**Практична робота №2. Основи алгоритмізації обчислювальних процесів. Принципи побудови алгоритму вирішення задач.**

**Мета:** навчитися надавати в графічній формі схему алгоритму процесу оброблення інформації, використовуючи засоби UML – моделювання як інструменту спілкування економіста з розробником програмного забезпечення.

**Завдання практичної роботи**

1. Ознайомтесь з постановкою задачі та теоретичною частиною.
2. Розглянути надану на рис.1 блок-схему системи ведення звітності головного офісу мережі супермаркетів, яка щоденно збирає звіти про продаж товарів в своїх магазинах, виконує їх перевірку та накопичення, а також формує зведений звіт про продажі. Доповнити цю схему необхідними виробничими процесами.
3. Визначити осіб (акторів), які будуть працювати з системою.
4. Визначити вимоги до системи через діаграму прецедентів (варіантів використання). Приклад надано на рис.2, для розглянутих на ПР01 блок-схем, цю схему потрібно доповнити процесами контроля форм, відсилки фори, накопичення форм.
5. Визначити поведінку системи через діаграму послідовностей (необов‘язкове завдання, *додатково 2 бали*) спираючись на приклад, наданий на рис.5 .
6. Результати оформити відповідно до вимог оформлення звітів.

Для створення UML-діаграм можна використати пропонується ряд безкоштовних інструментів за наданим нижче посиланням:

<https://www.quality-assurance-group.com/top-10-bezkoshtovnyh-onlajn-redaktoriv-dlya-stvorennya-uml-diagram-na-probu/> - перелік безкоштовних on-line редакторів. При відсутності можливості використати такі інструменти схему можна накреслити в зошиті, сфотографувати і вкласти файл з фото в звіт.

**Результати надсилати на електронну адресу викладача** [**t.i.lumpova@gmail.com**](mailto:t.i.lumpova@gmail.com)

Файл повинен мати назву в такому форматі:

**АР<Номер групи><Номер лекції / практичної / лабораторної [літера позначення типу роботи L – лекція, P – практична]<Прізвище англійською>**. Наприклад, **АРPTBD-2102**Fbuts.doc.

Не копіюйте фрагментів з різних інформаційних джерел, подумайте і викладіть свою точку зору. При наявності робіт -"близнюків" відповідь буде зараховуватися першому за часом надсилання.

Всі запитання, що виникнуть, надсилайте на електронну адресу викладача. Тему в заголовку листа записати

**АР <Номер групи>-Запитання-<Прізвище >**.

**Строк виконання цієї роботи - 09.02.2024**

**ПОСТАНОВКА ЗАДАЧІ**

В загальних рисах блок-схема системи ведення звітності головного офісу мережі супермаркетів, яка щоденно збирає звіти про продаж товарів в своїх магазинах, виконує їх перевірку та накопичення, а також формує зведений звіт про продажі, може виглядати як надано на рис.1.

При постановці задачі для розробника визначаються:

* очікувані результати;
* вхідні дані;
* процес перетворення вхідних даних у результати.

У вас є щоденний звіт, який потрібно прийняти, проконтролювати, в разі потреби виправити і зберегти для подальшої обробки ( в загальних рисах це представлено на рис.1).

При спілкуванні з розробником потрібно визначити осіб (акторів), які будуть працювати з системою, та вимоги до системи. Для цього потрібно відповісти на запитання:

1. Кого цікавить конкретна системна вимога?
2. Яку роль відіграватиме система в організації?
3. Хто отримає користь від використання системи?
4. Хто буде постачати систему інформацією, використовувати її та отримувати інформацію від системи?
5. Хто підтримуватиме систему?
6. Чи використовує система зовнішні ресурси?
7. Чи виступає будь-який член системи в декількох ролях?
8. Чи грають різні актори однакову роль?

На рис.1 представлена схема оброблення звітів в системі.

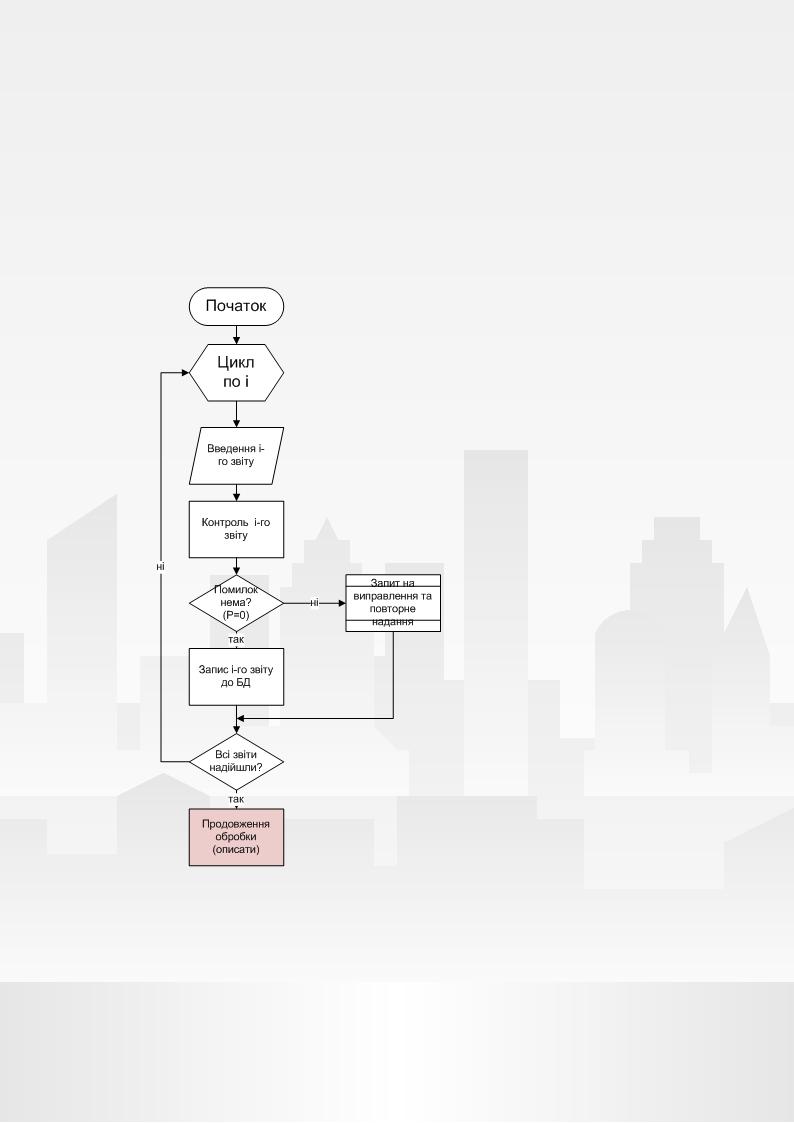


Рисунок 1 - Блок-схема системи ведення звітності головного офісу (початок)

На рис.2 надано діаграму прецедентів, яка в загальних рисах надає огляд вимог щодо оброблення звітності.

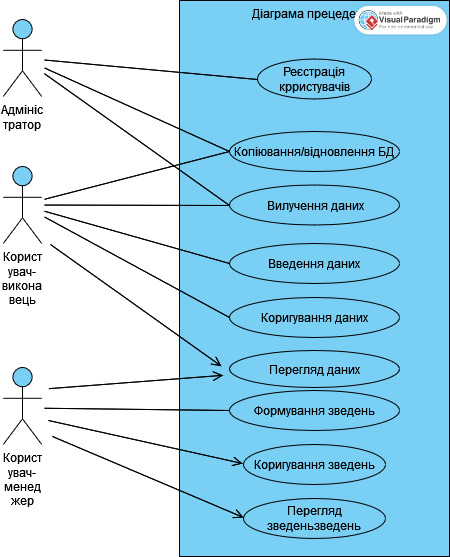


Рисунок 2 – Діаграма прецедентів (виконано на https://online.visual-paradigm.com/ru/diagrams/features/uml-tool/)

**ТЕОРЕТИЧНІ ВІДОМОСТІ**

**Діаграма варіантів / випадків використання або прецедентів**

Для визначення вимог зазвичай використовується *Діаграма* *варіантів / випадків використання або прецедентів* (use case diagram) показує дієвих осіб (людей або інших користувачів системи), випадки використання (сценарії використання системи) та їх взаємодію. Діаграми випадків використання призначено для полегшення обміну інформацією між майбутніми користувачами системи і замовником, вони особливо корисні для визначення переліку можливостей, які повинна мати система. За діаграмами випадків використання можна сказати, що система має робити, але не те, як вона досягає потрібних результатів, для останнього ці діаграми просто не придатні. Приклад діаграми випадків використання надано на рис.2.

Розпочинають створення діаграм використання із ідентифікації варіантів використання (які ще називають прецедентами) та дійових осіб. Це необхідно для того, щоб:

* чітко розмежувати систему і її оточення;
* визначити, які дійові особи і як саме взаємодіють з системою, який функціонал (варіанти використання) очікується від системи;
* визначити і описати в словнику предметної області (глосарії) загальні поняття, які необхідні для детального опису функціонала системи (прецедентів).

Подібний вид діяльності зазвичай виконується в такій послідовності:

1. Визначення дійових осіб.
2. Визначення варіантів використання.
3. Складання опису кожного варіанту використання.
4. Опис моделі прецедентів в цілому (цей етап включає створення словника предметної області).

Спочатку вимоги оформляються у вигляді звичайного текстового документа, який створюється або самим користувачем, або користувачем і розробником разом. Надалі вимоги оформляють у вигляді таблиці. У ліву колонку поміщають прецеденти, а в праву - дійових осіб, що беруть участь в прецеденті.

Приклад. Необхідно створити форум з наступним функціоналом. Читати повідомлення мають право всі, створювати повідомлення та теми – зареєстровані користувачі, видаляти повідомлення та теми – модератори й адміністратори. Система повинна бути написана на php.

Таблиця з описом вимог може бути, наприклад, такою:

|  |  |
| --- | --- |
| Прецедент | Дійова особа |
| Читати повідомлення | Користувач (гість), зареєстрований користувач, модератор, адміністратор |
| Створити повідомлення | Зареєстрований користувач, модератор, адміністратор |
| Створити тему | Зареєстрований користувач, модератор, адміністратор |
| Видалити повідомлення | Модератор, адміністратор |
| Видалити тему | Модератор, адміністратор |

В таблиці нема відомостей про те, що система повинна бути написана на php, адже це нефункціональна вимога.

Наступним етапом є побудова діаграми варіантів використання.

У самому загальному випадку, діаграма варіантів використання являє собою граф спеціального виду, який є графічною нотацією для представлення певних варіантів використання, акторів і відносин між цими елементами. Окремі елементи діаграми можуть буду розміщені у прямокутнику, який позначає границі проектованої системи. Відносини, які можуть бути зображені на даному графі, являють собою тільки фіксовані типи взаємозв'язків між акторами й варіантами використання, які в сукупності описують сервіси або функціональні вимоги до системи, що моделюється.

Базовими елементами діаграми варіантів використання є варіант використання й актор.

**Актор**

Актором(actor) або діючою особою називається будь-який об'єкт, суб'єкт або система, які взаємодіють з бізнес-системою, що моделюється, ззовні. Це може бути людина, технічний пристрій, програма або будь-яка інша система, що служить джерелом впливу на систему, що моделюється, так, як це визначає розроблювач.

Стандартним графічним позначенням актора на діаграмах є фігурка "чоловічка", під якою записується ім'я актора (рис. 3).

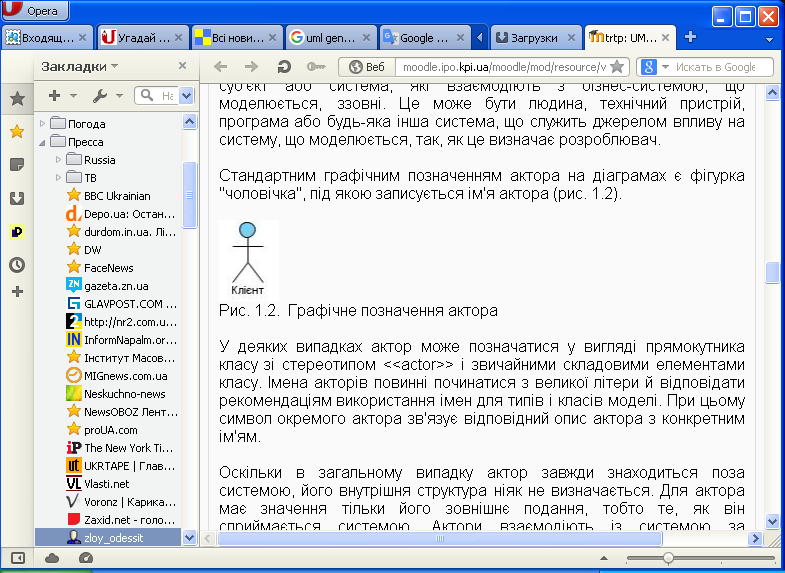


Рисунок 3 - Графічне позначення актора

Оскільки в загальному випадку актор завжди знаходиться поза системою, його внутрішня структура ніяк не визначається. Для актора має значення тільки його зовнішнє подання, тобто те, як він сприймається системою. Актори взаємодіють із системою за допомогою передачі й прийому повідомлень від варіантів використання. Повідомлення являє собою запит актором сервісу від системи та одержання цього сервісу. Ця взаємодія може бути виражена за допомогою асоціацій між окремими акторами та варіантами використання. Крім цього, з акторами можуть бути зв'язані інтерфейси, які визначають, яким чином інші елементи моделі взаємодіють із цими акторами.

Актор — це зовнішній чинник (поза межами системи), який взаємодіє з системою шляхом участі (і часто ініціювання) у випадку використання. Акторами, на практиці, можуть бути звичайні люди (наприклад, користувачі системи), інші комп’ютерні системи або зовнішні події.

Акторам відповідають не реальні люди або системи, а лише їх ролі. Це означає, що коли особа у різний спосіб взаємодіє з системою (виконуючи різні ролі), їй відповідають декілька акторів.

**Випадок / варіант використання (use-case)**

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 системи, що моделюється, 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.

Зміст варіанту використання може бути надано у формі додаткового пояснювального тексту, що розкриває зміст або семантику дій при виконанні даного варіанта використання. Такий пояснювальний текст називається *текстом-сценарієм* або просто сценарієм.

Окремий варіант використання позначається на діаграмі еліпсом, усередині якого міститься його коротке ім'я у формі дієслова (рис. 4, а) або іменника (рис. 4, б) с пояснювальними словами. Сам текст імені варіанта використання повинен починатися із великої літери.

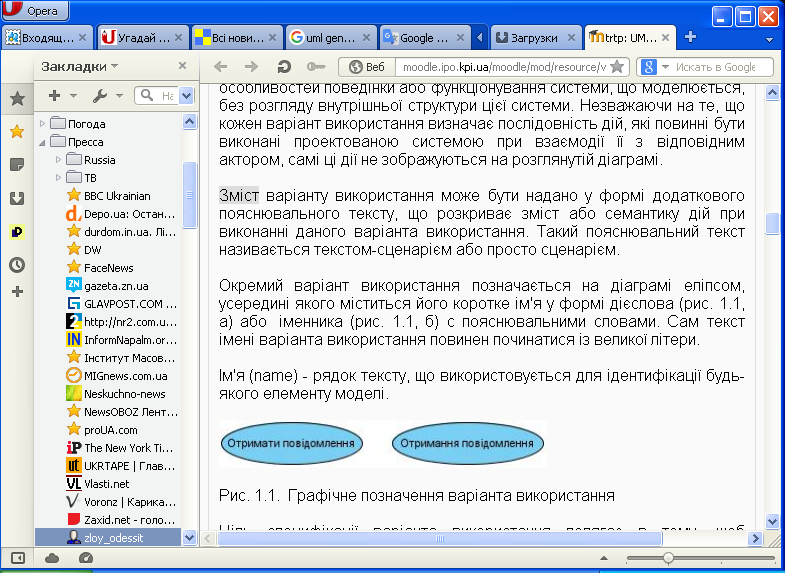


Рисунок 4 - Графічне позначення варіанта використання

Діаграма варіантів використання містить скінчену множину варіантів використання, які в цілому повинні визначати всі можливі сторони очікуваної поведінки системи. Для зручності множина варіантів використання може розглядатися як окремий пакет. Застосування варіантів використання на всіх етапах роботи над проектом дозволяє не тільки досягти необхідного рівня уніфікації позначень для зображення функціональності підсистем і системи в цілому, але і є потужним засобом послідовного уточнення вимог до проектованої системи на основі їх ітеративного обговорення з усіма зацікавленими фахівцями.

Випадок/варіант використання визначає, з точки зору акторів (користувачів), групу дій у системі, які призводять до конкретного видимого результату.

Випадки використання є описом типових елементів взаємодії користувачів системи з самою системою. Вони відповідають зовнішньому інтерфейсу системи і визначають форму вимог до того, що має робити система (зауважте, лише «що», а не «як»).

Під час роботи з випадками використання важливо пам’ятати декілька правил:

* Кожен випадок використання має бути пов’язано принаймні з одним актором
* У кожного з випадків використання має бути ініціатор (тобто актор)
* Кожен з випадків використання має призводити до відповідного результату (результату з “комерційним значенням”)
* Випадки використання можуть мати зв’язки з іншими випадками використання.

**Діаграма послідовності**

***Діаграма послідовностей*** відображає взаємодію об’єктів у динаміці. *В UML взаємодія об’єктів розглядається як обмін між ними інформацією у вигляді повідомлень. Діаграма послідовностей потрібно використовувати для уточнення діаграм прецедентів, надаючи більш детальний опис логіки сценаріїв використання.* Ця обставина робить їх хорошим засобом документування проекту з точки зору сценаріїв використання, оскільки діаграми послідовностей зазвичай містять об’єкти, які взаємодіють у рамках сценарію, повідомлення, якими вони обмінюються та пов’язані з повідомленнями результати, що повертаються.

***Повідомлення*** – сигнал до виконання певної операція, яка повинна бути здійснена об'єктом, що прийняв повідомлення.

Фактично, діаграма послідовності – це запис протоколу конкретного сеансу роботи системи (або фрагменту такого протоколу). Важливим аспектом діаграм послідовності є явне відображення плину часу, оскільки на ній має значення не тільки наявність графічних зв’язків між елементами, але й взаємне розташування елементів на діаграмі відповідно до явно не відображеної вісі часу, яка може бути направлена згори донизу, і пізніше направлене повідомлення розташоване нижче.

Нижче представлена діаграма послідовності із використанням стереотипних значень (рис. 5). На діаграмі проілюстровано скорочену варіацію на тему казки про «Ріпку». Зауважимо, що на початку ріпки в системі не існувало, даний об’єкт був посаджений (створений) дідом, що відображається стереотипом «create». Після невдалою спробі витягти ріпку, дід звернувся до бабці (стереотип «send»), яка в свою чергу покликала мишу(стереотип «send»). Знищення ріпки відображено стереотипом «destroy».

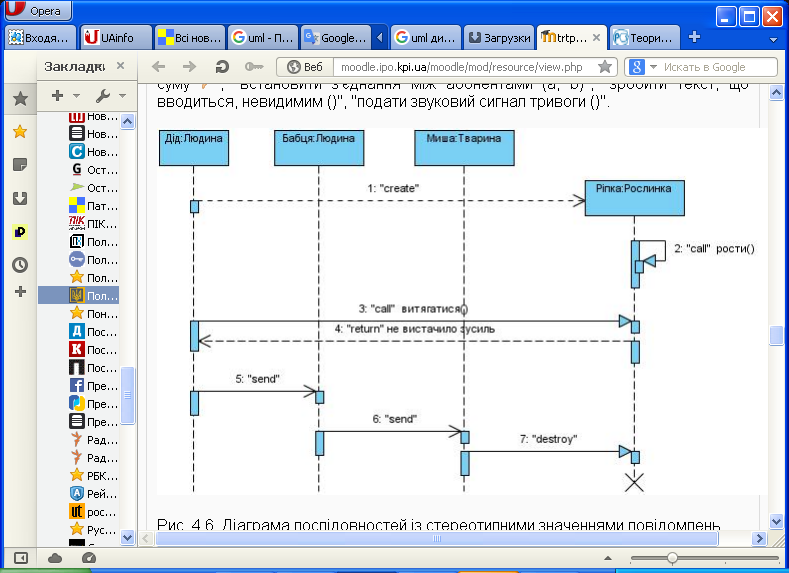


Рисунок 5 - Діаграма послідовностей із стереотипними значеннями повідомлень

На діаграмі послідовності зображуються об'єкти, які безпосередньо беруть участь у взаємодії, при цьому ніякі статичні зв'язки з іншими об'єктами не візуалізуються. Для діаграми послідовності ключовим моментом є саме динаміка взаємодії об'єктів у часі. При цьому діаграма послідовності має як би два виміри. Один – простягається зліва направо у вигляді вертикальних ліній, кожна з яких зображує лінію життя окремого об'єкту, що бере участь у взаємодії. Другий вимір діаграми послідовності - вертикальна тимчасова вісь, спрямована зверху вниз.

***Фокус керування (focus of control)*** - спеціальний символ на діаграмі послідовності, що вказує період часу, протягом якого об'єкт виконує деяку дію, перебуваючи в активному стані.

Фокус керування зображується у формі витягнутого вузького прямокутника (об'єкт Клієнтська сесія на рис 6.), верхня сторона якого позначає початок одержання фокуса управління об'єкта (початок активності), а її нижня сторона - закінчення фокуса керування (закінчення активності). Цей прямокутник розташовується нижче позначення відповідного об'єкта й може заміняти його лінію життя (об'єкт Користувач Петро на рис. 6), якщо на всьому її протязі він активний.

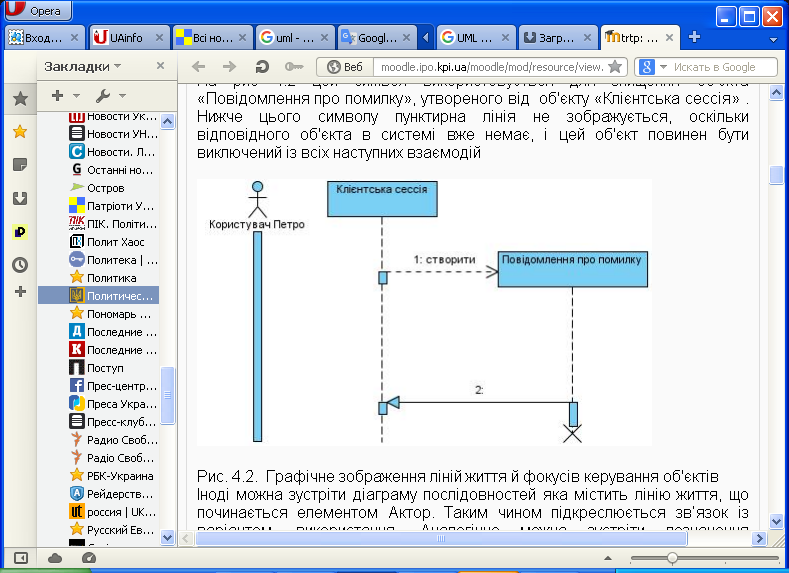


Рисунок 6 - Графічне зображення ліній життя й фокусів керування об'єктів