

МЕТОДИЧНІ ВКАЗІВКИ ДЛЯ САМОСТІЙНОЇ РОБОТИ

**з дисципліни «Архітектура та проектування
програмного забезпечення»**

Денна форма навчання

Київ 2017

ЗМІСТ

| | |
|--|----|
| Вступ1..... | 3 |
| 1. Методологія об'єктно-орієнтованого аналізу і проектування | 4 |
| 2. Особливості процесу об'єктно-орієнтованого проектування | 5 |
| 3. Класи та об'єкти. Класифікація | 7 |
| 4. Поняття візуального програмування..... | 9 |
| 5. Елементи графічної нотації варіантів використання.... | 10 |
| 6. Огляд CASE-засобів для побудови UML діаграм | 11 |

ВСТУП

Сучасна індустрія розробки програмного забезпечення вимагає від майбутніх програмістів не лише знань з основних мов програмування та комп'ютерної техніки, але й розвинених аналітичних здібностей у процесі життєвого циклу проектування ПЗ. Складність сучасного програмного забезпечення призводить до збільшення ролі власне розробки та поступової деталізації моделі майбутнього програмного продукту, яка служить не лише засобом візуалізації абстрактних сутностей та їх поведінки, але засобом комунікації між розробником та замовником, між керівником проекту та командою програмістів.

Курс присвячений вивченню основ нотації уніфікованої мови моделювання або, скорочено, мови UML, який призначений для опису, візуалізації і документування об'єктно-орієнтованих систем та бізнес-процесів з орієнтацією на їх подальшу реалізацію у вигляді програмного забезпечення. Вивчення матеріалу курсу спрямоване на формування та вдосконалення знань з методології опису, візуалізації і документування об'єктно-орієнтованих систем за допомогою мови UML. Отримані в ході вивчення курсу знання можуть бути успішно використані в подальшому при управлінні проектами в ході розробки інформаційних моделей і програмних додатків.

Курс має своєю основною метою навчити студентів базових конструкцій мови UML використовувати різноманітні засоби роботи з використовувати CASE-засоби з метою автоматизації виконання всіх етапів концептуального, логічного та фізичного проектування архітектури програмних додатків.

При викладанні курс «Моделювання ПЗ» ґрунтується на знаннях студентів, які одержані під час вивчення дисциплін на попередніх курсах, таких як об'єктно-орієнтоване програмування, технології проектування програмного забезпечення.

1 МЕТОДОЛОГІЯ ОБ'ЄКТНО-ОРІЄНТОВАНОГО ПРОЕКТУВАННЯ ТА АНАЛІЗУ

Мета - повторити та узагальнити методологію об'єктно-орієнтованого проектування та аналізу.

Оскільки викладання даного курсу ґрунтується на знаннях студентів, які одержані під час вивчення дисциплін на попередніх курсах, таких як об'єктно-орієнтоване програмування, технології проектування програмного забезпечення, то завданням даного самостійного завдання є повторення та узагальнення попереднього вивченого матеріалу.

Після вивчення даної теми студенти повинні розуміти основні ознаки складних систем, причини складності програмних систем, математичні основи та етапи становлення об'єктно-орієнтованого підходу, історія розвитку мов програмування з точки зору ООП, особливості, принципи та переваги ОО аналізу та проектування, повторять основні поняття: клас, об'єкт, відношення між класами та об'єктами.

Вивчення даної теми доцільно починати з огляду вказаної літератури, на основі чого скласти план конспекту чи реферату, при цьому слід тримати на увазі список питань на які необхідно звернути увагу при вивченні даної теми. Результатом даної самостійної роботи є конспект чи реферат, який оцінюється викладачем.

Порядок виконання роботи

1. Вивчення даного питання у вказаних джерелах.
2. Пошук відповіді на контрольні питання.
3. Складання плану конспекту чи реферату.
4. Написання конспекту чи реферату.

Контрольні запитання

Рекомендована література: [5, гл.1-2], [8, с. 5 - 6]

1. Що таке складність, що притаманна програмному забезпеченню? Опишіть основні причини складності ПЗ.
2. Порівняйте методи аналізу і декомпозиції систем: алгоритмічну (структурну) та об'єктно-орієнтовану. В чому переваги останньої?
3. В чому суть проектування складних систем. Для чого необхідно створення об'єктно-орієнтованої моделі у процесі проектування.
4. Наведіть визначення об'єктно-орієнтованого програмування, проектування, аналізу. Що являється базисом ООП методології?
5. В чому переваги ООП А підходу над структурним підходом?
6. Опишіть розвиток мов програмування з точки зору розвитку ООП. Який вклад у розвиток ООП належить різним мовам програмування третього покоління?
7. Діаграми структурного програмування.

Теми рефератів.

1. "Срібної кулі немає". Обговорення стат'ї Фредеріка Брукса сорок років потому. [7, гл. 16-19]
2. Математичні основи об'єктно-орієнтованого аналізу та проектування. (Теорія множин. Теорія графів. Семантичні мережі). [1, с. 5 - 6].
3. Передісторія об'єктно-орієнтованого аналізу та проектування. (Діаграми функціонального моделювання. Діаграми сутність-зв'язок. Діаграми потоків даних). [1,с.5-6].
4. Основні етапи розвитку UML. [2, лек. 1]

2 ОСОБЛИВОСТІ ПРОЦЕСУ ОБ'ЄКТНО-ОРІЄНТОВАНОГО ПРОЕКТУВАННЯ ТА АНАЛІЗУ

Мета - повторити та узагальнити знання щодо процесу об'єктно-орієнтованого проектування та аналізу.

Оскільки викладання даного курсу ґрунтується на знаннях студентів, які одержані під час вивчення дисциплін на попередніх курсах, таких як об'єктно-орієнтоване програмування, технології проектування програмного забезпечення, то завданням даного самостійного завдання є повторення та узагальнення попереднього вивченого матеріалу.

Після повторення даної теми студенти повинні пригадати особливості етапи об'єктно-орієнтованого процесу розробки програмних систем, звернути увагу на роль моделювання та види моделей, які розробляються на кожному з етапів ООП проектування, повторити особливості метрик, управління ризиком і тестування за ООП методологією.

Вивчення даної теми доцільно починати з огляду вказаної літератури, на основі чого скласти план конспекту чи реферату, при цьому слід тримати на увазі список питань на які необхідно звернути увагу при вивченні даної теми. Результатом даної самостійної роботи є конспект чи реферат, який оцінюється викладачем.

Порядок виконання роботи

1. Вивчення даного питання у вказаних джерелах.
2. Пошук відповіді на контрольні питання.
3. Складання плану конспекту чи реферату.
4. Написання конспекту чи реферату.

Контрольні запитання

Рекомендована література: [5, гл.3],[9, гл.15].

1. Які риси характеризують вдалі проекти? Які ознаки гарного архітектурного рішення проекту?
2. Що Ви розумієте під раціональним процесом проектування?
3. Що Ви розумієте під еволюційно-ітеративним циклом розвитку проекту?
4. Опишіть основні етапи процесу розробки (початок, розвиток, Архітектура, перехід). Поясніть як співвідносяться потоки процесу розробки та ітерації.
5. Опишіть потоки процесу розробки (збір відомостей, аналіз, проектування, реалізація, тестування). Поясніть яку роль у процесі розробки грають моделі та моделювання.
6. Опишіть особливості кожного з технічних артефактів (набір вимог, реалізації, проектування, розміщення).
7. Як співвідносяться мікропроцес та мікропроцес об'єктно-орієнтованої розробки?
8. Опишіть етап мікропроцесу проектування, що пов'язаний з ідентифікацією об'єктів та даних на даному рівні абстракції. Які основні види діяльності на цьому етапі?
9. Опишіть етап мікропроцесу проектування, що пов'язаний з виявленням семантики класів і об'єктів. Які основні види діяльності на цьому етапі?
10. Опишіть етап мікропроцесу проектування, що пов'язаний з виявленням зв'язків між класами та об'єктами. Які основні види діяльності на цьому етапі?
11. Опишіть етап мікропроцесу проектування, що пов'язаний з реалізацією класів та об'єктів. Які основні види діяльності на цьому етапі?
12. Яка мета першого кроку мікропроцесу ООП - концептуалізації, які основні види діяльності на цьому кроці?
13. Яка мета другого кроку мікропроцесу ООП - аналізу, які основні види діяльності на цьому кроці?

1. Яка мета третього кроку мікропроцесу - проектування, які основні види діяльності на цьому кроці?
2. Яка мета четвертого кроку мікропроцесу проектування - еволюції, яка меті і основні види діяльності на цьому кроці?
3. Яка мета четвертого кроку мікропроцесу проектування - супроводу, яка меті і основні види діяльності на цьому кроці?

Теми рефератів.

1. Управління ризиком у процесі об'єктно-орієнтованого проектування.
2. Особливості метрик об'єктно-орієнтованих програмних систем.
3. Метрики об'єктно-орієнтованого тестування.
4. Приклад об'єктно-орієнтованого процесу розробки програмного забезпечення.
5. Особливості об'єктно-орієнтованого тестування.
6. Проектування об'єктно-орієнтованих тестових варіантів.
7. Моделі реалізації об'єктно-орієнтованих програмних систем.

3 КЛАСИ ТА ОБ'ЄКТИ. КЛАСИФІКАЦІЯ

Мета - повторити та узагальнити методологію об'єктно-орієнтованого проектування та аналізу.

Оскільки викладання даного курсу ґрунтується на знаннях студентів, які одержані під час вивчення дисциплін на попередніх курсах, таких як об'єктно-орієнтоване програмування, технології проектування програмного забезпечення, то завданням даного самостійного завдання є повторення та узагальнення попереднього вивченого матеріалу.

Після вивчення даної теми студенти повинні розуміти основні поняття: клас, об'єкт, відношення між класами та об'єктами, а також основні правила вибору відношень, операцій та реалізації. Студенти повинні також усвідомити види та роль класифікації у об'єктно-орієнтованому аналізі та проектуванні.

Вивчення даної теми доцільно починати з огляду вказаної літератури, на основі чого скласти план конспекту чи реферату, при цьому слід тримати на увазі список питань на які необхідно звернути увагу при вивченні даної теми. Результатом даної самостійної роботи є конспект чи реферат, який оцінюється викладачем.

Порядок виконання роботи

1. Вивчення даного питання у вказаних джерелах.
2. Пошук відповіді на контрольні питання.
3. Складання плану конспекту чи реферату.
4. Написання конспекту чи реферату.

Контрольні запитання

Рекомендована література: [8, с. 7- 18], [5, гл. 2-3], [9, гл. 9].

1. Опишіть основні принципи ОО підходу (абстрагування, модульність, ієрархія, обмеження доступу). Які переваги та особливості надає ОО підходу кожен з цих принципів?
2. Опишіть додаткові принципи ОО підходу (типізація, збереження, паралелізм). Які переваги та особливості надає ОО підходу кожен з цих принципів?
3. Природа класу та об'єкту. Спільне та відмінне.
4. Об'єкти: стан, поведінка, операції, ідентичність. Час життя об'єктів.
5. Відношення між об'єктами. Ролі об'єктів. Зв'язки, синхронізація, агрегація.
6. Природа класів. Життєвий цикл класів. Інтерфейс та реалізація.
7. Відношення між класами. Наслідування, асоціація, агрегація, використання, інстанціювання, метаклас.
8. Відношення між класами та об'єктами. Роль класів та об'єктів у аналізі та проектуванні.
9. Якість класів та об'єктів. Критерії: зчеплення, зв'язність, повнота, достатність, примітивність.
10. Правила вибору операцій. Функціональність. Пам'ять.
11. Правила для визначення типу відношень. Співробітництво. Механізми та видимість.
12. Вибір реалізації. Представлення. Модульна структура.
13. Завдання класифікації. Підходи до класифікації: розподіл по категоріям, концептуальна кластеризація, теорія прототипів. Роль класифікації в ООП.
14. Класичний підхід до ідентифікації класів та об'єктів.
15. Сучасні підходи до класифікації: аналіз поведінки, предметної області, варіантів, CRC картки.
16. Класифікація через ключові абстракції та механізми.

Теми рефератів.

1. Розвиток теорій класифікацій у філософії (від Платона і Арістотеля до Фоми Аквінського, Лока, Декарта).
2. Методологія системного аналізу і системного моделювання та її використання у процесі розробки програмного забезпечення.
3. Принципи об'єктно-орієнтованих представлення програмних систем.
4. Методи декомпозиції моделі на модулі та основні властивості модулів програмної системи.
5. Класичні методи структурного проектування програмних систем.
6. Поглиблене вивчення класів у мові UML. [6, зл.10]
7. Поглиблене вивчення відношень у мові UML. [6, зл.11]

4 ПОНЯТТЯ ВІЗУАЛЬНОГО ПРОГРАМУВАННЯ

Мета - ознайомитися з поняттям візуального програмування.

Після вивчення даної теми студенти повинні розуміти основні поняття: візуальне програмування, CASE-засіб, множинність точок зору при візуальному моделюванні, семантичний розрив між моделлю і програмним кодом, “невидимість” ПО, графі та діаграми візуальних моделей.

Вивчення даної теми доцільно починати з огляду вказаної літератури, на основі чого скласти план конспекту чи реферату, при цьому слід тримати на увазі список питань на які необхідно звернути увагу при вивченні даної теми. Результатом даної самостійної роботи є конспект чи реферат, який оцінюється викладачем.

Порядок виконання роботи

1. Вивчення даного питання у вказаних джерелах.
2. Пошук відповіді на контрольні питання.
3. Складання плану конспекту чи реферату.
4. Написання конспекту чи реферату.

Контрольні запитання

Рекомендована література: [4, лек. 1]

1. Яка відмінність між засобами проектування у інженерії загалом та у програмній інженерії. Роль креслень.
2. Визначення візуального моделювання. Засоби візуального моделювання.
3. Програмні засоби візуального моделювання: універсальні та предметно-орієнтовані CASE-засоби.
4. Місце мов візуального моделювання у еволюції засобів програмування.
5. У чому полягає семантичний розрив візуальних моделей і програмного коду і його наслідки.
6. Поясніть терміни предметна область, модель, метамодель і метаметамодель?
7. Поясніть, що являють собою чотири рівня візуального моделювання. Чим вони відрізняються?
8. З чим пов'язано використання множинності точок зору при візуальному моделюванні ПЗ?
9. Навіщо для візуальних моделей виділяти граф моделі та діаграми? Що таке браузер моделі і навіщо він потрібен?
10. Розкажіть про операції над графом моделі. Розкажіть про операції над діаграмами. Розкажіть про поєднанні операцій над діаграмами з операціями над графом моделі.
11. Що таке репозиторій CASE-пакета? Розкажіть про способи його реалізації. Розкажіть про операції над графом моделі через браузер і засобами стороннього додатка (через відкритий програмний інтерфейс).

Теми рефератів.

1. Порівняльна характеристика найпоширеніших CASE-засобів.
2. Історія розвитку CASE-засобів і їх перспективи на майбутнє.
3. Історія розвитку мов візуального проектування та об'єктно-орієнтованого моделювання.

5 ЕЛЕМЕНТИ ГРАФІЧНОЇ НОТАЦІЇ ВАРІАНТІВ ВИКОРИСТАННЯ.

Мета - повторити та узагальнити теоретичний матеріал щодо діаграм прецедентів використання.

Після повторення даної теми студенти повинні знати, які основні графічні елементи використовується на діаграмах прецедентів, які типи зв'язків та стереотипів використовуються на цих діаграмах, хто може бути актором, на якому етапі проектування і з якою цілю використовуються ці діаграми.

Вивчення даної теми доцільно починати з огляду вказаної літератури, на основі чого скласти план конспекту чи реферату, при цьому слід тримати на увазі список питань на які необхідно звернути увагу при вивченні даної теми. Результатом даної самостійної роботи є конспект чи реферат, який оцінюється викладачем.

Порядок виконання роботи

1. Вивчення даного питання у вказаних джерелах.
2. Пошук відповіді на контрольні питання.
3. Складання плану конспекту чи реферату.
4. Написання конспекту чи реферату.

Контрольні запитання

Рекомендована література: [2, лек. 3-4], [б, гл.17]

1. До якого типу моделей системи, що проектується відносяться діаграма варіантів використання?
2. Для яких цілей використовують діаграми варіантів використання?
3. Які особливості позначень відношення на діаграмі варіантів використання?
4. Які графічні примітиви використовуються на діаграмі прецедентів?
5. Як можна формалізувати функціональні вимоги до системи за допомогою діаграми варіантів використання?
6. Що так вимога, що таке сценарій використання. Наведіть приклади.
7. Які особливості специфікації функціональних вимог на діаграмі варіантів використання?
8. Наведіть приклад рекомендації з розробки діаграм варіантів використання.

Теми рефератів.

1. Особливості використання зв'язків та стереотипів на діаграмі Use Case.
2. Стереотипи у мові UML. Порівняння версій UML 1.0 та 2.0.

6 ОГЛЯД CASE-ЗАСОБІВ ДЛЯ ПОБУДОВИ UML ДІАГРАМ.

Мета - розшири знання щодо класифікації та основних можливостей сучасних CASE засобів.

Після вивчення даної теми студенти повинні знати, які можливості є у сучасних CASE засобів щодо побудови діаграм UML, знати критерії вибору таких засобів, як вбудованих у середовища розробки так і незалежних, знати на яких етапах розробки програмного продукту доцільно CASE засоби і для чого, розуміти поняття кодогенерації, від чого залежить його якість та обмеження автоматичної кодогенерації.

Вивчення даної теми доцільно починати з огляду вказаної літератури, на основі чого скласти план конспекту чи реферату, при цьому слід тримати на увазі список питань на які необхідно звернути увагу при вивченні даної теми. Результатом даної самостійної роботи є конспект чи реферат, який оцінюється викладачем.

Порядок виконання роботи

1. Вивчення даного питання у вказаних джерелах.
2. Пошук відповіді на контрольні питання.
3. Складання плану конспекту чи реферату.
4. Написання конспекту чи реферату.

Орієнтовний план.

1. IBM Rational Rose
2. Borland Together
3. Microsoft Visio
4. Sparx Systems Enterprise Architect
5. Gentleware Poseidon
6. SmartDraw
7. Діаметр
8. Telelogic TAU G2
9. StarUML
10. Інші програми

Контрольні запитання

Рекомендована література: [3, лек.7], [8, с. 44-55]

1. До якого типу моделей системи, що проектується відносяться діаграма варіантів використання?
2. Для яких цілей використовують діаграми варіантів використання?
3. Які особливості позначень відношення на діаграмі варіантів використання?
4. Які графічні примітиви використовуються на діаграмі прецедентів?
5. Як можна формалізувати функціональні вимоги до системи за допомогою діаграми варіантів використання?
6. Що так вимога, що таке сценарій використання. Наведіть приклади.
7. Які особливості специфікації функціональних вимог на діаграмі варіантів використання?
8. Наведіть приклад рекомендації з розробки діаграм варіантів використання.

Теми рефератів.

1. Можливості CASE засобу IBM Rational Rose по побудові UML діаграм.
2. Можливості CASE засобу Borland Together по побудові UML діаграм.
3. Можливості CASE засобу Microsoft Visio по побудові UML діаграм.
4. Можливості CASE засобу Umbrello Modeling UML по побудові UML діаграм.

ЛІТЕРАТУРА.

1. Леоненков А.В. Самоучитель UML / А. В. Леоненков. - СПб. : БХВ-Петербург, 2004. - 432 с.
2. Леоненков А.В. Нотация и семантика языка UML. - Электронный ресурс. - <http://www.intuit.ru/department/pl/umlbasics/>
3. Бабич А. В. Введение в UML. - Электронный ресурс. - <http://www.intuit.ru/department/se/intuml/>
4. Д.В.Кознов Визуальное моделирование: теория и практика. - Электронный ресурс. - <http://www.intuit.ru/department/se/vismodtp/>
5. 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.
6. Г. Буч, Дж. Рамбо, А. Джекобсон Язык UML. Руководство пользователя.: Пер. с англ. - М.: ДМК, 2000. - 432с.
7. Брукс Ф. Мифический человеко-месяц или как создаются программные системы. — Пер. с англ. — СПб.: Символ-Плюс, 2001. — 304 с.
8. Глотова Т. В. Объектно-ориентированная методология разработки сложных систем. Учебное пособие. - 49 с.
9. Орлов С. Технологии разработки программного обеспечения: Учебник/ С. Орлов. — СПб.: Питер, 2002. — 464 с.: ил.
10. Кватрани Т. Визуальное моделирование с помощью Rational Rose 2002 и UML. - М.: Вильямс, 2003. - 192 с.