

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування

імені адмірала Макарова

Є. Ю. БЕРКУНСЬКИЙ

МЕТОДИЧНІ ВКАЗІВКИ

до виконання курсових робіт з дисципліни

«Технологія об'єктно-орієнтованого проектування»

Рекомендовано Методичною радою НУК

Електронне видання

комбінованого використання на DVD-ROM

Миколаїв ♦ НУК ♦ 2020

Вступ

За останні двадцять років платформа Java набула великої популярності у користувачів, це стало можливим завдяки великим можливостям, які закладені у її основу – мову програмування Java та віртуальну машину Java (JVM). Програмна платформа Java – ряд програмних продуктів та специфікацій компанії Sun Microsystems, раніше незалежної компанії, а нині підрозділу корпорації Oracle, які спільно надають систему для розробки прикладного програмного забезпечення та вбудовування її в будь-яке крос-платформенне програмне забезпечення. Java використовується в самих різних комп'ютерних платформах від вбудованих пристроїв і мобільних телефонів у нижньому ціновому сегменті, до корпоративних серверів і суперкомп'ютерів у вищому ціновому сегменті. Хоча Java-аплети рідко використовуються в настільних комп'ютерах, проте вони в них іноді використовуються для поліпшення функціональності і підвищення безпеки при перегляді всесвітньої павутини.

Програмний код, написаний на Java, віртуальна машина Java перетворює в байт-код Java.

Існує чотири варіанти платформ Java: Java Card, Java ME, Java SE, Java EE.

Java Card: Технологія, яка дозволяє невеликим Java-програмам надійно працювати на смарт-картах та інших пристроях з малим об'ємом пам'яті.

Java ME: Включає в себе кілька різних наборів бібліотек (відомих як профілі) для пристроїв з обмеженим об'ємом місця для зберігання, невеликим розміром дисплея і батареї. Використовується для розробки програм для мобільних пристроїв, КПК, TV-ресиверів і принтерів.

Java SE: для використання на ПК, ноутбуках, та невеликих серверах.

Java EE: Java SE плюс API, корисне для багатопотокових клієнт-серверних бізнес-застосувань.

В курсовій роботі розроблюється програма для платформи Java SE.

1 Порядок виконання курсової роботи

1.1 Загальні положення

Курсова робота з дисципліни "Мова програмування Java" виконується студентами, як підсумкове завдання, та відіграє роль підсумкового контролю з дисципліни. Згідно вимог кафедри, курсова робота, яка присвячена розробці програмного забезпечення, повинна складатися з таких розділів:

1. Опис та аналіз предметної області
2. Постановка задачі на розробку програмного забезпечення
3. Розробка програмного забезпечення
4. Аналіз отриманих результатів
5. Список використаних джерел інформації
6. Додатки (Технічний опис, інструкція з експлуатації, код програми, знімки екранних форм і т.п.)

1.2. Опис розділів

1.2.1 Опис та аналіз предметної області

Опис предметної області виконується з метою формулювання технічного завдання на розробку програмного забезпечення, та повинен містити такі етапи:

1. Визначення мети та призначення розробки, виявлення предметної області проекту.
2. Збирання інформації по темі роботи, аналіз існуючих у даній галузі рішень. Головною метою даного етапу є виявлення основних характерних рис, технологій, архітектури та дизайну проекту.
3. Вибір та обґрунтування критеріїв якості програмного забезпечення, формування вимог та обмежень.

4. Визначення рішень, технологій (або навіть програмних бібліотек), які присутні у готових продуктах, та можуть бути застосовані в даній розробці.
5. Визначення напрямків досліджень для реалізації механізмів, що не є очевидними та не можуть бути повторно використаними чи запозиченими.

Після аналізу предметної області визначаються можливості використання певних технологій проектування. На цьому етапі треба також враховувати архітектурні та функціональні вимоги до проекту, кількість учасників, перспективи подальшого розвитку, вимоги до дизайну. Якщо проект, що розроблюється, буде частиною іншого проекту, треба передбачити механізми інтеграції з існуючими частинами проекту. Для цього може знадобитися визначити технологію та інструментальні засоби розробки, а саме – мову програмування, бібліотеки та каркаси розробки.

1.2.2 Постановка задачі на розробку програмного забезпечення

На основі результатів, отриманих внаслідок аналізу предметної області, виконується розробка технічного завдання (постановка задачі). У технічному завданні потрібно чітко та однозначно сформулювати галузь застосування, призначення та мету проекту; підстави для розробки, вимоги до архітектури, технологій та інструментів розробки проекту; вимоги до функціональних, технічних і експлуатаційних характеристик об'єкту розробки; визначення структури вхідних і вихідних даних; зв'язок проекту з іншими продуктами, характер використання результатів; вимоги до документації, основні етапи і терміни розробки; методи тестування.

Проект програми

Виконання проекту передбачає наступні етапи: *ескізний проект*, на якому приймаються загальні рішення щодо обрання інструментів

розробки, побудування моделей і визначення засобів взаємодії продукту з оточенням (користувачем, обладнанням); *технічний проект*, на якому приймаються спеціальні рішення з деталізації алгоритмів, документування алгоритмів та інтерфейсів; *робочий проект*, на якому проводиться кодування програми.

На основі технічного завдання і аналізу вимог до технічних і експлуатаційних характеристик об'єкта розробки необхідно визначити технологію проектування.

Після вибору технології проектування треба визначити інструментальні засоби створення проекту. До них можна віднести мову програмування, транслятор, бібліотеки, та середовище розробки, що відповідають сучасним вимогам до мови програмування, наприклад, якщо обрано мову Java, компілятор повинен підтримувати сучасну версію JDK – Java Development Kit (JDK 8, JDK11 чи більш нову).

За останні роки для будь-якої мови програмування створено велику кількість бібліотек. Навіть стандартна бібліотека мови Java складається з декількох тисяч класів, що дозволяють вирішувати найрізноманітніші завдання. Щодо середовища програмування, воно повинне бути інтуїтивно зрозумілим, зручним та швидким в роботі. Фактично, стандартними середовищами для мови програмування Java є NetBeans IDE – вільне безкоштовне середовище, та IntelliJ IDEA – багатофункціональне комерційне середовище, проте безкоштовне для використання у освітніх закладах.

Оскільки мова Java є об'єктно-орієнтованою, обов'язковим елементом проекту є побудова діаграми ієрархії об'єктів програми. На цьому етапі виконується побудова ієрархії об'єктів, що відображає склад і структуру необхідних для реалізації проекту абстракцій. Ієрархія об'єктів повинна бути описана з визначенням специфікацій для всіх об'єктів та проілюстрована графічно. Окремим об'єктам на графічному

зображені повинні відповідати окремі прямокутники, у яких указуються назви відповідних об'єктів і назви (прототипи) інтерфейсних функцій. Зв'язки підпорядкування між об'єктами зображуються у вигляді ліній або стрілок, які вказують напрямок поширення керуючих впливів. Інтерфейси, що реалізують додаткові засоби взаємодії, графічно повинні бути зображені відмінними від інтерфейсів підпорядкування засобами з обов'язковим вказанням напрямку поширення керуючих впливів.

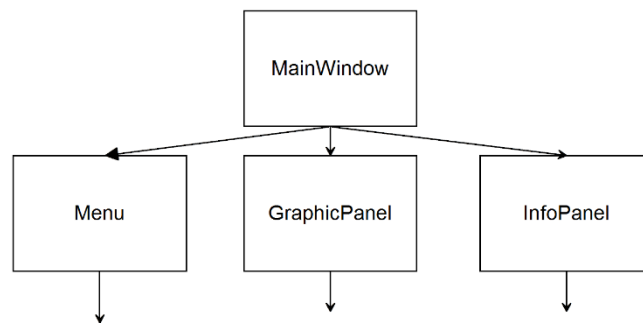


Рисунок 1 – Ієрархія об'єктів програми

На основі ієрархії об'єктів, описаної в попередньому розділі, розробляється ієрархія успадкування класів проекту. Опис ієрархії успадкування повинен включати специфікації класів та інтерфейсів. Результати проектування класів повинні бути подані графічно з виділенням абстрактних (графічно зображуються у вигляді прямокутників з переривчастими контурами) і кінцевих (графічно зображуються у вигляді прямокутників із суцільними контурами) класів. В прямокутниках необхідно вказати назви відповідних класів.

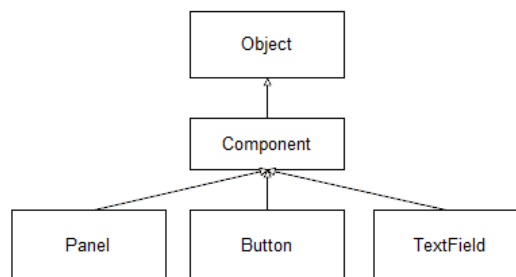


Рисунок 2 – Ієрархія успадкування класів

Після опису ієрархії успадкування класів, необхідно провести опис класів, їхніх полів та методів. Цей розділ повинен містити деталізацію алгоритмів та технічну документацію на складові частини програми з докладним описом інтерфейсів, властивостей і поведінки класів, принципів організації даних та алгоритмів функцій з обґрунтуванням прийнятих рішень. Важливі для проекту алгоритми потрібно ілюструвати за допомогою блок-схем або діаграм взаємодії (UML). Якщо в розробці було використано готові програмні рішення інших розробників (бібліотеки, фреймворки та ін.), необхідно навести короткі описи цих рішень з обґрунтуванням доцільності їхнього вибору.

1.2.3 Розробка програмного забезпечення

На основі рішень, прийнятих у попередніх розділах можна приступати до реалізації проектних рішень та тестування програми, що буде створена. Цей розділ роботи повинен містити опис основних етапів реалізації програми, а також використаних на кожному з цих етапів, методик тестування. Необхідно також враховувати, що розробка програми проводиться на комп'ютері розробника в умовах, що можуть достатньо сильно відрізнятись від умов експлуатації. Тому, для остаточного (приймального) етапу тестування, необхідно вказати умови, в яких буде працювати закінчений програмний продукт. Якщо при тестуванні окремих елементів програми використовувались додаткові програми, або тестові фреймворки, то необхідно навести код цих програм, або дати посилання на опис тестового фреймворку.

1.2.4 Аналіз отриманих результатів

Основне призначення такого аналізу – визначення, в якій мірі результати виконаної роботи відповідають початковим вимогам до програмного продукту і поточного стану справ у предметній області. Також, цей аналіз може бути використаний для формування на його основі планів і перспективних напрямків розвитку проекту або його частин. Виконання

аналізу може бути необхідним з таких причин: у роботі над проектом функціональність або властивості окремих його частин були виконані частково, або з обмеженнями; в результаті роботи над проектом було з'ясовано, що його можна розширити та доповнити новими функціями, або поліпшити якість виконання існуючих функцій. Також, не виключається ситуація, що під час роботи над проектом у предметній області відбулись важливі для проекту зміни.

1.2.5 Список літератури

Впорядкований список літератури повинен бути пронумерований арабськими цифрами з крапкою. Основні вимоги до списку літератури:

- відповідність темі курсової роботи;
- різноманітність видів видань (нормативні, навчальні, наукові та ін.)
- перелік літератури подається в алфавітному порядку. Посилання на неї в тексті беруться у квадратні дужки. На неопубліковані роботи посилання не допускаються;
- бібліографічний опис літератури наводять згідно з діючим стандартом ГОСТ 7.1–2006 для бібліографічної та видавничої справи

1.2.6 Додатки

Тексти програм повинні починатися з найменування програмних модулів (класів, файлів). Кожен з них повинен починатися з нової сторінки. Кожний клас повинен мати початковий коментар, що описує призначення класу, його взаємодію з іншими класами та інші додаткові відомості, що необхідні для розуміння роботи класу. При написанні програм бажано враховувати необхідність друку кодів програмних модулів та, за можливістю, не припускати розривання рядків коду засобами друку текстів.

Якщо програма має графічний інтерфейс користувача, це повинно бути відображено у додатках із використанням знімків екранних форм – ”скріншотів”.