

Затверджено науково-методичною  
радою ЖДТУ  
протокол від «\_\_» \_\_\_\_\_ 20\_\_ р. №\_\_

Методичні рекомендації  
для виконання курсових робіт  
з навчальної дисципліни  
**«МОДЕЛЮВАННЯ ТА АНАЛІЗ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ»**

для студентів освітнього рівня «БАКАЛАВР»  
напряму 6.050103 «Програмна інженерія»  
освітньо-професійна програма «Програмна інженерія»  
факультет інформаційно-комп'ютерних технологій  
кафедра інженерії програмного забезпечення

Розглянуто і рекомендовано  
на засіданні кафедри інженерії  
програмного забезпечення  
протокол від «\_\_» \_\_\_\_\_ 20\_\_ р.  
№ \_\_\_\_

Завідувач інженерії програмного  
забезпечення  
\_\_\_\_\_ А.В. Панішев

Розробники: к.т.н., доцент кафедри інженерії програмного забезпечення І.І. Сугоняк, ст.  
викладач кафедри інженерії програмного забезпечення Власенко О.В., к.пед.н., доцент кафедри  
інженерії програмного забезпечення Ковальчук В.Н.

Житомир, 2018

## ЗМІСТ

ВСТУП .....	4
1. ОРІЄНТОВАНА ТЕМАТИКА КУРСОВИХ РОБІТ .....	5
2. ГРАФІК ВИКОНАННЯ КУРСОВОЇ РОБОТИ.....	6
3. КРИТЕРІЇ ОЦІНЮВАННЯ КУРСОВОЇ РОБОТИ .....	7
4. СТРУКТУРА ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ ПО КУРСОВІЙ РОБОТІ .....	8
5. ВИМОГИ ПО ОФОРМЛЕННЮ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ .....	10
РЕКОМЕНДОВАНА ЛІТЕРАТУРА .....	15
ДОДАТКИ .....	16

## ВСТУП

Важливим етапом вивчення дисципліни „Моделювання та аналіз програмного забезпечення” є написання курсової роботи. Задачами курсової роботи є:

- систематизація і закріплення теоретичних та практичних фахових знань, виявлення уміння студента застосовувати ці знання при вирішенні конкретних наукових, технічних, економічних і виробничих задач;
- перевірка уміння студента самостійно освоювати та використовувати сучасні інформаційні технології, програмно-апаратні засоби обчислювальної техніки;
- розвинення у студента навичок ведення самостійного науковопрактичного пошуку, оволодіння методикою дослідження й експериментування при вирішенні проблем і питань, поставлених на курсове проектування;
- закріплення знань і навичок виконання графічних робіт та інших конструкторських документів у відповідності до вимог і правил, встановлених державними стандартами, Єдиною системою конструкторської документації (ЄСКД), Єдиною системою проектної документації (ЄСПД), іншими чинними нормативно-технічними документами;

Головна мета курсової роботи – визначення рівня теоретичної та практичної підготовки студента з курсу „Моделювання та аналіз програмного забезпечення”.

Курсова робота сприяє розвитку у студента творчої ініціативи і самостійності в проведенні аналізу, добору й обґрунтування найбільш раціональних інженерних рішень.

Курсова робота надає студентіві таких навичок виконання виробничих завдань, які допоможуть йому швидко адаптуватися до умов праці у професійному колективі.

Керівнику курсової роботи необхідно забезпечити оптимальні умови для плідної самостійної роботи студентів. Цій меті повинні сприяти продумана методика керівництва і консультування, допомога в пошуку методичної та технічної документації, науково-технічної літератури, а також систематичний контроль за виконанням курсового проекту.

## 1. Орієнтована тематика курсових робіт<sup>1</sup>

№ з.п.	Тематика
1.	Проектування системи керування режимом роботи ПК
2.	Проектування архітектури інтернет-форуму
3.	Модель структури програмного забезпечення інтернет-форуму
4.	Моделювання програмного забезпечення станції техобслуговування
5.	Проектування навчального програмний комплекс «Адресація в IPмережах»
6.	Модель програмного генератора повідомлень каналного рівня мереж Ethernet
7.	Проектування архітектури інтернет-сайту «Оцінки викладача»
8.	Моделювання структури сайт для інтернет-магазину автозапчастин
9.	Проект модуля фінансової звітності та роботи з клієнтами для система автоматизації прокатного бізнесу
10.	Моделювання структури програмного комплексу інтернет-магазину цифрової техніки
11.	Моделювання архітектури сайт-каталогу систем відеонагляду "Bielion"
12.	Модель інформаційної он-лайн бібліотеки
13.	Модель сайту подачі безкоштовних оголошень «Besplatka.ua»
14.	Проектування навчальної програми «Симплекс-метод»
15.	Проектування навчальної програми «Метод золотого перерізу»
16.	Модель структури веб-сайту модмейкерського ресурсу
17.	Проект системи управління версіями файлів на сервері
18.	Модель підсистеми обліку рейсів на автопідприємстві
19.	Розробка архітектури для веб-орієнтована підсистема управління робочих завдань працівників
20.	Проектування програмного комплексу навчально-тренувального практикуму з програмування для дітей

<sup>1</sup> За бажанням студент може визначити собі іншу тему, та узгодити це з викладачем. Рекомендовано обирати тему у відповідності до теми випускної роботи

## 2. ГРАФІК ВИКОНАННЯ КУРСОВОЇ РОБОТИ

№ п/п	Зміст проведених робіт	Термін виконання у % до загального терміну виконання	Форма звітності
1.	Вибір тематичного напрямку та узгодження теми курсової роботи	10%	Закріплення теми курсової роботи за студентом
2.	Розробка завдання, визначення методів та засобів реалізації поставленої задачі	15%	Роздрукована постановка завдання за підписом викладача
3.	Аналіз теоретичних матеріалів за напрямком дослідження, вивчення предметної області	10%	Оформлення п.1,2 курсової роботи в електронному вигляді, оформлення в ел. вигляді списку використаної літератури
4.	Розробка концептуальної та логічної моделі програмної системи	35%	Реалізація та специфікація моделей, опис побудованих діаграм
5.	Розробка фізичної моделі та реалізація прототипу програмного забезпечення	15%	Розробка тестового прикладу та опис порядку дій користувача в ел. вигляді
6.	Оформлення пояснювальної записки	15%	Роздрукована пояснювальна записка та готовий програмний продукт на електронному носії інформації (дискета, CD-R, RW, DVD-R, RW)

3.

### КРИТЕРІЇ ОЦІНЮВАННЯ КУРСОВОЇ РОБОТИ

Курсова робота оцінюється комплексним рейтинговим показником, який враховує наступні складові:

Критерій	Максимальна кількість балів	Примітка
Своєчасність затвердження теми та завдання на курсову роботу	10	Бали не нараховуються у випадку несвоєчасного затвердження теми та завдання

Якість оформлення пояснювальної записки	20	Бали може бути знижено за порушення вимог до оформлення роботи, невідповідність структури курсової визначеним вимогам та ін.
Алгоритмічна, структурна та функціональна складність розробленої моделі та глибина опрацювання елементів	30	Оцінюється якість та коректність побудованої математичної моделі, правильність оформлення алгоритмів та складність структури
Якість підготовленої доповіді та рівень захисту курсової роботи	10	Бали може бути знижено за відсутність презентації (5 балів), некоректні відповіді на питання 10 балів

Сумарний показник переводиться у оцінку за затвердженою шкалою оцінювання.

#### **4. Структура пояснювальної записки по курсовій роботі**

Курсова робота складається з наступних розділів:

**ВСТУП**

**1. АНАЛІЗ ВИМОГ КОРИСТУВАЧНА ТА КОНЦЕПТУАЛЬНЕ МОДЕЛЮВАННЯ**

1.1. Технічне завдання на розробку системи 1.2. Обґрунтування вибору засобів моделювання.

1.3. Аналіз вимог до програмного продукту

**2. РОЗРОБКА МОДЕЛІ ПРОГРАМНОГО КОМПЛЕКСУ НА ЛОГІЧНОМУ РІВНІ**

2.1. Алгоритм роботи та стани програмної системи

2.2. Об'єктно-орієнтована модель системи

2.3. Комунікації і послідовність взаємодії об'єктів системи

**3. ФІЗИКА МОДЕЛЬ ТА ПРОТОТИП ПРОГРАМНОГО КОМПЛЕКСУ**

3.1. Взаємодія компонентів системи

3.2. Архітектура програмного комплексу та його розгортання

3.3. Генерування програмного коду для прототипу програмного комплексу

### 3.4. Моделі тестування системи

#### ВИСНОВКИ

#### ЛІТЕРАТУРА

#### ДОДАТКИ

Приклад змісту курсової роботи наведено в додатку А.

У Вступі зазначається мета, завдання, предмет, об'єкт та методи дослідження курсової роботи. Орієнтований обсяг 1 стр.

Метою курсової роботи є дослідження особливостей моделювання та аналізу програмних комплексів за визначеним темою курсової роботи напрямком з використанням CASE-технологій.

Завданням на курсову роботу є:

- аналіз теоретичних засад моделювання програмного забезпечення;
- аналіз та опис вимог користування;
- методи модулювання функцій та поведінки системи;
- проектування об'єктної структури системи; □ фізичне моделювання програмних комплексів; □ кодогенерація із моделей.

Предметом дослідження є можливості застосування CASE-засобів проектування програмного забезпечення.

Об'єктом дослідження є методи та засоби проектування програмного забезпечення та уніфікація процесу проектування.

В процесі роботи над курсовим проектом студент може використовувати монографічні, аналітичні, математичні, графічні методи, методи об'єктоорієнтованого проектування та програмування та інші методи дослідження.

Пункт 1.1 містить технічне завдання на дипломне проектування. Орієнтований обсяг до 3 стр.

Пункт 1.2 містить порівняльний аналіз не менше трьох засобів моделювання програмного забезпечення на мові UML (за варіантом) + засіб в якому буде реалізовано.

Пункт 1.3 містить опис вимог до програмного забезпечення. Побудовану діаграму Use Cases. Каталог вимог та глосарій наводяться в додатках орієнтований обсяг до 7 стр.

Пункт 2.1 Містить діаграми активності та станів, що ілюструють поведінку системи і послідовність виконання операцій із описом.

Пункт 2.2 Містить діаграми взаємодії компонентів, діаграму кооперації та діаграми послідовності для варіантів використання системи.



Пункт 2.3 Містить об'єктна-орієнтовану модель програмного комплексу (діаграму класів) виконану із використанням патернів проектування. Орієнтований обсяг 7 стор.

Пункт 3.1. Містить взаємодії компонентів систем зроблений із використанням діаграми компонентів. Орієнтований обсяг до 7 стор.

Пункт 3.2. Містить опис архітектурною будови системи на основі діаграми розгортання. Орієнтований обсяг 7 стор.

Пункт 3.3. Містить порядок кодогенерації моделей програмного забезпечення та опис прототипу системи.

У висновках наводяться стислий перелік результати дослідження можливостей спроектованого ПЗ тощо.

Література містить список всіх літературних джерел, кількість джерел не повинна бути менше за 25.

У додатках наводяться Каталог вимог, Глосарій, Згенерований код, код реалізованого інтерфейсу та інші елементи за бажанням студента.

**Обов'язковим є реалізація робочого прототипу програмної системи.**

## **5. Вимоги по оформленню пояснювальної записки**

Пояснювальну записку до курсового проекту оформляють у відповідності з вимогами ГОСТ 2.105-95 ЄСКД “Загальні вимоги до текстових документів”, ГОСТ 19.105-78 ЄСПД “Загальні вимоги до програмних документів” та ГОСТ 19.404-79 ЄСПД “Пояснювальна записка. Вимоги до змісту та оформленню.”

Варто звернути увагу на окремі особливості оформлення ПЗ. Пояснювальна записка подається у друкованому вигляді та ел. вигляді. **Обсяг пояснювальної записки** – 40-50 сторінок друкованого тексту (враховуючи таблиці, схеми, графіки, діаграми та ін.)

**Оформлення ОСНОВНОГО ТЕКСТУ** через 1,5 інтервали, на одній стороні листа папера формату А4 (210x297 мм). Відступ зліва 1,27. Шрифт Times New Roman, кегль 14. Кожен розділ ПЗ починають з нового листа.

**НУМЕРАЦІЯ СТОРІНОК** ПЗ повинна бути наскрізною, першою сторінкою є титульний лист, друга та третя - завдання на курсову роботу, четверта - зміст і т.п. У нумерацію сторінок ПЗ включають графіки, таблиці, схеми, креслення й інші матеріали, виконані на окремих аркушах і вшиті в загальну підшивку.

**РОЗДІЛИ І ПІДРОЗДІЛИ** повинні мати найменування у вигляді заголовків. Найменування розділів записують прописними буквами не підкреслюючи, а підрозділів - рядковими (крім першої прописної). Переноси слів

у заголовках не допускаються. Крапку наприкінці заголовків не ставлять. Якщо заголовок складається з декількох речень, їх розділяють крапкою. Усі складові частини ПЗ (розділи, підрозділи і пункти) повинні мати порядкові номери, позначені арабськими цифрами з крапкою, наприклад, 2.2 - (перший підрозділ другого розділу).

Номер розділу і підрозділу ставлять перед найменуванням і відокремлюють його крапкою.

Розділи вирівнюються посередині. Підрозділи мають виключку двосторонню (вирівнювання по ширині), з відступом 1,25. Відстань між заголовком і наступним текстом – 24 пт., відстань між заголовком підрозділу і останнім рядком попереднього тексту – 36 пт..

**ВИКЛАД ТЕКСТУ** ПЗ рекомендується вести від третьої особи: “як показують наші розрахунки”; “ми вважаємо”; “наше рішення” тощо, або використовувати безособове викладення, „підсумовуючи викладене вище”, „в роботі наведено”, „у курсовій визначено”. В тексті ПЗ потрібно дотримуватися єдиної термінології. Не варто зловживати іноземними словами, особливо в тих випадках, коли знаходяться рівнозначні українські слова (терміни). Найменування фірм, заводів, організацій не відмінюються, їх треба включати у лапки.

**ФОРМУЛИ** нумерують арабськими цифрами в межах розділу. Номер формули складається з номера розділу і порядкового номера формули в розділі, розділених крапкою. Номер указують із правої сторони листа на рівні формули в круглих дужках, наприклад: (3.15) (п'ятнадцята формула третього розділу).

Значення символів і коефіцієнтів, що входять у формулу, приводять під формулою. Після формули пишуть слово "де" без двокрапки після нього, за ним - символи і числові коефіцієнти розшифровують у такій послідовності, у якій вони приведені у формулі. Значення кожного символу приводять з нового рядка. Наприклад:

$$ЗП \square P * Vnd * \frac{Om}{ds * Dr}, \quad (2.7)$$

де  $P$  – число працівників певної кваліфікації;

$Vnd$  – час участі працівників певної кваліфікації в даному виді робіт;

$Om$  – місячний оклад, визначуваний відповідно до категорії і тарифного розряду, згідно таблиці 1;

$ds$  – тривалість зміни (8 годин);

$Dr$  – середнє число робочих днів (26).

**Всі ІЛЮСТРАЦІЇ в ПЗ** (креслення, схеми, фотографії, графіки) називають рисунками. Вони повинні мати нумерацію в межах розділів (наприклад: Рис. 1.5, Рис. 2.3 і т.д.).

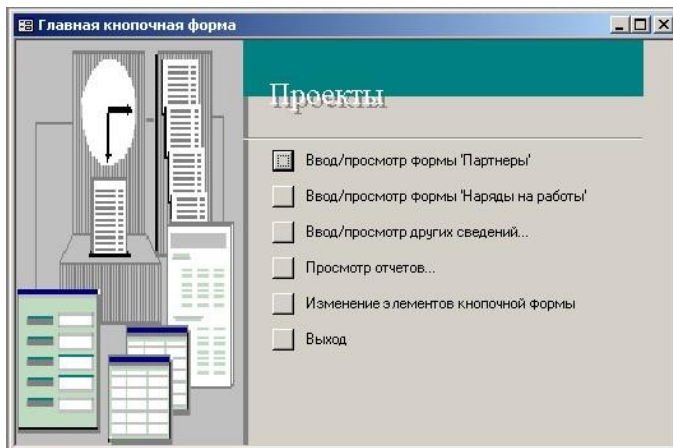


Рис. 3.1. Головна кнопкова форма

Кожен рисунок повинний мати смисловий заголовок, який пишуть під рисунком, указують його номер.

Посилання на рисунки вказують у круглих дужках, наприклад: (рис.1.5). При другому і наступному посиланнях на той самий рисунок додають олово "дивися", наприклад (див. рис.1.5).

Результати розрахунків і деякі розрахунки зводять у таблиці. У відповідності до ГОСТ 2.105-95 до кожної таблиці дають точний і короткий заголовок, що відбиває її зміст і ознаку, що відрізняє її від інших таблиць. Розміщують заголовок над таблицею, підкреслювати його не слід. Заголовок пишуть у називному відмінку однини. Крапку після нього не ставлять, заголовки підлеглих ступенів пишуть з малої літери.

**Всі ТАБЛИЦІ** нумерують в межах розділу. Над тематичним заголовком, праворуч з прописної букви цілком пишуть слово "Таблиця" і проставляють її порядковий номер.

Номер таблиці складається з номера розділу і порядкового номера таблиці, розділених крапкою, наприклад; "Таблиця 2.2" (друга таблиця другого розділу).

При посиланні на таблицю в тексті пишеться табл. I її порядковий номер (наприклад табл.2.3., див.табл.2.3) слово "таблиця" пишеться повністю лише при відсутності номера.

Над продовженням таблиці на новій сторінці пишуть "Продовження табл. 2.2", якщо продовження займає більше однієї сторінки та перед останнім фрагментом таблиці пишуть "Закінчення табл. 2.2" (без лапок), тематичний заголовок не повторюють; всі частини розбитої таблиці починають не передруком заголовку, а рядком з нумерацією, що замінює назви стовпців. Вертикальні стовпці нумерують тільки в тих випадках, коли в тексті на них даються посилання або коли таблиця продовжується на наступній сторінці.

## Програмне забезпечення інформаційної системи ТОВ „ІТС”

N	Найменування	Тип
1.	2.	3. <sup>2</sup>
1.	Операційна система	OEM Windows XP Home Rus
2.	Текстовий редактор	OEM MS Office XP
3.	Антивірусне забезпечення	Антивірус UNA PRO WS BOX

У таблиці дотримують рівновагу її частин: ліва бокова частина не повинна займати більш третини її формату, а висота заголовку – не більше третини висоти таблиці.

Одиниці вимірювань вказують у заголовках. Цифри в стовпцях розташовують так, щоб одиниці знаходилися під одиницями, десятки під десятками і т.д. На цифрові групи числа (починаючи з п’яти знаків) розбивають справа наліво по три цифри проміжками без крапки.

Бажано всі ілюстративні матеріали розміщувати відразу після посилання на них. Розташовувати таблиці і графічний матеріал слід так, щоб їх можна було читати без повороту аркуша. Якщо таке розміщення неможливе, то таблиці і ілюстрації розташовують так, щоб для їхнього читання треба було повернути аркуш по годинниковій стрілці.

Усі метричні величини приводять тільки в одиницях СІ.

Усі терміни та визначення приводять у відповідності до ГОСТ 19.781-90 ЄСПД “Забезпечення систем обробки інформації: програми, терміни та визначення.”

Схеми алгоритмів розробляють у відповідності з ГОСТ 19.005-85 ЄСПД “Р-схеми алгоритмів та програм. Позначення умовні графічні та правила виконання”.

Наведений текст програми треба виконувати у відповідності до ГОСТ 19.401-78 ЄСПД “Текст програми. Вимоги до змісту та оформленню”.

Опис та призначення окремих програм треба виконувати у відповідності до ГОСТ 19.402-78 ЄСПД “Опис програми”.

При проведенні розрахунків на ЕОМ у ПЗ необхідно привести використану для розрахунків формулу, схему і програму розрахунку, дати результати у виді даних, видрукованих цифро-друкуючим пристроєм ЕОМ.

Усі цитати, а також узяті з друку дані, повинні мати посилання на першоджерела.

<sup>2</sup> рядок нумерації стовпців вставляється лише за умови, що таблиця займає більше 1 сторінки

**ДОДАТКИ.** Після останньої сторінки списку джерел перед додатками необхідно помістити чистий лист паперу, на якому посередині великими буквами написати "ДОДАТКИ". Номер сторінки на цьому аркуші не ставиться. У змісті роботи можна вказувати як кожне з наявних у роботі додатків з відповідними їм сторінками, так і перший лист додатків.

Додатки позначаються заголовними буквами за абеткою, за винятком букв Є, С, З, Ї, Е, І, Й, О, Ч, Ь. При необхідності текст додатку може бути розділений на розділи, у цьому випадку номер розділу додатку складається з букви додатка і поточного номера розділу додатка, розділених крапкою.

Додаток повинен мати узагальнюючий заголовок, надрукований угорі малими літерами з першої прописної симетрично щодо тексту сторінки. Праворуч над заголовком, вирівняним посередині малими літерами з першої прописної повинне бути написане слово «Додаток \_» і прописна буква, що позначає додаток.

Додатки повинні мати загальну з іншою частиною роботи наскрізну нумерацію сторінок.

Наявні в тексті додатка ілюстрації, таблиці, формули варто нумерувати в межах кожного додатка. Наприклад, рисунок А.1, таблиця Б.2, формула Д.3.

Якщо в роботі як додатки використовуються документи (оригінали чи їхні копії), що мають самостійне значення й оформляються відповідно з вимогами до документів даного виду, їхні копії містяться в роботі. На першій сторінці документа праворуч (якщо є місце) пишуть:

«Додаток \_» і на другому рядку – його найменування. Якщо місця немає, то перед додатком поміщають чистий лист паперу, на якому посередині листа пишуть номер і назву додатка. Сторінки документа нумеруються наскрізною нумерацією, номер проставляється в правому нижньому куті без крапки наприкінці.

**ПОСИЛАННЯ НА ЛІТЕРАТУРНІ ДЖЕРЕЛА** приводяться в тексті у квадратних дужках. У дужках ставлять порядковий номер джерела, приведенного в описку використаної літератури, номер тому, якщо необхідно, - сторінку, наприклад: [3], [8, т. 2, с. 42], [15, с. 553].

Список літератури рекомендується складати або в алфавітному порядку або у порядку згадування в тексті. Зразок оформлення літератури наведено в додатку Г

Вимоги щодо оформлення літературних джерел за IEEE див. за посиланнями [https://ztu.edu.ua/ua/science/files/1\\_IEEE-style.pdf](https://ztu.edu.ua/ua/science/files/1_IEEE-style.pdf) та <http://libguides.murdoch.edu.au/IEEE>.

Автоматичне оформлення списку літературних джерел <http://vak.in.ua/>. Для оформлення у IEEE style - <http://www.citethisforme.com/citation-generator>.

Зразки титульного аркушу та завдання курсову роботу наведено в додатку Д.

### Рекомендована література

1. Леоненков А. В. Самоучитель UML / А. В. Леоненков. – СПб. : БХВ-Петербург, 2004. – 432 с.
2. Г. Буч, Дж. Рамбо, А. Джекобсон Язык UML. Руководство пользователя.: Пер. с англ. – М.: ДМК, 2000. – 432с.
3. Кватрани Т. Визуальное моделирование с помощью Rational Rose 2002 и UML. – М.: Вильямс, 2003. – 192 с.
4. Г. Буч Объектно-ориентированный анализ и проектирование с примерами приложений на C++. 2-е изд.: Пер. с англ. – М.: Издательство Бином, СПб.: Невский диалект, 1999.
5. К. Ларман, Применение UML 2.0 и шаблонов проектирования (3-е издание) Вильямс. 2006. – 496 с.
6. Брукс Ф. Мифический человеко-месяц или как создаются программные системы. — Пер. с англ. — СПб.: Символ-Плюс, 2001. — 304 с.
7. Орлов С. Технологии разработки программного обеспечения: Учебник. — СПб.: Питер, 2002. — 464 с.: ил. 8. Коберн А. - Современные методы описания функциональных требований к системам. Лори, 2011 г. – 288 с.
9. Гамма, Э. Приемы объектно - ориентированного проектирования. Паттерны проектирования. СПб. Питер, 2006. - 366 с.

## **ДОДАТКИ**



## ДОДАТОК А

## Приклад змісту курсової роботи

## ВСТУП

1. АНАЛІЗ ВИМОГ КОРИСТУВАЧНА ТА КОНЦЕПТУАЛЬНЕ  
МОДЕЛЮВАННЯ WEB-ОРІЄНТОВАНОЇ СИСТЕМИ БРОНЮВАННЯ  
АВІОКВІТКІВ

1.1. Постановка завдання

1.2. Методи та засоби моделювання предметної області

1.3. Аналіз вимог до системи бронювання квитків з боку авіакомпанії

2. РОЗРОБКА КОНЦЕПТУАЛЬНОЇ МОДЕЛІ WEB-ОРІЄНТОВАНОЇ  
СИСТЕМИ БРОНЮВАННЯ АВІОКВІТКІВ

2.1. Алгоритм роботи та стани системи бронювання

2.2. Об'єктно-орієнтована модель системи бронювання

2.3. Комунікації і послідовність взаємодії об'єктів системи

3. ФІЗИКА МОДЕЛЬ ТА ПРОТОТИП WEB-ОРІЄНТОВАНОЇ СИСТЕМИ  
БРОНЮВАННЯ АВІОКВІТКІВ

3.1. Структура та взаємодія компонентів системи

3.2. Архітектура програмного комплексу та його розгортання

3.3. Генерування програмного коду для прототипу програмного комплексу

3.4. Моделі тестування системи

## ВИСНОВКИ

## ЛІТЕРАТУРА

## ДОДАТКИ

## ДОДАТОК Б

## Постановка завдання

Назва системи – Web-орієнтована система бронювання квитків «Aviosale».

Призначення системи – система впроваджується в структурі оператора авіоперельотів для підтримки дистанційного бронювання та продажу квитків через глобальну мережу Інтернет.

Мета. Впровадження Web-орієнтована система бронювання квитків дозволить забезпечити максимальний доступ до послуг авіаперевізників, спростити процедуру бронювання та продажу авіаквитків, дозволить користувачам користуватись механізмами багатокритеріального пошуку та відбору квитків та умов продажу, спростить формування звітності та оновлення актуальної інформації про стан продажу авіобілетів.

Вхідні дані. Відомості про дзвінки за визначений період, тарифні плани тощо.

Технічні вимоги: апаратне та системне забезпечення: РС 486 та вище, операційна система Windows-98, 2000, NT, XP Функціональні характеристики системи:

- Надати постійний доступ співробітникам оператора CDMA до оперативної інформації про стан рахунку розрахунків окремих клієнтів
- Вхідні дані формуються на базі інформації системи зв'язку щодо переговорів.
- Число спеціалізованих робочих місць – 20.
- Максимальна кількість клієнтів за місяць -1000000.
- орієнтовна кількість записів, що зберігаються в базі даних – 100000000.
- кількість споживачів інформації – 80-100.
- Перелік звітних форм – рахунок за поточний період:
  - Дані про динаміку витрат за 10 попередніх періодів; ○ роздруківка дзвінків за період.
- Перелік запитів, що повинні виконуватися у режимі on-line: 1) Поточна заборгованість за зазначений період часу; 2) Показники стану рахунку та тарифного плану.

Якісні характеристики роботи системи:

Таблиця 2.1

Опис	Потрібно	Допустимо	Примітка
Часу реакції	3 секунди	3-7 секунд	
Часи роботи	цілодобово щоденно		
Доступність	95%	90-95%	

Результат науково-дослідної та дослідно-конструкторської роботи (НДДКР) - розробка всіх видів забезпечення інформаційної системі оператора CDMA-зв'язку.

## ДОДАТОК Г

### Приклади оформлення літературних джерел **Державні**

#### **стандарти:**

1. Інформаційні технології. Процеси життєвого циклу програмного забезпечення (ISO/IEC 12207:1995): ДСТУ 3918-1999. – [Чинний від 200001-

- 01]. – К.: Держстандарт України, 2000. – 50 с. – (Національний стандарт України).
2. Інформаційні технології. Основні напрямки оцінювання та відбору CASE-інструментів (ISO/IEC 14102:1995) – ДСТУ 3919-1999 [Чинний від 2000-0101]. – К.: Держстандарт України, 2000. – 470 с. – (Національний стандарт України).
- Книги:**
3. Вендров А. М. CASE-технологии – современные методы и средства проектирования информационных систем / Вендров А. М. – М.: Финансы и статистика, 1998 – 171 с.
4. Кальянов Г. Н. CASE. Структурный системный анализ (автоматизация и применение) / Кальянов Г. Н. – М.: Лори, 1996. – 242 с.
5. Марка Д., Методология структурного анализа и проектирования / Д. Марка , К. МакГоуэн. – М.: МетаТехнология, 1993. – 240 с.
6. Черемных С. В. Структурный анализ систем: IDEF-технологии / Черемных С. В., Семенов И. О., Ручки В. С. – М.: Финансы и статистика, 2003. – 208 с.
7. Кватрани Терри. Визуальное моделирование с помощью Rational Rose 2002 и UML / Кватрани Т.; пер. с англ. – М.: Издательский дом „Вильямс”, 2003. – 192 с.
8. Новоженев Ю. В. Объектно-ориентированные технологии разработки сложных программных систем / Ю. В. Новоженев. – М.: Наука, 1996 – 356 с.
9. Фаулер М, Скотт К. UML в кратком изложении. Применение стандартного языка объектного моделирования / М. Фаулер, К. Скотт. – М: «Мир», 1999 – 191 с.
10. Дейт К. Введение в системы баз данных / Дейт К. – [8-е изд. ] – СПб.: Вильямс, 2005 – 1328 с.
11. Праг К. Н. Access 2002 / К. Н. Праг, М. Р. Ирвин; пер. с англ. – М. Диалектика, 2003. – 1216 с. – (Серия “Библия пользователя”).
12. Фаронов В. Программирование баз данных Delphi 7. Учебный курс / Фаронов В.В. – СПб: Питер, 2005. – 459 с.
13. Чкалов А. П. Базы данных: от проектирования до разработки приложений / Чкалов А. П. – СПб.: БХВ-Петербург, 2003. – 384 с.
14. Искусственный интеллект: справочник в 3 кн. / [под ред. Э. В. Попова, Д. А. Поспелова, В. Н. Захарова, В. Ф. Хорошевского] – М.: . - Радио и связь, 1990. - Кн. 2: Модели и методы / [под ред. Д. А. Поспелова]. – 1990. – 304 с.

Кн. 3: Программные и аппаратные средства: справочник / [под ред. В. Н. Захарова, В. Ф. Хорошевского] – 1990. – 368 с.

**Статті:**

15. Глущенко В. В. Исследование множеств и разработка одного типа представления и метода планирования испытания сложных технических систем / В. В. Глущенко // Кибирнетика и системный анализ. – 1992. – № 2. – С. 27-28
16. Сугоняк І. І. Структура та сценарії роботи системи підтримки прийняття рішень з оптимального керування інноваційними процесами підприємств / І. І. Сугоняк // Наукові проблеми модернізації та застосування інформаційних систем: XVII наук. – тех. конф: 24–25 квіт. 2008 р.: тези доп. Ч. I – Житомир, 2008. – С. 86 - 87. **Електроні ресурси:**
17. Кириллов В. В. Структуризованный язык запросов (SQL): учебн. пособ.: [Электронный ресурс] / В. В. Кириллов, Г. Ю. Громов. – СПб: Санкт<http://www.ifmo.ru/Петербург.госуд.техн.универ.>, [каф. выч. техники](#), 1998. – Режим доступа к пособию: [http://www.citforum.ru/database/sql\\_kg/](http://www.citforum.ru/database/sql_kg/).
18. Иванова В. Б. Алгоритм принятия решения о внедрении инновационного проекта [Электронный ресурс] / В. Б. Иванова // Вісн. міжнар. слов`ян. унту. Серія: Екон. науки. – 2006. – 9, № 1. – С. 20-24 – Режим доступу до статті: <http://www.nbu.gov.ua/articles/2006/06ivboip.html>

**Приклади оформлення літературних джерел за IEEE**

- [1] М. Гленфорд, Т. Баджетт, С. Кори, Искусство тестирования программ, 3-е изд. Москва: Дилектика, 2012, 345 с.
- [2] С. Хольцнер, PHP в примерах. М.: ООО«Бином-Пресс», 2007 г., 352 с
- [3] Д. Марка, К. МакГоуэн, Методология структурного анализа и проектирования. М.: МетаТехнология, 1993, 240 с.
- [4] М. Фаулер, К. Скотт. UML в кратком изложении. Применение стандартного языка объектного моделирования. М: «Мир», 1999, 191 с.
- [5] К. Канер, Дж. Фолк, К.Е. Нгуен, Тестирование программного обеспечения.  
Фундаментальные концепции менеджмента бизнес-приложений. Киев: ДиаСофт, 2001, 544 с.
- [6] А. Гончаров, Самоучитель HTML. СПб.: Питер, 2002, 240 с.
- [7] Д. Котеров, Самоучитель PHP4. СПб.: БХВ, Петербург, 2003, 576 с.
- [8] Э. Харрис, PHP/MySQL для начинающих, пер. с англ. М.: КУДИЦ–ОБРАЗ, 2005, 384 с.
- [9] Л. Криспин, Д. Грегори, Гибкое тестирование: практическое руководство для тестировщиков ПО и гибких команд, М.: «Вильямс», 2010, 464 с.
- [10] С.А. Мартишин, В.А. Симонов, М.В. Храпченко, Проектирование и реализация баз данных в СУБД MySQL с использованием MySQL Workbench, Учебное пособие. М.: ИНФРА-М, 2012, 160 с.
- [11] С.В. Сеницын, Н.Ю. Налютин, Верификация программного обеспечения. М.: БИНОМ, 2008, 368 с.
- [12] Довідниково-інформаційний портал htmlbook [Електроний ресурс]. Доступ за посиланням: <http://htmlbook.ru/>. [Дата звернення 22.05.2017 р.]
- [13] PHPFAQ [Електроний ресурс]. Доступ за посиланням: <http://phpfaq.ru/pdo>. [Дата звернення 22.05.2017 р.]
- [14] E.Wurzer, Why you Should be using PHP's PDO for Database Access, 2012. [Електроний ресурс]. Доступ за посиланням: <https://code.tutsplus.com/tutorials/why-you-should-be-using-phps-pdo-for->

[database-access--net-12059using-phps-pdo-for-database-access--net-12059](http://www.citforum.ru/database/sql_kg/).

[Дата

звернення 22.05.2017 р.]

- [15] В.В. Кириллов, Г.Ю. Громов, Структуризованный язык запросов (SQL): учебн. пособ.. СПб: Санкт-Петерб. госуд. техн. универ., 1998. [Електроний ресурс]. Доступ за посиланням: [http://www.citforum.ru/database/sql\\_kg/](http://www.citforum.ru/database/sql_kg/). [Дата звернення 22.05.2017 р.]

ДОДАТОК Д

Зразок титульного аркушу

Міністерство освіти і науки України

Житомирський державний технологічний університет

Кафедра інженерії програмного забезпечення

## КУРСОВА РОБОТА

з дисципліни «Моделювання та аналіз»

на тему: \_\_\_\_\_

Студента (ки) \_\_\_\_\_ курсу \_\_\_\_\_ групи напрям  
підготовки 6.05010301 «Програмна інженерія»

\_\_\_\_\_  
(прізвище та ініціали)

Керівник:

Національна шкала \_\_\_\_\_

Кількість балів: \_\_\_\_\_ Оцінка: ECTS \_\_\_\_\_

Члени комісії: \_\_\_\_\_ О.В. Власенко

\_\_\_\_\_ С.М.Кравченко

\_\_\_\_\_ В.Н. Ковальчук

м. Житомир – 2017 рік

## ДОДАТОК Ж

UML 2.0 прийнятий в якості міжнародного стандарту ISO / IEC 19501:2005. У UML використовуються наступні види діаграм (для виключення неоднозначності наведені також позначення англійською мовою):

Structure Diagrams:

Class diagram  
 Component diagram  
 Composite structure diagram  
 Collaboration (UML2.0)  
 Deployment diagram  
 Object diagram  
 Package diagram  
 Profile diagram (UML2.2)

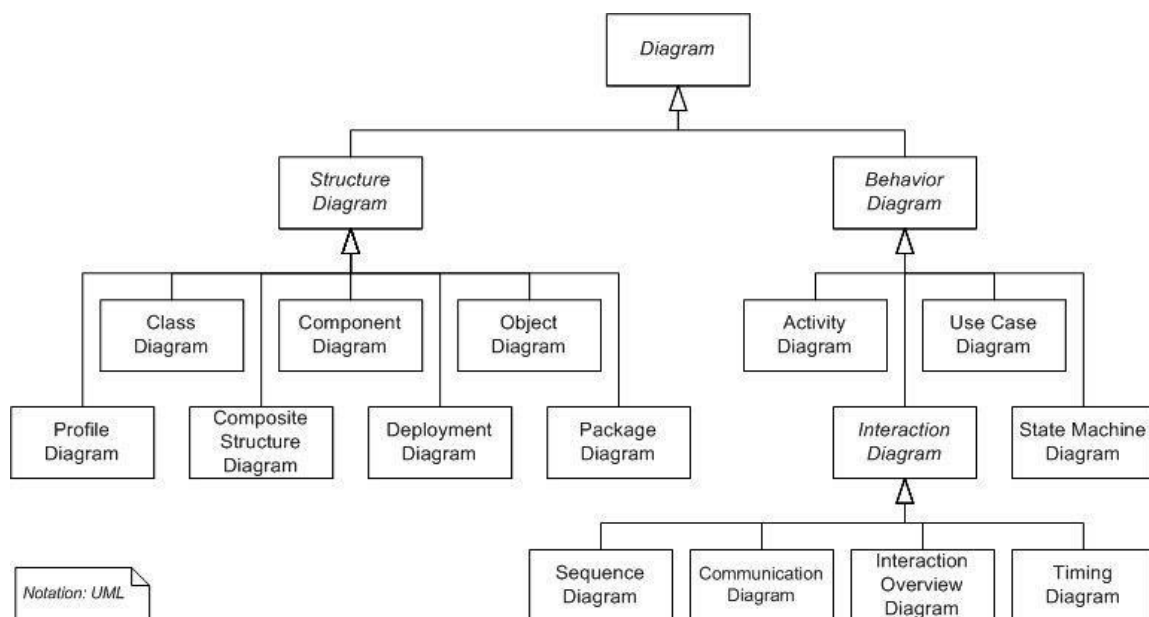
Behavior Diagrams:  
 Activity diagram  
 State Machine diagram Use  
 case diagram

Interaction Diagrams:  
 Communication diagram  
 (UML2.0) / Collaboration (UML1.x)  
 Interaction overview diagram  
 (UML2.0)  
 Sequence diagram  
 Timing diagram (UML2.0)

Структурні діаграми:

- Класів
- Компонентів
- композитної / складовою структури
- Кооперації (UML2.0)
- Розгортання
- Об'єктів
- Пакетів
- Профілів (UML2.2) Діаграми поведінки:
- Діяльності
- Станів
- Варіантів використання • Діаграми взаємодії:
  - o Комунікації (UML2.0) /
  - Кооперації (UML1.x) o Огляду взаємодії (UML2.0) o Послідовності o Синхронізації (UML2.0)

Структуру діаграм UML 2.3 можна представити на діаграмі класів UML:



## Опис типів діаграм

### 1. Діаграма класів

Діаграма класів (Class diagram) - статична структурна діаграма, що описує структуру системи, вона демонструє класи системи, їх атрибути, методи і залежності між класами.

Існують різні точки зору на побудову діаграм класів залежно від цілей їх застосування:

□ концептуальна точка зору - діаграма класів описує модель предметної області, в ній присутні лише класи прикладних об'єктів;

□ точка зору специфікації - діаграма класів застосовується при проектуванні інформаційних систем;

□ точка зору реалізації - діаграма класів містить класи, що використовуються безпосередньо в програмному коді (при використанні об'єктно-орієнтованих мов програмування).

### 2. Діаграма компонентів

Діаграма компонентів (Component diagram) - статична структурна діаграма, показує розбиття програмної системи на структурні компоненти та зв'язку (залежності) між компонентами. Як фізичних компонент можуть виступати файли, бібліотеки, модулі, виконувані файли, пакети тощо.

### 3. Діаграма композитної / складовою структури

Діаграма композитної / складовою структури (Composite structure diagram) - статична структурна діаграма, демонструє внутрішню структуру класів і, по можливості, взаємодію елементів (частин) внутрішньої структури класу.

Підвидом діаграм композитної структури є діаграми кооперації (Collaboration diagram, введені в UML 2.0), які показують ролі й взаємодію класів у рамках кооперації. Кооперації зручні при моделюванні шаблонів проектування.

Діаграми композитної структури можуть використовуватися спільно з діаграмами класів.

### 4. Діаграма розгортання



Діаграма розгортання (Deployment diagram) - служить для моделювання роботи вузлів (апаратних засобів, англ. Node) і артефактів, розгорнутих на них. У UML 2.X на вузлах розгортаються артефакти (англ. artifact), в той час як в UML 1 на вузлах розгорталися компоненти. Між артефактом і логічним елементом (компонентом), який він реалізує, встановлюється залежність маніфестації.

#### 5. Діаграма пакетів

Діаграма пакетів (Package diagram) - структурна діаграма, основним змістом якої є пакети і відносини між ними. Діаграми пакетів служать, в першу чергу, для організації елементів у групі за певною ознакою з метою спрощення структури та організації роботи з моделлю системи.

#### 6. Діаграма діяльності

Діаграма діяльності (Activity diagram) - діаграма, на якій показано декомпозиція певної діяльності на її складові частини. Під діяльністю (англ. activity) розуміється специфікація поведінки у вигляді координованого послідовного і паралельного виконання підлеглих елементів - вкладених видів діяльності та окремих дій (англ. action), 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.

#### 7. Діаграма автомата / станів

Діаграма автомата (State Machine diagram, діаграма кінцевого автомата, діаграма станів) - діаграма, на якій представлений кінцевий автомат з простими станами, переходами і композитними станами.

Кінцевий автомат (англ. State machine) - специфікація послідовності станів, через які проходить об'єкт або взаємодія у відповідь на події свого життя, а також відповідні дії об'єкта на ці події. Кінцевий автомат прикріплений до вихідного елемента (класу, кооперації або методу) і служить для визначення поведінки його екземплярів.

#### 8. Діаграма прецедентів

Діаграма прецедентів (Use case diagram, діаграма варіантів використання) - діаграма, на якій відображені відносини, що існують між акторами і прецедентами. Основне завдання - засіб, що дає можливість замовнику, кінцевому користувачеві і розробнику спільно обговорювати функціональність і поведінку системи.

#### 9. Діаграми комунікації та послідовності

Діаграми комунікації і послідовності транзитивні, описують взаємодію, але показують її різними способами і з достатнім ступенем точності можуть бути перетворені одна в іншу.

Діаграма комунікації (Communication diagram, в UML 1.x - діаграма кооперації, collaboration diagram) - діаграма, на якій зображується взаємодія між частинами композитної структури або ролями кооперації. На відміну від діаграми послідовності, на діаграмі комунікації явно вказуються відносини між елементами (об'єктами), а час як окремий вимір не використовується (застосовуються порядкові номери викликів).

Діаграма послідовності (Sequence diagram) - діаграма, на якій зображено впорядковану в часі взаємодію об'єктів. Зокрема, на ній зображуються об'єкти, що беруть участь у взаємодії і послідовність повідомлень, якими вони обмінюються.

#### 10. Діаграма співпраці

Цей тип діаграм дозволяє описати взаємодію об'єктів, абстрагуючись від послідовності передачі повідомлень. На цьому типі діаграм в компактному вигляді відображаються всі прийняті і передані повідомлення конкретного об'єкта і типи цих повідомлень.

### 11. Діаграма взаємодії




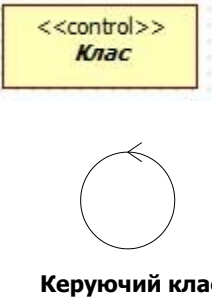
Діаграма огляду взаємодії (Interaction overview diagram) - різновид діаграми діяльності, що включає фрагменти діаграми послідовності і конструкції потоку управління.


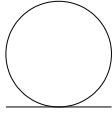
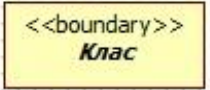

Цей тип діаграм включає в себе діаграми Sequence diagram (діаграми послідовностей дій) і Collaboration diagram (діаграми співробітництва). Ці діаграми дозволяють з різних точок зору розглянути взаємодію об'єктів у створюваній системі.

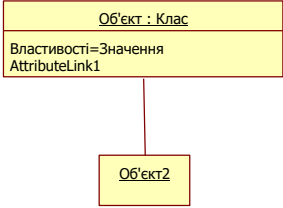
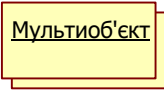
### 12. Діаграма синхронізації

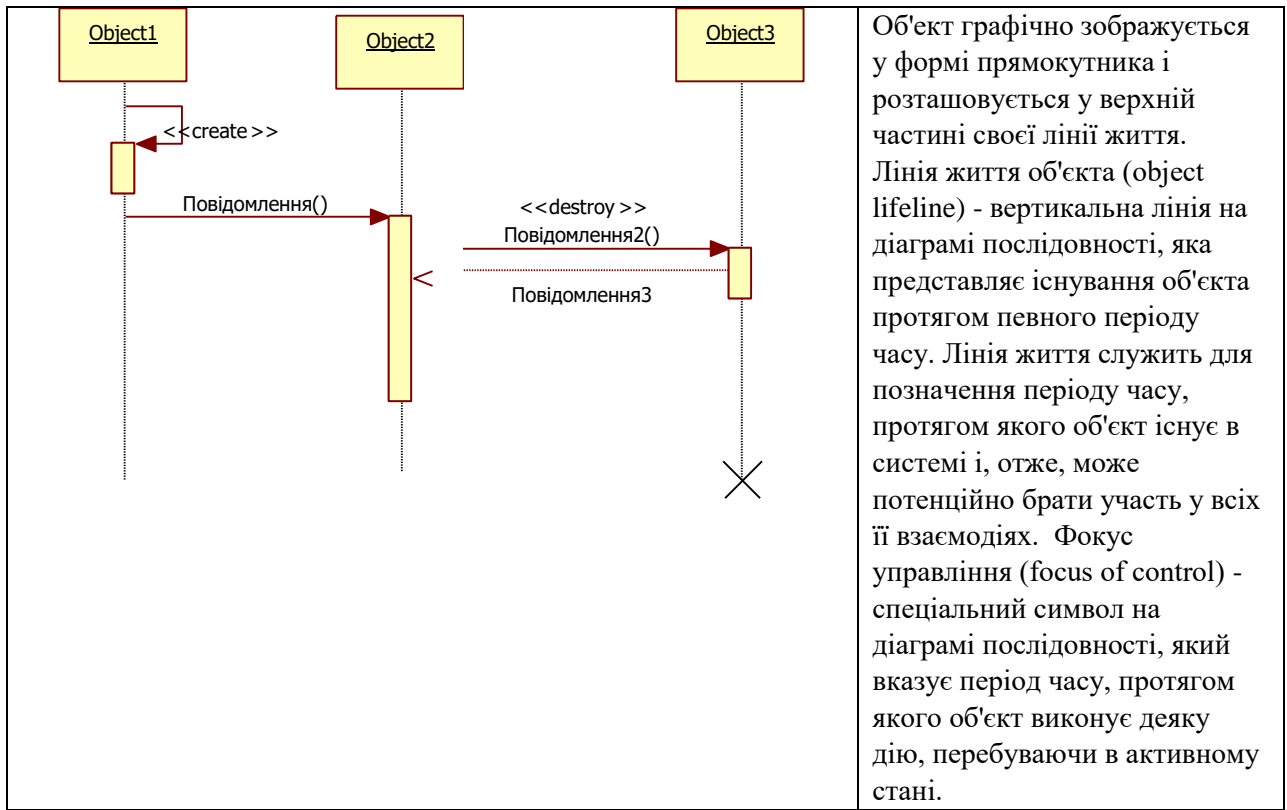
Діаграма синхронізації (Timing diagram) - альтернативне представлення діаграми послідовності, явним чином показує зміни стану на лінії життя із заданою шкалою часу. Може бути корисна в додатках реального часу.

## Основні позначення елементів діаграм

Діаграма прецедентів	
	<p>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 проєктованою системою при взаємодії її з відповідним актором, самі ці дії не зображуються на розглянутій діаграмі.</p>
	<p>актор (actor) - узгоджена множина ролей, які мають зовнішні сутності по відношенню до варіантів використання при взаємодії з ними.</p> <p>Актор являє собою будь-яку зовнішню стосовно моделюється системі сутність, яка взаємодіє з системою і використовує її функціональні можливості для досягнення певної мети або вирішення приватних завдань. Кожен актор може розглядатися як якась окрема роль щодо конкретного варіанту використання.</p>
Діаграма класів	
	<p>Клас (class) - абстрактне опис безлічі однорідних об'єктів, що мають однакові атрибути, операції і відносини з об'єктами інших класів.</p>
	<p>Керуючий клас (control class) - клас, що відповідає за координацію дій інших класів. На кожній діаграмі класів повинен бути хоча б один керуючий клас, причому кількість повідомлень, що надсилаються об'єктам керуючого класу менша, за число повідомлень, що розсилаються ними. Керуючий клас відповідає за координацію дій інших класів. Як правило, даний клас є активним і ініціює розсилку безлічі повідомлень інших класів моделі. Крім спеціального позначення керуючий клас може бути зображений у формі прямокутника класу із стереотипом &lt;&lt;control &gt;&gt;</p>

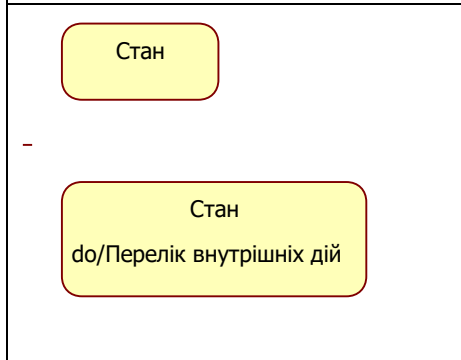
   <b>Клас-сутність</b>	<p>Клас-сутність (entity class) - пасивний клас, інформація про який повинна зберігатися постійно і не знищуватися з виключенням системи. Клас-сутність містить інформацію, яка повинна зберігатися постійно і не знищується із знищенням об'єктів даного класу або припиненням роботи модельованої системи, пов'язані з виключенням системи або завершенням програми. Як правило, цей клас відповідає окремій таблиці бази даних. У цьому випадку його атрибути є полями таблиці, а операції - приєднаними або збереженими процедурами. Цей клас пасивний і лише приймає повідомлення від інших класів моделі. Клас - сутність може бути зображений також стандартним чином у формі прямокутника класу із стереотипом &lt;&lt;entity&gt;&gt;</p>
   <b>Граничний клас</b>	<p>Граничний клас (boundary class) - клас, який розташовується на кордоні системи із зовнішнім середовищем і безпосередньо взаємодіє з акторами, але є складовою частиною системи. Граничний клас може бути зображений також стандартним чином у формі прямокутника класу із стереотипом &lt;&lt;boundary&gt;&gt;</p>

<b>Діаграма кооперації</b>	
<p style="text-align: center;"><b>Об'єкт : Клас</b></p> <p style="text-align: center;">Властивості=Значення</p>	<p>Об'єкт (object) - сутність з добре визначеними кордонами і індивідуальністю, яка інкапсулює стан і поведінку.</p>
	<p>Активний об'єкт (active object) має власний процес управління і може ініціювати діяльність з управління іншими об'єктами.</p>
	<p>Мультиоб'єкт (multiobject) являє собою безліч анонімних об'єктів, які можуть бути утворені на основі одного класу.</p>
<b>Діаграма послідовностей</b>	

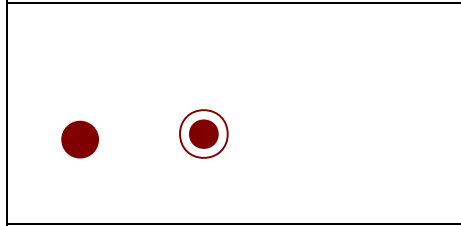


Об'єкт графічно зображується у формі прямокутника і розташовується у верхній частині своєї лінії життя. Лінія життя об'єкта (object lifeline) - вертикальна лінія на діаграмі послідовності, яка представляє існування об'єкта протягом певного періоду часу. Лінія життя служить для позначення періоду часу, протягом якого об'єкт існує в системі і, отже, може потенційно брати участь у всіх її взаємодіях. Фокус управління (focus of control) - спеціальний символ на діаграмі послідовності, який вказує період часу, протягом якого об'єкт виконує деяку дію, перебуваючи в активному стані.

Діаграма станів



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



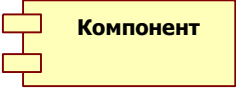

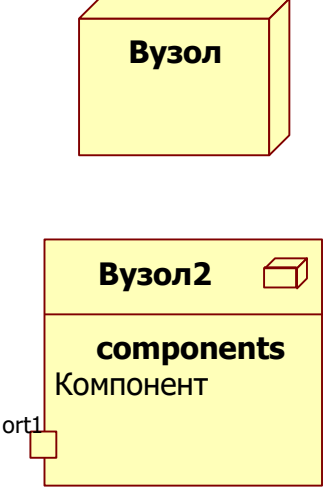


Початковий стан (start state) - різновид псевдостану, що позначає початок виконання процесу зміни станів кінцевого автомата або знаходження модельованого об'єкта в складеному стані. Кінцеве стан (final state) - різновид псевдостану, що позначає закінчення процесу зміни станів кінцевого автомата або знаходження модельованого об'єкта в складеному стані.



Складений стан (composite state) - складний стан, який складається з інших вкладених у нього станів.

Діаграма активностей

	<p>Стан діяльності (activity state) - стан у графі діяльності, яке служить для представлення процедурної послідовності дій, що вимагають певного часу. Стан дії (action state) - спеціальний випадок стану з деякою вхідною дією і, принаймні, одним виходять зі стану переходом.</p>
	<p>Розгалуження - послідовно виконувана діяльність повинна розділитися наальтернативні гілки в залежності від значення проміжного результату. Така ситуація отримала назву розгалуження, а для її позначення застосовується спеціальний символ рішення.</p>
<p>Діаграма компонентів</p>	
	<p>Компонент (component). Компонент реалізує деякий набір інтерфейсів і служить для загального позначення елементів фізичного представлення моделі.</p>
	<p>інтерфейси забезпечують не тільки сумісність різних версій, але і можливість вносити істотні зміни в одні частини програми, не змінюючи інші її частини</p>
<p>Діаграма розгортання (deployment diagram) (діаграма топологій)</p>	
	<p>Вузол (node) є деяким фізично існуючим елементом системи, що володіє деяким обчислювальним ресурсом. Як обчислювальний ресурс вузла може розглядатися наявність щонайменше деякого об'єму електронної або магнітооптичної пам'яті та/або процесора. В останній версії UML поняття вузла розширено і може містити не тільки обчислювальні пристрої (процесори), але і інші механічні або електронні пристрої: принтери, модеми, цифрові камери, сканери і ін.</p>



28,1,2,27, 26,3, 4, 25, 24, 5, 6,23, 22, 7, 8, 21, 20, 9, 10, 19, 18,11,12,17,  
16,13,14,15