, що задається ім'ям процесу. Кожен процес повинен мати унікальний номер для посилань на нього всередині діаграми. Цей номер може використовуватися спільно з номером діаграми для отримання унікального індексу процесу у всій моделі.

Сховище даних дозволяє на певних ділянках визначати дані, які будуть зберігатися в пам'яті між процесами. Фактично сховище представляє "зрізи" потоків даних у часі. Інформація, яку воно містить, може використовуватися в будь-який час після її визначення, при цьому дані можуть вибиратися в будь-якому порядку. Ім'я сховища має ідентифікувати його вміст. У випадку, коли потік даних входить в сховище або виходить з нього і його структура відповідає структурі сховища, він повинен мати те ж саме ім'я.

Зовнішня сутність – це сутність поза контекстом системи, що є джерелом або приймачем даних системи. Передбачається, що об'єкти, що представлені такими вузлами, не повинні брати участь ні в якій обробці. Зовнішні сутності зображуються у вигляді прямокутника з тінню і зазвичай розташовуються по краях діаграми. Одна зовнішня сутність може бути використана багаторазово на одній, або декількох діаграмах.

Використання методології IDEF0. Приклад.

Розглянемо приклад розробки моделі гірничозбагачувального комбінату (ГЗК).

Процес моделювання на мові IDEF0 включає збір інформації про область, яку досліджують, документування отриманої інформації і представлення її у вигляді моделі і уточнення моделі за допомогою ітеративного рецензування. Крім того, цей процес підказує цілком певний шлях виконання погодженої та достовірної структурної декомпозиції, що є ключовим моментом у кваліфікованому аналізі системи.

Отже, на першому етапі моделювання необхідно провести збір інформації про область, яку досліджують. Уявімо інформацію про ГЗК в такому вигляді, який дозволить сформулювати суб'єкт, мету і точку зору. Спробуємо записати відповіді на деякі питання.

1. Які роботи (функції) виконуються на ГЗК?
2. Які роботи можуть повторюватися, тобто що підлягає ітерації?
3. Для чого призначена модель?
4. Чий точці зору відповідає модель?

На початковому етапі, намагаючись відповісти на поставлені питання, сформулюємо:

Мета: отримання загального уявлення про систему для того, щоб надалі вдосконалити її;

Точка зору: Керуючий гірничозбагачувальним комбінатом.

Робота ГЗК складається з наступних складових: виробництво сировини; роботи відділу менеджменту; роботи відділу розвитку комбінату; роботи акціонерної компанії; роботи відділу соціального забезпечення; документообігу.

Розглянемо в початковому варіанті надану інформацію більш структурованою.

Виробництво сировини для металургійних заводів України та країн Європи (організація робіт з видобутку залізної руди та її збагачення)

- Видобуток залізної руди.
- Збагачення залізної руди.
- Виготовлення залізородного концентрату.

Робота відділу менеджменту

- Реклама готової продукції.
- Надання послуг у пресі для потенційних клієнтів.

- Консультації з клієнтом по продукції.
- Прийом замовлень на поставку продукції.
- Поставка клієнтам готової продукції.
- Запити відгуків про продукцію у своїх клієнтів.
- Укладання договорів на співпрацю з юридичними особами.

Робота відділу розвитку комбінату

- Закупівля допоміжних засобів виробництва.

Робота акціонерної компанії

- Продаж акцій комбінату.
- Проведення щорічних зборів акціонерів.
- Звіт про діяльність підприємства перед акціонерами.

Робота відділу соціального забезпечення

Підтримка соціально–культурних потреб міста–супутника.

Ведення документації з усіх проблемних питань.

Перерахуємо, які можливі ітерації:

***ітерація 1:** якщо поставка продукції не сталася в зазначені договором строки, то проводяться переговори з клієнтами, службове розслідування і виконання повторної поставки, якщо доведена вина комбінату;*

***ітерація 2:** якщо роботи з видобутку руди не почалися в належний час, то проводиться розслідування, за результатами якого приймаються відповідні заходи, але робота виконується в будь–якому випадку в перенесені терміни;*

***ітерація 3:** якщо відбулася поставка неякісного обладнання, то після проведення службового розслідування і переговорів з постачальником, обладнання повертається постачальнику, який усуває дефекти, або виконує поставку нової партії, виплативши неустойку;*

Наведена вище первинна інформація про гірничозбагачувальний комбінат є основою для створення моделі IDEF0 – моделі ГЗК.

Виконаємо моделювання в програмі Ramus Educational. Виділимо суб'єкт, мету і точку зору.

Згідно з методологією IDEF0, модель є деяким тлумаченням системи. Тому суб'єктом моделювання є сама система. Однак система, яку моделюють, ніколи не існує ізольовано: вона завжди пов'язана з навколишнім середовищем. Причому найчастіше важко сказати, де закінчується система і починається середовище. З цієї причини в методології IDEF0 (SADT) підкреслюється необхідність точного визначення меж системи. SADT–модель завжди обмежує свій суб'єкт, тобто модель встановлює точно, що є і що не є суб'єктом моделювання, описуючи те, що входить

в систему, і маючи на увазі те, що лежить за її межами. Обмежуючи суб'єкт, SADT-модель допомагає сконцентрувати увагу саме на описі системі і дозволяє уникнути включення сторонніх суб'єктів. Ось чому SADT-модель повинна мати єдиний суб'єкт.

Отже, **суб'єкт**: функціонування гірничозбагачувального комбінату (ГЗК), різні аспекти діяльності ГЗК. Таким чином, сформульована модель матиме статус "as-is" (як є).

Мета: Представлення моделі функціонування ГЗК (моделі as-is) для проведення аналізу діяльності підприємства та вироблення рекомендацій щодо вдосконалення багатосторонньої діяльності ГЗК.

Точка зору: Керуючий ГЗК.

На другому етапі створимо контекстну діаграму А-0, на якій у вигляді блоку представлений суб'єкт і у вигляді дуг (стрілок) показані ICOM-коди(рисунок 2.1). При оформленні листа з діаграмою А-0 передбачено заповнення верхньої та нижньої рядків діаграми відповідно до методології IDEF0. Так, у правому нижньому куті розташований С-номер, який показує зв'язок з діаграмами різних рівнів декомпозиції, а також з іншими версіями моделі.

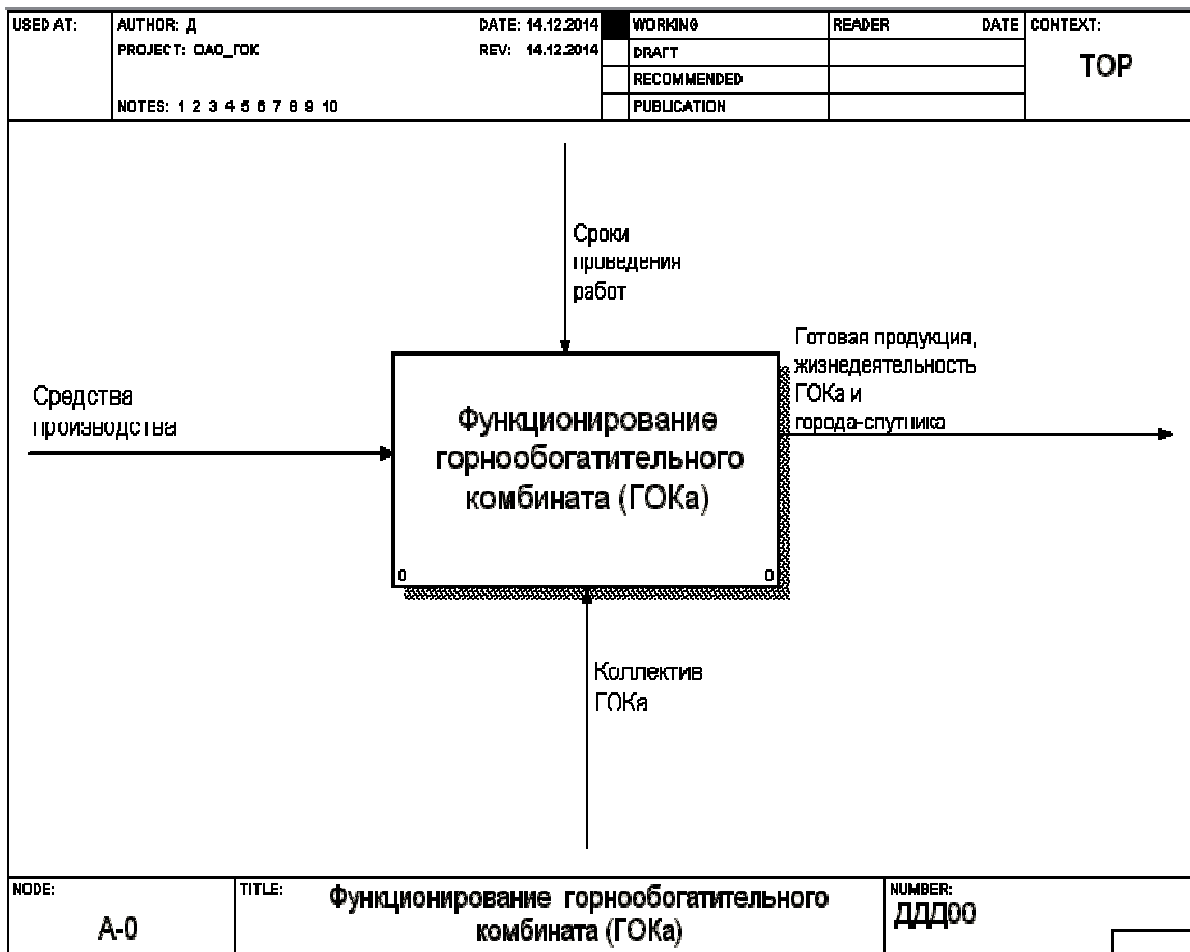


Рисунок 2.1 - Контекстна діаграма А-0

На третьому етапі представимо декомпозицію діаграми А-0 у вигляді діаграми А0 «Функціонування ГЗК»(рисунк 2.2). Діаграму представимо у вигляді п'яти робіт, кожна з яких вказана у відповідному блоці. Блоки розташовані в порядку домінування з лівого верхнього кута в правий нижній. Таким чином, домінуючим є блок 1 – виробництво сировини. Кожен блок повинен обов'язково мати дугу управління відповідно до методології IDEF0. Тому дуга «Терміни проведення робіт» розгалужується і підходить до кожного з п'яти блоків діаграми А0.

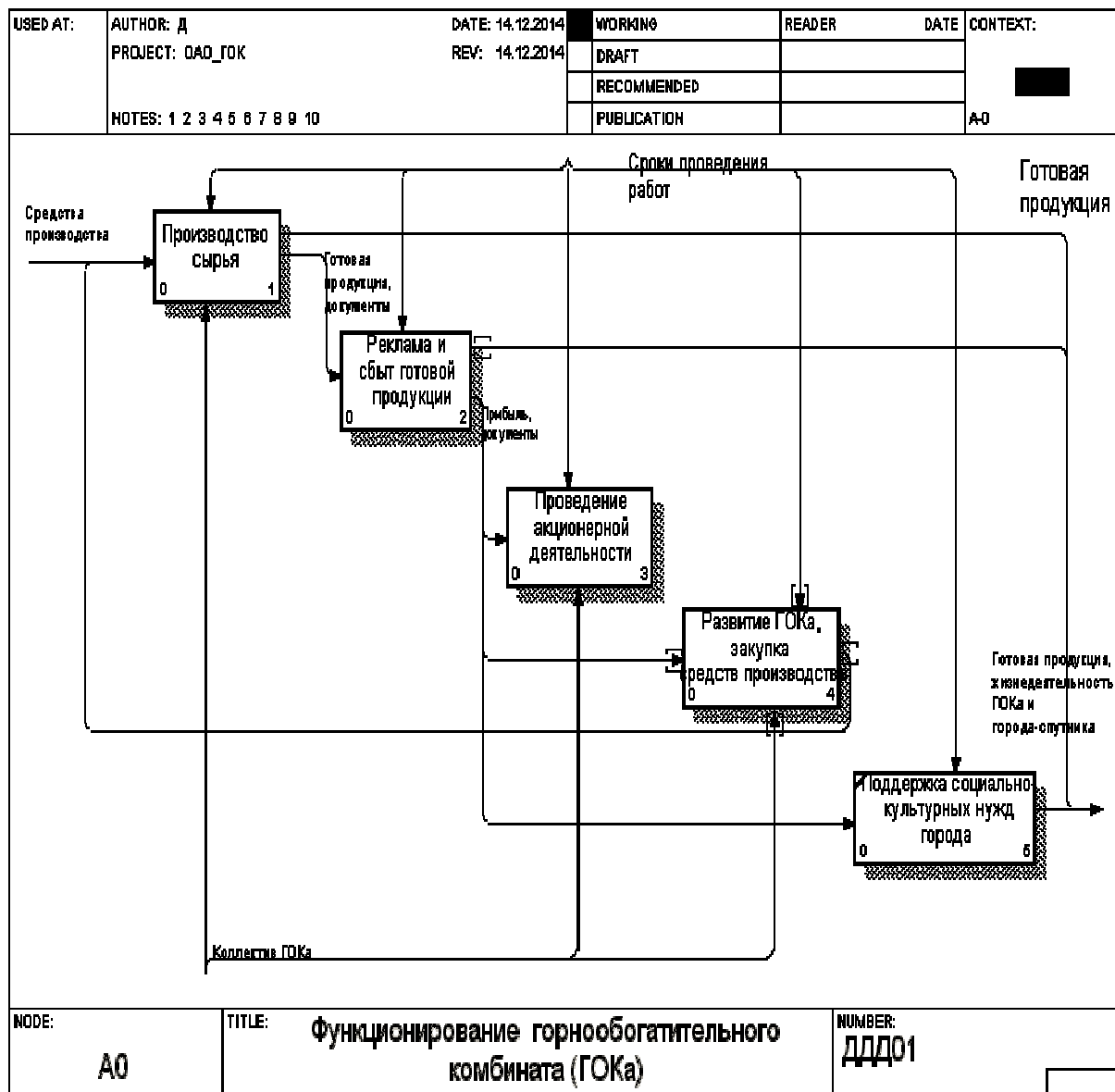


Рисунок 2.2 -Декомпозиція контекстної діаграми. Функціонування ГЗК

Проведемо декомпозицію блоку «Виготовлення сировини» діаграми А-0. На рисунку 2.3 наведено діаграму А1.

Блок «Виготовлення сировини» показаний на діаграмі А1 у вигляді чотирьох блоків. Домінуючим є блок «Контроль якості залізорудного концентрату». Вхідний дугою цього блоку є дуга, відповідна зворотного зв'язку по входу і виходить з блоку «Виготовлення залізорудного концентрату».

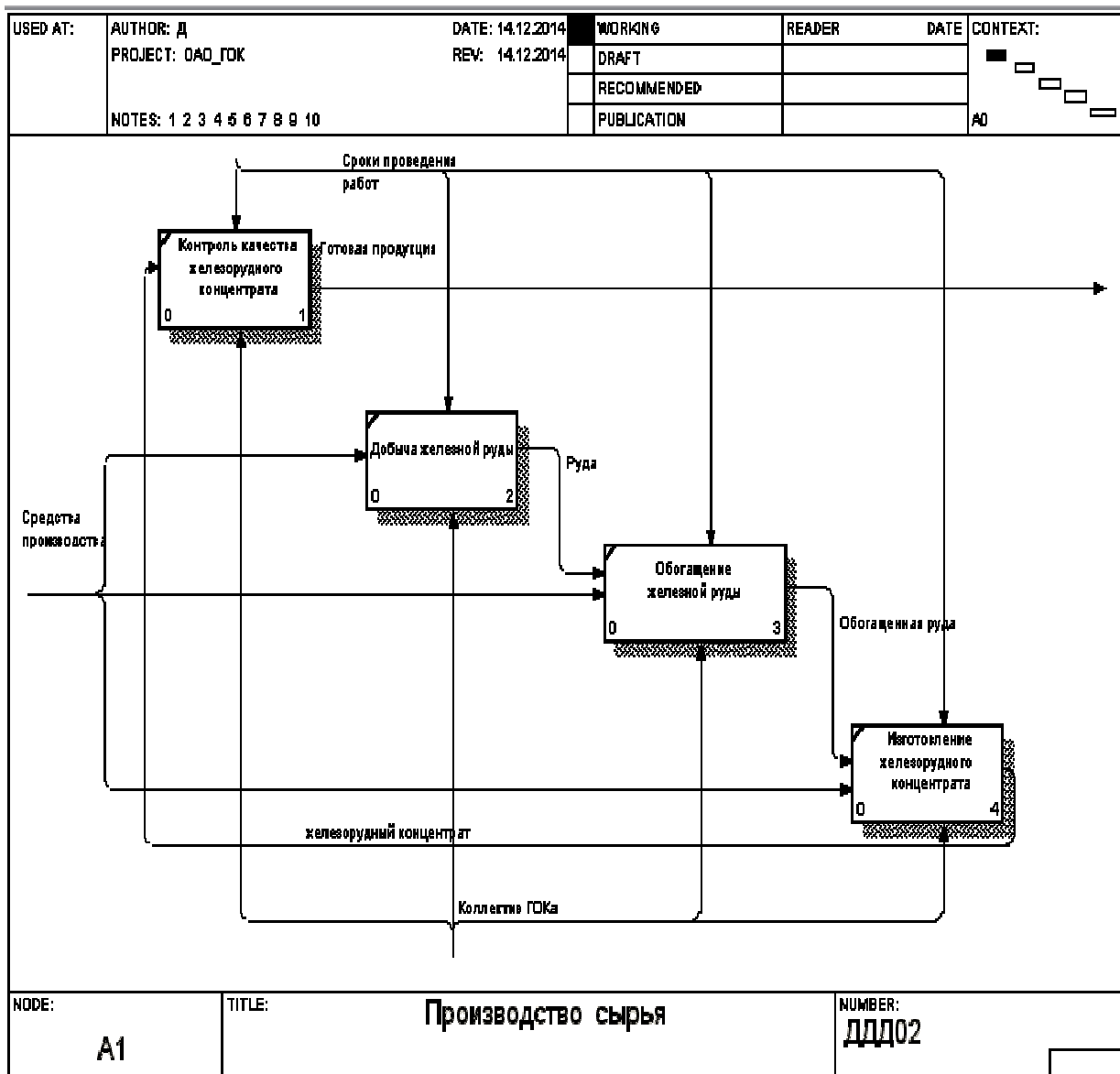


Рисунок 2.3-Діаграма А1

У моделі ГЗК, проведено також декомпозицію другого та третього блоків.

Блок «Реклама і збут готової продукції» розглянуто на діаграмах А2 (рисунок 2. 4) і А23 (рисунок 2.5), тобто показані два рівні ієрархії блоку. Діаграма А2 містить роботи по рекламі і збуту продукції, а діаграма А23 ілюструє роботи по збуту продукції ГЗК.

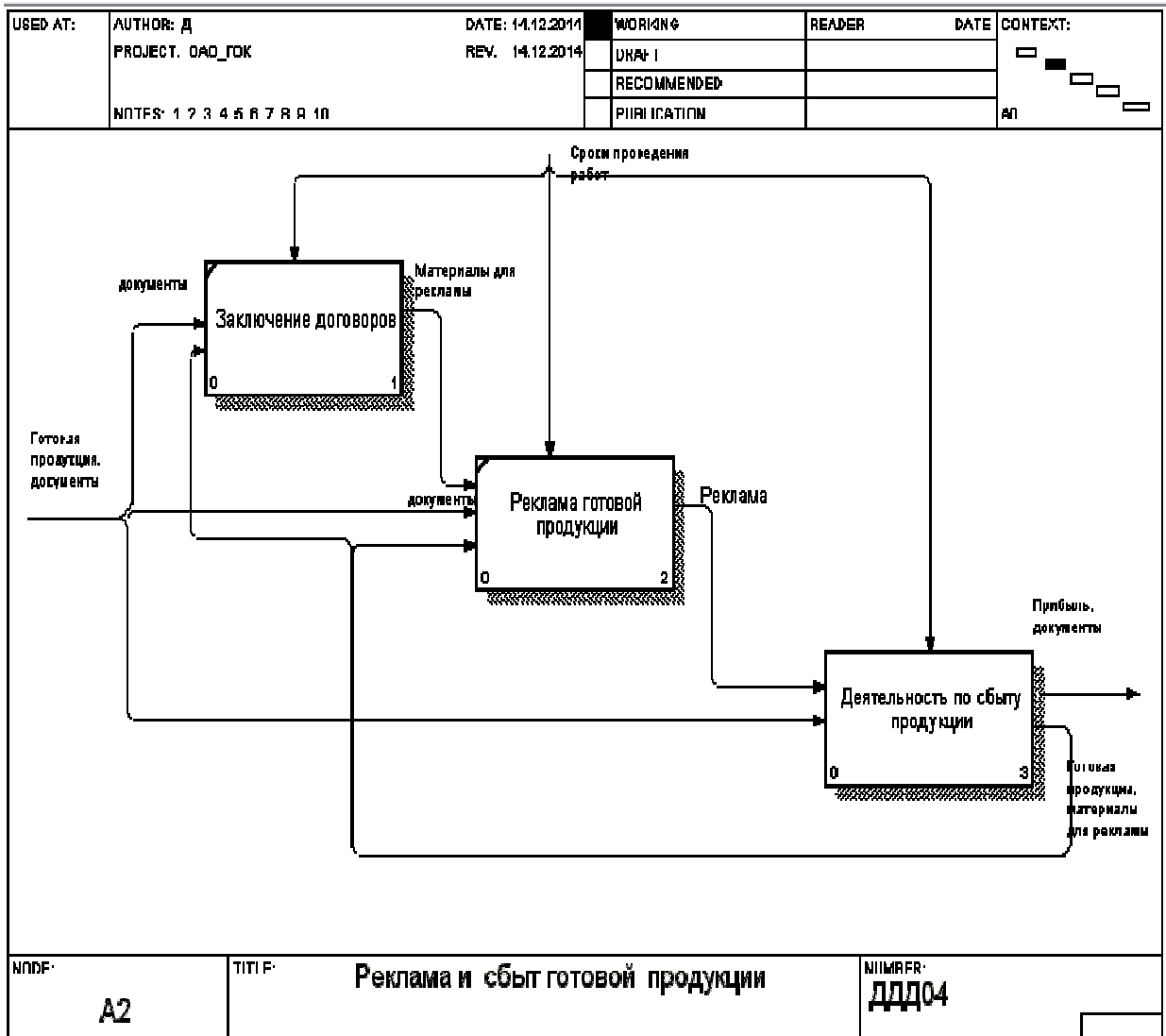


Рисунок 2.4 - Реклама і збут готової продукції

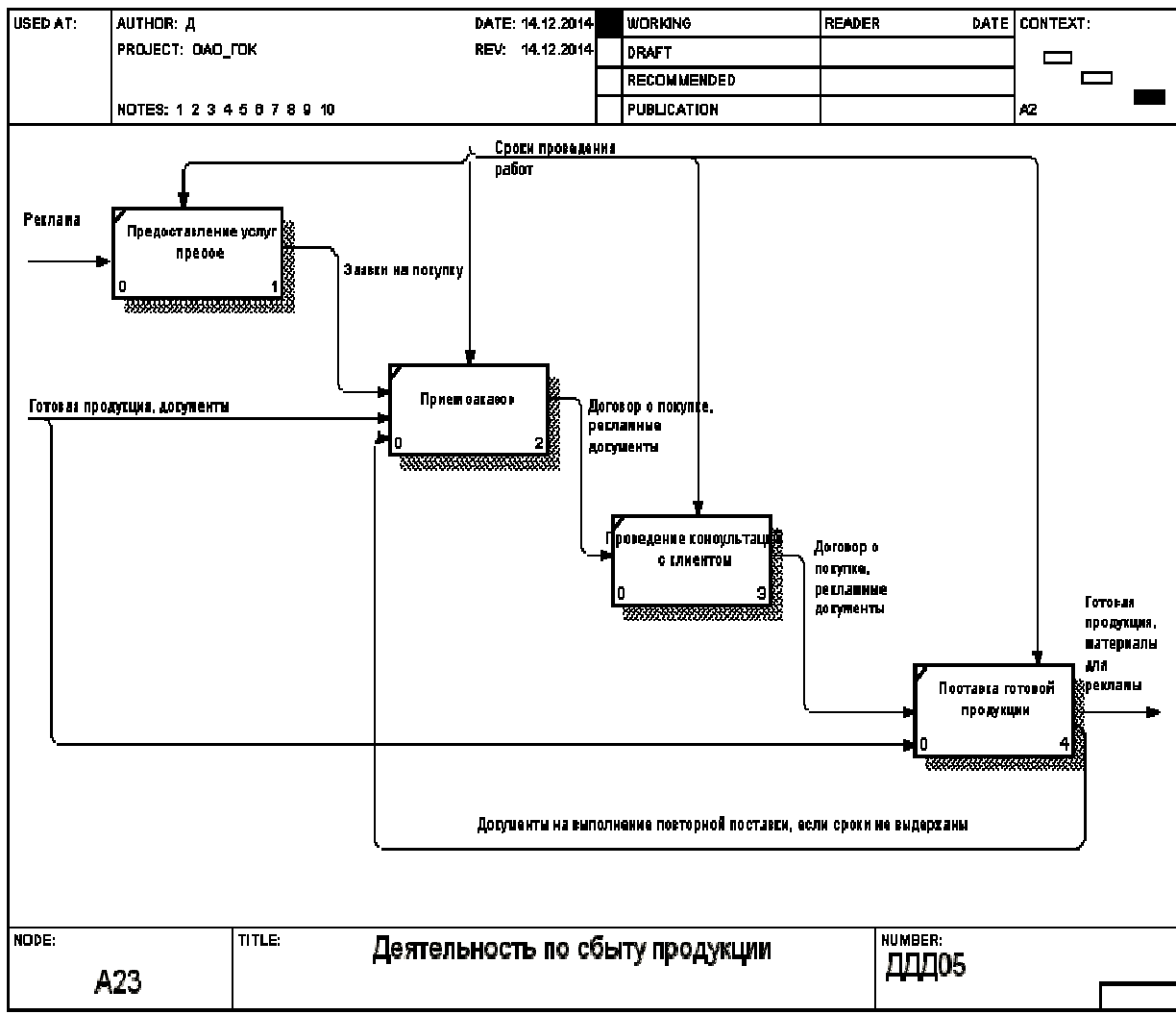


Рисунок 2.5 - Діяльність по збуту продукції

Для блоку «Розвиток ГЗК» діаграми А0 побудована DFD–діаграма, яка відображає процеси поставки обладнання навіть у разі поставки неякісного обладнання (рисунок 2.6).

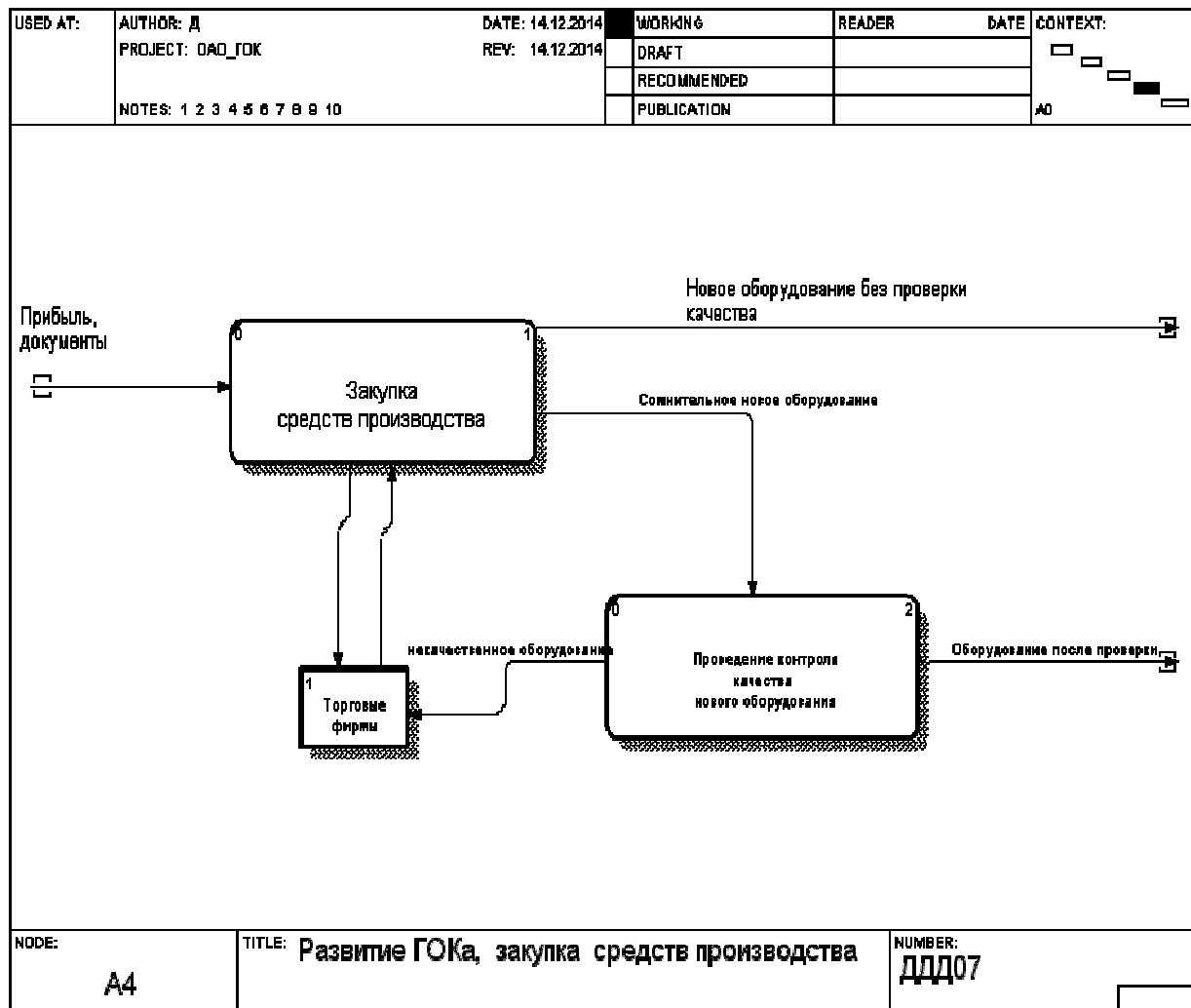


Рисунок 2.6-Розвиток ГЗК, закупка засобів виробництва

Для обліку діаграм в першій версії моделі ГЗК і в наступних її версіях можна використовувати С-номери, які показані в правому нижньому кутку кожної діаграми. Згідно використовуваної методології, С-номер включає ініціали автора моделі (ДДД). Бланк реєстру номерів має вигляд, показаний в таблиці 2.3.

Таблиця 2.3С-номера моделі ГЗК

С-номер	Позначення	Назва діаграми
1	2	3
ДДД00	A-0	Функціонування гірничозбагачувального комбінату (ГЗК)
ДДД01	A0	Функціонування гірничозбагачувального комбінату (ГЗК)
ДДД02	A1	Виробництво сировини

1	2	3
ДДД03	В-0	Проведення розслідування невиконання видобутку в строк (DFD)
ДДД04	A2	Реклама і збут готової продукції
ДДД05	A23	Діяльність по збуту продукції
ДДД06	A3	Проведення акціонерної діяльності
ДДД07	A4	Закупівля засобів виробництва (DFD)

Контрольні запитання для самоперевірки

1. Що являє собою модель в нотації IDEFO?
2. Що позначають роботи в IDEFO?
3. Назвіть порядок найменування робіт?
4. Яка кількість робіт має бути присутнім на одній діаграмі
5. Що називається порядком домінування?
6. Як розташовуються роботи за принципом домінування?
7. Яке призначення сторін прямокутників робіт на діаграмах?
8. Перерахуйте типи стрілок.
9. Назвіть види взаємозв'язків.
10. Що називається граничними стрілками?
11. Поясніть принцип іменування стрілок, які розгалужуються і зливаються.
12. Як провести зв'язок між роботами?
13. Як задати ім'я роботи?
14. Опишіть процес декомпозиції роботи.
15. Як додати роботу на діаграму?
16. Як дозволити тунелювання стрілки?
17. Що описує діаграма DFD?
18. Перерахуйте складові частини діаграми DFD.
19. Що називається зовнішньої сутністю?
20. Що описують сховища?

2.3 Розробка структурної моделі з використанням системи «Ramus Educational»

Створення контекстної діаграми

1. Після запуску програми на екрані з'явиться вікно початку роботи (рисунок 2.7). Виберіть опцію "Создать" та натисніть "ОК".

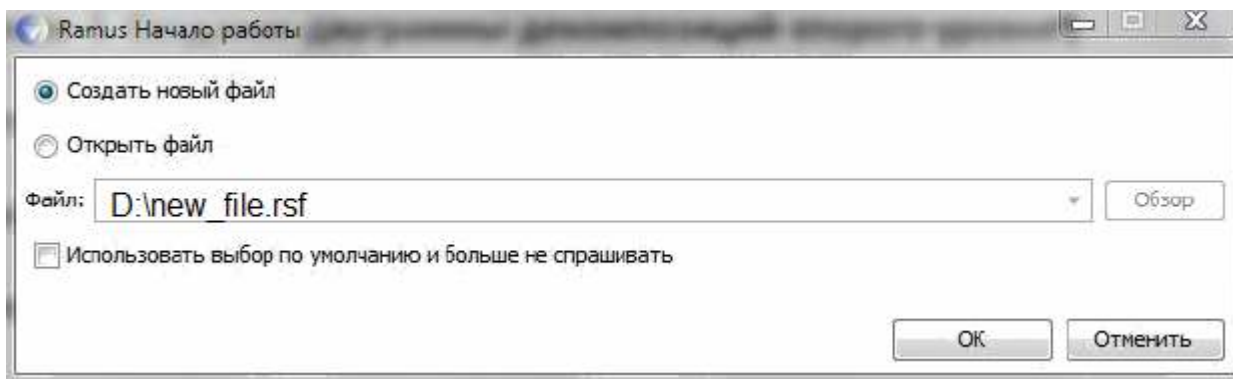



Рисунок 2.7-Діалогове вікно початку роботи в Ramus Educational

2. Внесіть ім'я автора, назву проекту, назву модулю та оберіть опцію "IDEF0". На наступному кроці вкажіть, де використовується модель.


3. Розділ "классификаторы" залиште незаповненим та натисніть "Дальше".

4. В наступному діалоговому вікні натисніть "Окончить" та перейдіть до робочого інтерфейсу програми.

5. Через меню *Диаграмма*→*Свойства модели* можна відкоригувати мета-дані моделі, а саме: назву моделі, опис, місце її використання.

6. Активуйте вікно моделі, клікнувши на область моделювання. Створіть контекстну діаграму, натиснувши на кнопку .

7. Перейдіть в режим редагування контекстної діаграми, натиснувши правою кнопкою миші на об'єкті та вибрав опцію "Редактировать активный элемент". В закладці "Название" введіть назву вашого блока.

8. Створіть стрілки на контекстній діаграмі. Для цього необхідно перейти в режим побудови стрілок за допомогою кнопки , навести курсор на вихідну точку стрілки (ліва, верхня та нижня границя області побудови моделі або права границя контекстної діаграми), після того, як область буде підсвічена чорним кольором, клікнути один раз та аналогічним чином визначити кінець стрілки (права, верхня та нижня границя контекстної діаграми або права границя області

побудови моделі). Переміщувати стрілки та їхні назви можна за принципами стандартного механізму drag&drop.

Створення діаграми декомпозицій

1. Оберіть кнопку переходу на рівень нижче ▼ в панелі інструментів.
2. В діалоговому вікні вкажіть число робіт на діаграмі нижчого рівня, а нотацію декомпозиції – *IDEFO* (рисунок 2.8), потім натисніть "OK". Автоматично буде створена діаграма декомпозиції.

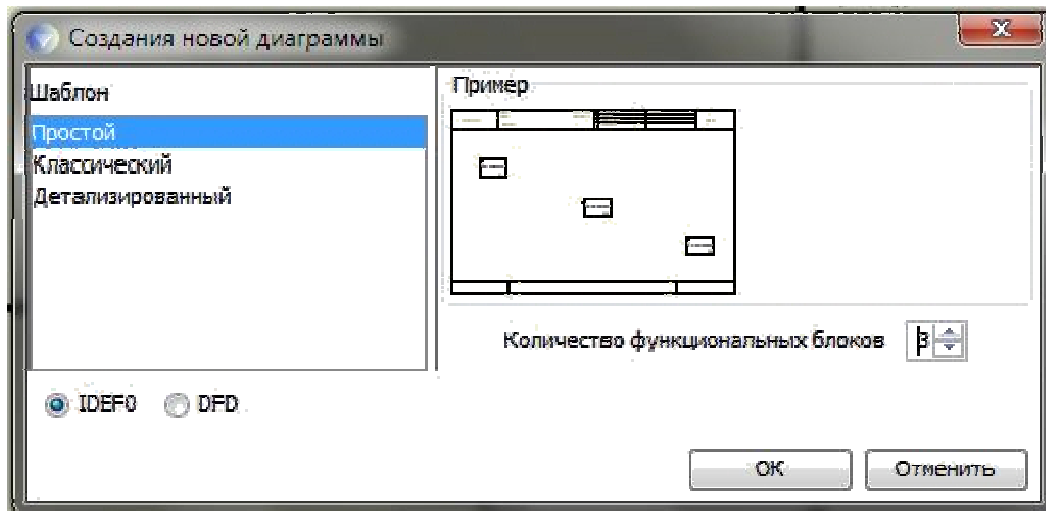


Рисунок 2.8-Діалогове вікно декомпозиції робіт


3. Правою кнопкою миші натисніть по 1-ій роботі, оберіть "Редактировать активный элемент" та на вкладці "Название" вкажіть ім'я роботи. Повторіть операцію для всіх трьох робіт, а також внесіть їхній опис в відповідну вкладку

4. Перейдіть в режим створення стрілок. Проведіть зв'язування граничних стрілок з функціональними об'єктами. Для зв'язування граничних стрілок наведіть курсор на самі стрілки, а не на межі області побудови моделей.




5. Для перейменування стрілок необхідно правою кнопкою миші клацнути по гілці стрілки і ввести її нову назву.

6. Для створення нових внутрішніх стрілок перейдіть в режим побудови стрілок, клікніть лівою кнопкою миші на вихідному блоці і ведіть її до вхідного блоку.

7. Також мається можливість змінити стиль стрілки, її товщину (права кнопка миші → "Редактировать активный элемент" → вкладка "Линия"). Методом drag&drop можна переносити стрілки та їхні назви. При необхідності можливо встановити "тильду" (опція контекстного меню при натисканні на стрілці правою кнопкою миші) для явного зв'язку стрілки і підписи до неї.

8. Для створення граничних стрілок (ця стрілка автоматично не потрапляє на діаграму верхнього рівні і має квадратні дужки у закінченні ) клікніть правою кнопкою миші по квадратним дужкам та оберіть в контекстному меню "Туннель" одну з двох опцій: "Создать стрелку", або "Обозначить туннель круглыми скобками".

Створення діаграми потоків даних (DFD)

1. Створіть контекстну діаграму процесу (Файл → Новый проект).
2. Декомпозуйте створену контекстну діаграму для чого в діалоговому вікні оберіть кількість елементів декомпозиції, тип діаграми –DFD. Натисніть "ОК" та внесіть на діаграму DFD назви робіт.
3. Створіть класифікатори.
4. Внесіть в модель відповідні сховища даних за допомогою кнопки , а також зовнішню сутність , використовуючи кнопку . Майте на увазі, що зв'язок між деякими функціональними об'єктами та сховищами даних може бути двонаправленим (стрілки, що входять и виходять).

2.4 Основи мови моделювання UML

Окремі мови об'єктно-орієнтованого моделювання почали з'являтися в середині 1970-х років, коли різні дослідники і програмісти пропонували свої підходи до об'єктно-орієнтованого аналізу та проектування (ООАП). У період між 1989 –1994 рр. загальне число найбільш відомих мов моделювання зросло з 10 до більш ніж 50. Багато користувачів випробовували серйозні труднощі при виборі мови ООАП, оскільки жодна з них не задовольняла всім вимогам, що пред'являються до побудови моделей складних систем. Ухвалення окремих методик і графічних нотацій в якості стандартів (IDEF0, IDEF1X) не змогло змінити ситуацію непримиренної конкуренції між ними на початку 90-х років, яка отримала назву "війни методів".

До середини 1990-х деякі методи були істотно поліпшені і придбали самостійне значення при вирішенні різних завдань ООАП. Найбільш відомими в цей період стають:

- Метод Граді Буча (Grady Booch), що отримав умовну назву Booch або Booch'91, Booch Lite (пізніше – Booch'93).

- Метод Джеймса Румбаха (James Rumbaugh), найменовано Object Modeling Technique– OMT (пізніше – OMT–2).

- Метод Айвара Джекобсона (Ivar Jacobson), під назвою Object–Oriented SoftwareEngineering– OOSE.

Кожен з цих методів був орієнтований на підтримку окремих етапів ООАП. Наприклад, метод OOSE містив засоби представлення варіантів використання, які мають істотне значення на етапі аналізу вимог у процесі проектування бізнес–додатків. Метод OMT–2 найбільш підходив для аналізу процесів обробки даних в інформаційних системах. Метод Booch'93 знайшов широке застосування на етапах проектування та розробці різних програмних систем.

Історія розвитку мови UML бере початок з жовтня 1994 року, коли Граді Буч і Джеймс Румбах з компанії Rational Software Corporation почали роботу з уніфікації методів Booch і OMT. Незважаючи на те, що самі по собі ці методи були досить популярні, спільна робота була спрямована на вивчення всіх відомих об'єктно–орієнтованих методів з метою об'єднання їхніх переваг. Проект уніфікованого методу (Unified Method) версії 0.8 був опублікований в жовтні 1995 року. Восени того ж року до них приєднався А. Джекобсон, головний технолог компанії Objectory AB (Швеція), з метою інтеграції свого методу OOSE з двома попередніми.

У той час мова UML придбала статус другого стратегічного напрямку в роботі консорціуму OMG (Object Management Group). Саме в OMG створюється команда розробників під керівництвом Р. Солі, яка забезпечила подальшу роботу з уніфікації та стандартизації мови UML. Зусилля групи розробників, в яку входили також Г. Буч, Дж. Румбах і А. Джекобсон, призвели до появи перших документів, що містять власне опис мови UML версії 0.9 (червень 1996 р) і версії 0.91 (жовтень 1996).

Канонічні діаграми мови UML

В рамках мови UML всі уявлення про моделі складної системи фіксуються у вигляді спеціальних графічних конструкцій, що одержали назву діаграм.

Діаграма (diagram) – графічне представлення сукупності елементів моделі у формі зв'язного графа, вершинам і ребрам (дугам) якого приписується певна семантика. Нотація канонічних діаграм – основний засіб розробки моделей на мові UML.

В нотації мови UML визначені наступні види канонічних діаграм:

- варіантів використання (use-case diagram),
- класів (class diagram),

- кооперації (collaboration diagram),
- послідовності (sequence diagram),
- станів (state chart diagram),
- діяльності (activity diagram),
- компонентів (component diagram),
- розгортання (deployment diagram).

Перелік цих діаграм і їх назви є канонічними в тому сенсі, що являють собою невід'ємну частину графічної нотації мови UML. Більше того, процес ООАП нерозривно пов'язаний з процесом побудови цих діаграм. При цьому сукупність побудованих таким чином діаграм є самодостатньою в тому сенсі, що в них міститься вся інформація, яка необхідна для реалізації проекту складної системи.

Кожна з цих діаграм деталізує і конкретизує різні уявлення про модель складної системи в термінах мови UML. При цьому діаграма варіантів використання являє собою найбільш загальну концептуальну модель складної системи, яка є первинною для побудови всіх інших діаграм. Діаграма класів, по своїй суті, логічна модель, що відображає статичні аспекти структурної побудови складної системи.

Діаграми кооперації та послідовностей є різновиди логічної моделі, які відображають динамічні аспекти функціонування складної системи. Діаграми станів і діяльності призначені для моделювання поведінки системи. І, нарешті, діаграми компонентів і розгортання служать для представлення фізичних компонентів складної системи і тому відносяться до її фізичної моделі.

В цілому інтегрована модель складної системи в нотації UML може бути представлена у вигляді сукупності зазначених вище діаграм (рисунок 2.9).



Рисунок 2.9 -Інтегрована модель складної системи в нотації UML
Розглянемо більш детально кожен з видів діаграм.

Діаграма варіантів використання (use case diagram)

Одна з моделей формалізації процесу постановки мети і завдань проекту була запропонована фірмою Rational і увійшла в стандарт мови UML. Для цього застосовуються діаграми варіантів використання (use-case), іноді звані діаграмами прецедентів. Базовими елементами діаграми варіантів використання є варіант використання і актор.

Варіант використання (use-case) – зовнішня специфікація послідовності дій, які система, або інша сутність можуть виконувати в процесі взаємодії з акторами. Варіант використання являє собою специфікацію загальних особливостей поведінки або функціонування модельованої системи без розгляду внутрішньої структури цієї системи. Незважаючи на те, що кожен варіант використання визначає послідовність дій, які повинні бути виконані проектованою системою при взаємодії її з відповідним актором, самі ці дії не зображуються на розглянутій діаграмі.

Варіанти використання характеризуються рядом *властивостей*:

- варіант використання охоплює деяку очевидну для користувачів функцію;
- варіант використання може бути як невеликим, так і досить великим;

- варіант використання вирішує деяку дискретну задачу користувача.

Актор (діюча особа)– це роль, яку користувач грає по відношенню до системи. Кожен актор може розглядатися як якась окрема роль щодо конкретного варіанту використання. Стандартним графічним позначенням актора на діаграмах є фігурка "чоловічка", під якою записується ім'я актора.

Відношення (relationship) – семантичний зв'язок між окремими елементами моделі.

В UML є декілька *стандартних видів відносин* між акторами і варіантами використання:

- *асоціації (association relationship)* – служить для позначення специфічної ролі актора при його взаємодії з окремим варіантом використання; позначається суцільною лінією між актором і варіантом використання.

- *включення (include relationship)* – встановлюється тільки між двома варіантами використання і вказує на те, що задана поведінка для одного варіанта використання включається як складова фрагмента в послідовність поведінки іншого варіанту використання; графічно позначається як відношення залежності у формі пунктирною лінії зі стрілкою, спрямованої від базового варіанту використання до такого, що включається у варіант використання.

- *розширення (extend relationship)* – визначає взаємозв'язок базового варіанту використання з іншим варіантом використання, функціональна поведінка якого задіюється базовим не завжди, а тільки при виконанні додаткових умов; позначається як відношення залежності у формі пунктирною лінії зі стрілкою, спрямованої від того варіанту використання, який є розширенням для базового варіанту використання.

- *узагальнення (generalization relationship)* – позначає спільність властивостей і поведінки варіантів використання. Графічно відношення узагальнення позначається суцільною лінією зі стрілкою у формі незафарбованого трикутника, який вказує на батьківський варіант використання.

На рисунку 2.10 показано приклад діаграми варіантів використання.

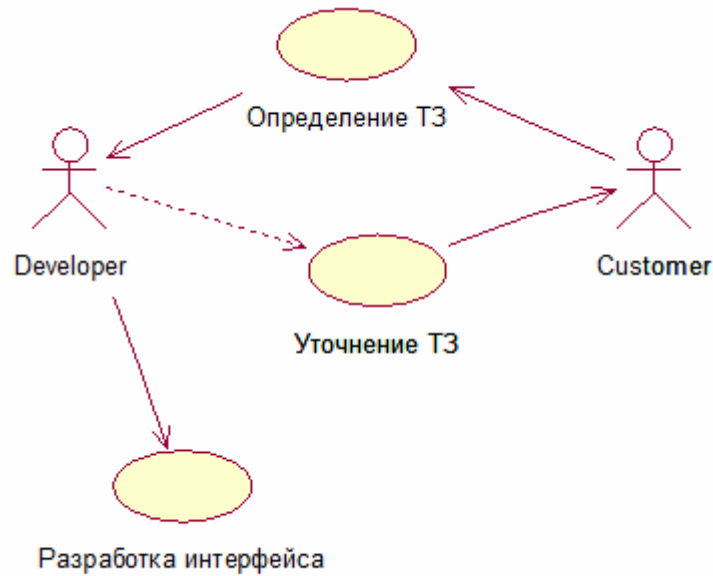


Рисунок 2.10 - Діаграма варіантів використання

Діаграма класів (class diagram)

Центральне місце в методології ООАП займає розробка логічної моделі системи у вигляді діаграми класів.

Діаграма класів показує класи і їхні відносини, тим самим представляючи логічний аспект проекту. Окрема діаграма класів становить певний ракурс структури класів. На стадії аналізу діаграми класів використовуються, щоб виділити загальні ролі та обов'язки сутностей, що забезпечують необхідну поведінку системи. На стадії проектування діаграми класів використовуються, щоб передати структуру класів, які формують архітектуру системи.

Діаграма класів визначає типи об'єктів системи і різного роду статичні зв'язки, які існують між ними. Є два основних види статичних зв'язків:

- *асоціації* – семантичне відношення між двома і більше класами, яке специфікує характер зв'язку між відповідними екземплярами цих класів,
- *підтипи* (працівник є різновидом особистості).

На діаграмах класів зображуються також атрибути класів, операції та обмеження, які накладаються на зв'язку між об'єктами (рисунок 2.11).

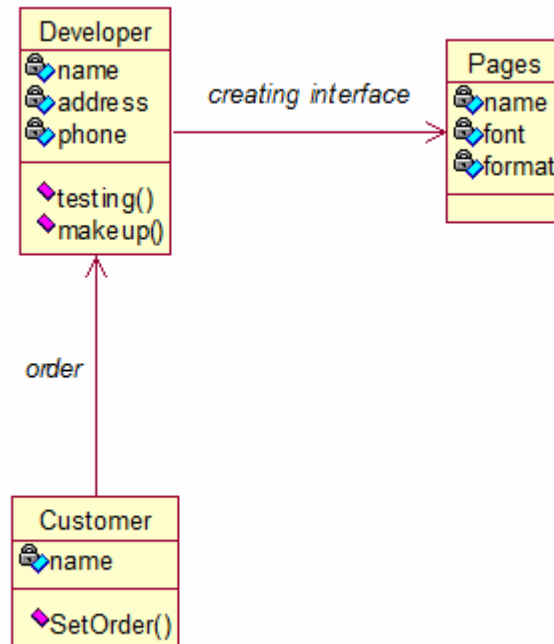


Рисунок 2.11- Діаграма класів

Діаграма діяльностей (activity diagram)

Діаграма діяльності – UML–діаграма, на якій показано розкладання деякої діяльності на її складові частини. Під *діяльністю* (англ. *activity*) розуміється поведінка у вигляді координованого послідовного і паралельного виконання підлеглих елементів – вкладених видів діяльності та окремих дій, з'єднаних між собою потоками, які йдуть від виходів одного вузла до входів іншого.

Діаграми діяльності використовуються при моделюванні бізнес–процесів, технологічних процесів, послідовних і паралельних обчислень.

Кожна діаграма діяльності повинна мати *один початковий і один кінцевий стани*. Вони мають такі ж позначення, як і на діаграмі станів. При цьому кожна діяльність починається в початковому стані і закінчується в кінцевому стані. Саму діаграму діяльності прийнято розташовувати таким чином, щоб дії слідували зверху вниз або зліва направо. У цьому випадку початковий стан буде зображуватися у верхній або лівій частині діаграми, а кінцеве – в її нижній або правій частині. В інтересах зручності візуального представлення на діаграмі діяльності допускається зображати кілька кінцевих станів. У цьому випадку всі їх прийнято вважати еквівалентними. На рисунку 2.12 представлено приклад діаграми діяльності.



Рисунок 2.12 - Діаграма діяльності

Діаграма станів об'єкта(state chart diagram)

Моделювання поведінки об'єктів і системи в цілому ґрунтується на понятті стану.

Стан (state) – умова, або ситуація в ході життєвого циклу об'єкта, протягом якої він задовольняє логічні умові, виконує певну діяльність або очікує події.

Стан може бути заданий у вигляді набору конкретних значень атрибутів об'єкта деякого класу, при цьому зміна окремих значень цих атрибутів буде відображати зміну стану модельованого об'єкта або системи в цілому. Однак не кожен атрибут класу може характеризувати стан його об'єктів. Як правило, мають значення тільки ті властивості елементів системи, які відображають динамічний, або функціональний аспект її поведінки. Стан на діаграмі зображується прямокутником із закругленими вершинами.

Діаграма станів (state chart diagram) будується для одного класу об'єктної моделі, при цьому клас потрібно вибирати таким чином, щоб були явно видно кілька різних станів. Стан можна добре ідентифікувати по зміні поведінки об'єкта класу.

Крім звичайних станів на діаграмі станів можуть розміщуватися псевдостани.

Псевдостан (pseudo-state) – вершина в кінцевому автоматі, яка має форму стану, але не володіє поведінкою. Прикладами псевдо станів, які визначені в мові UML, є початковий і кінцевий стани.

Початковий стан (start state) – різновид псевдостану, що позначає початок виконання процесу зміни станів або знаходження модельованого об'єкта в складеному стані. У цьому стані знаходиться об'єкт за умовчанням в початковий момент часу. Графічно початковий стан в мові UML позначається у вигляді замальованого круга, з якого може тільки виходити стрілка-перехід. Кожна діаграма або під-діаграма станів повинна мати єдиний початковий стан.

Кінцевий стан (final state) – різновид псевдостану, що позначає припинення процесу зміни станів або знаходження модельованого об'єкта в складеному стані. У цьому стані повинен перебувати модельований об'єкт або система за замовчуванням після завершення роботи кінцевого автомата. Графічно кінцевий стан в мові UML позначається у вигляді замальованого круга, поміщеного в коло, в яке може тільки входити стрілка-перехід. Кожна діаграма станів або підстанів може мати кілька кінцевих станів, при цьому всі вони вважаються еквівалентними на одному рівні вкладеності станів.

На рисунку 2.13 наведено приклад діаграми станів.

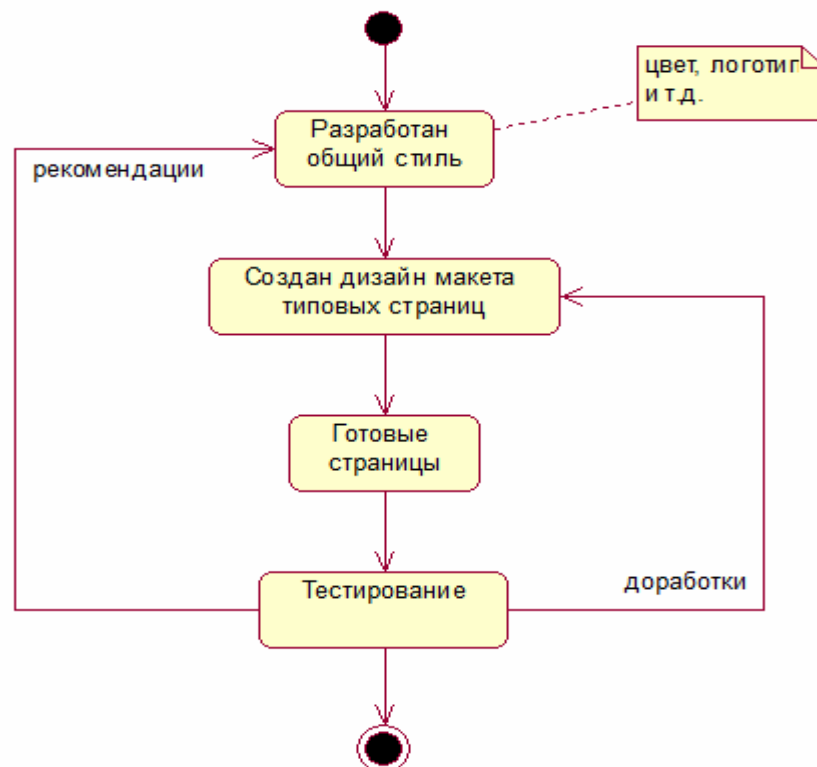


Рисунок 2.13- State-діаграма

Діаграма послідовності (sequence diagrams)

Діаграма послідовності (*sequence diagram*) – діаграма, на якій показані взаємодії об'єктів, впорядковані за часом їх прояву.

На діаграмі послідовності *об'єкт* зображується у вигляді прямокутника на вершині пунктирної вертикальної лінії.

Ця вертикальна лінія називається *лінією життя (life line)* об'єкта. Вона являє собою фрагмент життєвого циклу об'єкта в процесі взаємодії.

Кожне *повідомлення* представляється у вигляді стрілки між лініями життя двох об'єктів. Повідомлення з'являються в тому порядку, як вони показані на діаграмі (зверху вниз). Кожне повідомлення може бути позначено ім'ям, при бажанні можна показати також аргументи і деяку керуючу інформацію.

Побудову діаграми послідовності доцільно починати з виділення з усієї сукупності класів лише тих, об'єкти яких беруть участь у взаємодії, яку моделюють. Після цього всі об'єкти наносяться на діаграму, з дотриманням порядку ініціалізації повідомлень. Тут необхідно встановити, які об'єкти будуть існувати постійно, а які тимчасово – тільки на період виконання ними необхідних дій.

Коли об'єкти визначені на діаграмі, можна приступати до специфікації повідомлень. При цьому необхідно враховувати ті операції, які мають класи відповідних об'єктів в моделі системи.

На рисунку 2.14 показано діаграму послідовності.

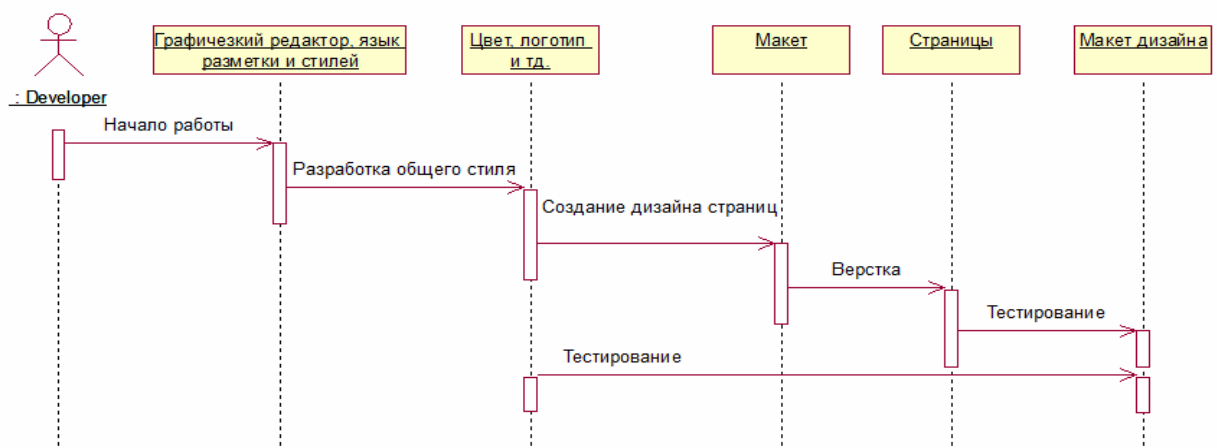


Рисунок 2.14 - Діаграма послідовності

Контрольні запитання для самоперевірки

1. У чому сенс варіантів використання?
2. Яке призначення діаграм варіантів використання?
3. Назвіть основні властивості варіантів використання.
4. Назвіть основні компоненти діаграм варіантів використання.
5. Що таке "дійова особа"?
6. Яку роль можуть грати дійові особи по відношенню до варіанту використання?
7. Яке призначення діаграм класів?
8. Назвіть основні компоненти діаграм класів.
9. Назвіть основні типи статичних зв'язків між класами.
10. Що являє собою асоціація?
11. У чому відмінність атрибутів від асоціацій?
12. У чому сенс узагальнення?
13. Яке призначення обмежень на діаграмах класів?
14. Яке призначення мають діаграми взаємодії?
15. Як відносяться між собою діаграми варіантів використання і діаграми взаємодії?
16. Назвіть два види діаграм взаємодії.
17. Що таке "життєва лінія" на діаграмі послідовності?
18. Як на діаграмі послідовності представляються повідомлення?
19. Як відображається умовне поведінка на діаграмах взаємодії?
20. Яке призначення діаграм стану?
21. Як відображаються дії і діяльності на діаграмах стану?
22. Що таке умовний перехід і як він описується на діаграмі?
23. Які особливі стани об'єкта відображаються на діаграмі?
24. Які переваги та недоліки діаграм стану?

3 КОНТРОЛЬНА РОБОТА

Тема роботи: розробка структурної моделі з використанням методології IDEFO у системі Ramus Educational; розробка об'єктної моделі з використанням мови UML у системі Rational Rose.

3.1 Завдання

Завдання 1. Розробити структурну модель за допомогою методології структурного аналізу IDEFO в середовищі Ramus Educational.

Вивчити основні принципи методології IDEFO, провести розробку IDEFO – моделі в середовищі Ramus Educational за варіантом. Побудувати діаграму потоків даних (DFD) і провести кількісну оцінку діаграм.

Завдання 2. Розробити модель системи з використанням мови UML (додаток Rational Rose).

Провести розробку діаграми класів, варіантів використання, взаємодії, станів, діяльностей і послідовностей згідно обраної предметної області.

3.2 Варіанти контрольних робіт

Варіант обрати за бажанням та погодити з викладачем

Теми робіт для Завдання 1

№ варіанта	Тема предметної області
1	2
1	Програмна реалізація побудови моделі складного об'єкту. з використанням будь-якої мови програмування
2	Комп'ютерне моделювання складних об'єктів.
3	Створення додатку для шифрування текстової інформації
4	Програмна реалізація розрахунку арифметичного виразу з використанням мереж Петрі
5	Розробка модуля автоматизованої інформаційної системи.
6	Розробка модуля автоматизованої системи управління відділу кадрів підприємства
7	Розробка реляційної бази даних складських запасів магазину методами MySQL
8	Розробка реляційної бази даних видавництва мовою SQL
9	Розробка нечіткої підсистеми оцінки вартості об'єктів

	нерухомості
1	2
10	Порівняльний аналіз та комп'ютерна реалізація методів моделювання систем масового обслуговування
11	Розробка бази даних для обліку діяльності автосалону
12	Створення веб-сайту інтернет-магазину
13	Проектування web- сайту
14	Розробка і реалізація модуля системи автоматизації укладення договорів на постачання будівельних матеріалів
15	Створення програмного продукту для банківської системи
16	Проектування та створення серверу
17	Моделювання вантажоперевезень фірми
18	Моделювання підприємства з виробництва рекламних щитів
19	Розробка моделі фірми, яка надає сервісні послуги з ремонту та продажу комп'ютерної техніки
20	Розробка моделі технологічного процесу

Орієнтовні теми робіт для Завдання 2

№ варіанта	Тема предметної області
1	2
1	Робота супермаркету
2	Розробка програмного забезпечення
3	Придбання квитків
4	Замовлення їжі в ресторані
5	Робота магазину електроніки
6	Розробка ВЕБ-додатку
7	Моделювання технічного пристрою
8	Моделювання модулю банківської системи надання кредиту
9	Моделювання модулю банківської системи начислення процентів депозитного рахунку
10	Моделювання автоматизованої системи для сільського господарства
11	Проектування інтернет-магазину

Н.Л. Дорош, Ю.В. Бабенко «Проектування організаційних і технологічних інформаційних управляючих систем»

12	Розробка системи розрахунку за комунальні послуги
1	2
13	Розробка системи збуту продукції
14	Розробка системи «Бібліотека»
15	Моделювання ведення складського господарства
16	Розробка системи «Деканат»
17	Розробка системи «Тесування з дисципліни»
18	Розробка модулю системи «Надання послуг»
19	Розробка модулю системи відділу кадрів
20	Моделювання комп'ютерної мережі підприємства

ВИСНОВКИ

Робоча програма, методичні вказівки та індивідуальні завдання до вивчення дисципліни «Проектування організаційних і технологічних інформаційних управляючих систем» призначені для студентів спеціальності 122 «Комп'ютерні науки» заочної форми навчання.

Приведена технологічна карта для дисципліни «Проектування організаційних і технологічних інформаційних управляючих систем».

Розглянуті основні питання дисципліни «Проектування організаційних і технологічних інформаційних управляючих систем»:

- принципи побудови інформаційних управляючих систем,
- задачі структурного та об'єктного підходів проектування інформаційних систем;
- інструментальні засоби проектування компонентів ІС.

Наведено короткі відомості з розробки інформаційних систем у програмах як Ramus Educational та Rational Rose.

Матеріал, який викладено, призначений для самостійної роботи студентів заочної форми навчання і може бути використаний для виконання контрольної роботи з дисципліни «Проектування організаційних і технологічних інформаційних управляючих систем».

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. Дэвид А. Марка и Клемент Мак Гоуэн. Методология структурного анализа и проектирования SADT. 241 с., - с ил.
2. Вендров А.М. «Практикум по проектированию программного обеспечения экономически хинформационных систем». – М.: «Финансы и статистика», 2006 г.– с.192
3. Интернет ресурс: Ramus–кросс платформенная система моделирования и анали за бизнес–процессов. Режим доступа: <http://ramussoftware.com/>
4. Буч Г. Объектно–ориентированный анализ и проектирование с примерами приложений на С+, 2–е изд./Пер. с англ. –М.: «ИздательствоБином», СПб.: «Невскийдиалект», 2000.– 560 с.
5. Попов А.И. Свободные инструменты проектирования информационных систем: учебн– метод. Пособие / А.И. Попов . Архангельск: ИПЦ САФУ, 2012. – 78 с.
6. Методичні вказівки до виконання лабораторних робіт з дисципліни "Проективання організаційних і технологічних інформаційних управляючих систем". Частина 1: Методологія IDEF0. Програми BPWIN, ERWIN". Для студентів спеціальностей 7(8).080401, 7(8).080402/Укл.:Н.Л. Дорош, В.І. Кузнецов, О.О. Кавац, – Дніпропетровськ: НМетАУ, 2006, – 58с.
7. Бенькович Е.С., Колесов Ю.Б., Сениченков Ю. Б. Практическое моделирование динамических систем – СПб; БХВ–Петербург, 2002. – С. 387.
8. Леоненков А. Самоучитель UML. Серия "Самоучитель". –СПб.: BHV–Санкт–Петербург, 2002 – С. 304 с.: ил.
9. Дж.Рамбо, М.Блаха. Объектно–ориентированное моделирование и разработка – СПб; Питер, 2007. – С. 544, с ил.
10. Трофимов С.А. Case–технологии: практическая работа в RationalRose – М: ЗАО «Издательство БИНОМ», 2001г.– 272с.
11. Федотова Д.Э. Семенов Ю.Д. Чижик К.Н. CASE–технологии: Практикум. – М: Горячая линия. – Телеком, 2003. – 160 с.