

МЕТОДИЧНІ ВКАЗІВКИ

до виконання курсової роботи з дисципліни

"ОБ'ЄКТНО – ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ"

для студентів спеціальності 121 «Інженерія програмного забезпечення»

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра програмного забезпечення

МЕТОДИЧНІ ВКАЗІВКИ

до виконання курсової роботи з дисципліни

"ОБ'ЄКТНО – ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ"

для студентів спеціальності 121 «Інженерія програмного забезпечення»

Вінниця
ВНТУ
2018

Рекомендовано до друку Методичною радою Вінницького національного технічного університету Міністерства освіти і науки України (Протокол № від 2018 р.)

Рецензенти:

Ю.В. Булига, кандидат технічних наук, доцент кафедри ГМ ВНТУ, голова комісії Методичної ради ВНТУ з удосконалення курсового проектування

В.О. Денисюк, кандидат технічних наук, доцент, кафедра ЕК ВНАУ

І.Р. Арсенюк, кандидат технічних наук, доцент, кафедра КН ВНТУ

Методичні вказівки до виконання курсової роботи з дисципліни «Об'єктно-орієнтоване програмування» / Уклад. Д.І. Кательніков, А.В. Денисюк. – Вінниця: ВНТУ, 2018. – 45 с.

Методичні вказівки призначені для надання допомоги при виконанні курсової роботи з дисципліни " Об'єктно-орієнтоване програмування " і оформленні пояснювальної записки до неї. Наведено перелік можливих тем для розробки, сформульовано вимоги до програмного засобу, що реалізуватиме основну мету роботи, дано рекомендації щодо коректного оформлення пояснювальної записки, наведено критерії оцінювання курсової роботи.

ЗМІСТ

1 ОСНОВНІ ВИМОГИ ДО КУРСОВОЇ РОБОТИ	5
1.1 Мета і задачі курсової роботи	5
1.2 Тематика курсової роботи	6
1.3 Вимоги до розробки програмного засобу	7
2 ОФОРМЛЕННЯ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ	9
2.1 Загальні вимоги до оформлення	9
2.2 Структура пояснювальної записки	12
2.3 Вступна частина пояснювальної записки	12
2.4 Вміст і оформлення основної частини	14
2.5 Оформлення додатків	20
2.6 Оформлення ілюстративної частини	21
3 ГРАФІК ВИКОНАННЯ КУРСОВОЇ РОБОТИ І ПОРЯДОК ЗАХИСТУ	22
3.1 Графік виконання	22
3.2 Оцінювання виконання курсової роботи	23
ПЕРЕЛІК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ	24
Додаток А. Варіанти завдань на курсову роботу	25
Додаток Б. Приклад оформлення титульного аркуша	36
Додаток В. Приклад оформлення індивідуального завдання	37
Додаток Г. Використання схем алгоритмів, програм, даних і систем	38
Додаток Д. Приклади UML-діаграм	44

1 ОСНОВНІ ВИМОГИ ДО КУРСОВОЇ РОБОТИ

1.1 Мета і задачі курсової роботи

Курсова робота (КР) з дисципліни «Об'єктно-орієнтоване програмування» спрямована на отримання студентом базових знань та практичних навичок з основ сучасної технології створення складних програмних продуктів на базі ідей і принципів об'єктно-орієнтованого методу. Такі знання призначені для використання у розробках програмного забезпечення інформаційних технологій у проектуванні з урахуванням сучасних вимог у відношенні до надійності, якості інтерфейсу та ефективності програмних продуктів, які створюються. Отримані знання та практичні навички мають служити базою для опанування у подальшому нових майбутніх систем програмування, які базуються на ідеях візуального програмування, CASE-технологіях, штучного інтелекту і т.і.

Під час виконання курсової роботи студенти повинні використати знання, отримані ними під час вивчення дисциплін “Основи програмування” та “Комп'ютерна дискретна математика”.

Для успішного виконання курсової роботи студенти повинні вміти:

- коректно сформулювати задачу для розв'язання її на певному типі обчислювальних машин, у певному операційному середовищі;
- виконувати об'єктно-орієнтований аналіз та проектування предметної області задачі;
- виконувати ідентифікацію та класифікацію класів та об'єктів предметної області для якої проектується інформаційна система;
- створювати об'єктно - орієнтовану модель інформаційної системи;
- виконувати проектування та моделювання інформаційних систем засобами середовища візуального проектування;
- звести поставлення задачі до розробки алгоритму і визначити структури даних, які дозволяють перейти від абстрактного формулювання алгоритму до конструювання структурної схеми;
- здійснювати програмну реалізацію інформаційної системи, використовуючи мову об'єктно-орієнтованого програмування Java;
- подати власну розробку таким чином, щоб нею могли користуватись інші, підготувавши для цього відповідний методичний матеріал у вигляді детальних інструкцій та рекомендацій.

Завдання для курсових робіт визначаються викладачем із загального списку завдань, затверджених кафедрою програмного забезпечення. Заохочуються пропозиції студентів щодо самостійного, за узгодженням з викладачем, вибору теми курсової роботи поза межами запропонованого в методичних вказівках переліку.

Метою індивідуальних завдань є закріплення теоретичних та практичних навичок у роботі з використання сучасних технологій об'єктно-

орієнтованого програмування шляхом використання сучасних мов і технологій програмування.

1.2 Тематика курсової роботи

Зміст курсової роботи відповідає навчальній програмі та робочому плану дисципліни “Об’єктно-орієнтоване програмування” і визначається завданням, яке видається не пізніше шести днів з початку семестру на консультації викладачем кожному студенту.

Курсове проектування включає декілька послідовних етапів, які повинні бути пов’язані зі змістовним постановленням задачі, розробкою індивідуального та технічного завдання, об’єктно-орієнтованим аналізом та проектуванням предметної області задачі з побудовою відповідних UML-діаграм, розробкою як самих класів та об’єктів так і засобів їх інстанціювання, серіалізації/де-серіалізації і синхронізації, побудови інтуїтивного графічного інтерфейсу програми.

Орієнтовний графік виконання курсової роботи наведено у подальших розділах. Студентам бажано дотримуватись наведеного графіка, починаючи роботу над проектом з перших тижнів семестру. Під час проведення поточного контролю знань з дисципліни "Об’єктно-орієнтоване програмування" якість, зацікавленість та ефективність роботи студентів буде оцінюватись відповідною кількістю балів, що впливатимуть на загальну оцінку з дисципліни за модуль.

Тематика курсової роботи пов’язана з майбутньою спеціальністю студентів. Програмна реалізація даної курсової роботи передбачає володіння студентом наступними механізмами об’єктно-орієнтованого програмування :

1. Структура об’єктів
2. Інкапсуляція
3. Повідомлення
4. Класи
5. Спадкоємство
6. Породження класів
7. Перевантаження методів
8. Модифікатори доступу
9. Обробка подій
10. Створення об’єктів, конструктори, деструктори
11. Опрацювання рядкових величин
12. Пакети та їх імпортування
13. Інтерфейси та їх реалізація
14. Виключення та їх реалізація
15. Ідентифікатори і типи даних
16. Оператори розгалуження
17. Оператори циклів

18. Масиви
19. Створення компонентів інтерфейсу користувача
20. Робота з вікнами
21. Робота з числовими типами даних
22. Проектування програм у вигляді ієрархії класів
23. Приведення типів між класами
24. Знищення об'єктів
25. Принципи роботи трансляторів мов програмування
26. Логічні типи даних та їх обробка

Крім того об'єктом для курсової роботи можуть бути реалізація тестових програм, розробка лабораторних практикумів для даної дисципліни та для інших дисциплін, пов'язаних із інженерією програмного забезпечення.

Метою проектування повинен бути вибір такого методу об'єктної декомпозиції предметної області, що забезпечить не тільки повноту відтворення бажаного функціоналу але й буде оптимальним з точки зору витрат часу на програмування. Таким чином, для вибору способу об'єктної декомпозиції предметної області важливим є індивідуальний підхід у кожному конкретному випадку.

Студент, вибравши собі завдання для курсової роботи, повинен чітко уявляти собі, вимоги користувача системи до функціоналу системи та юзабіліті її інтерфейсу.

Індивідуальне завдання для курсових робіт визначається викладачем із загального списку завдань на курсову роботу. Заохочуються пропозиції студентів щодо самостійного, за узгодженням з викладачем, вибору теми курсової роботи поза межами запропонованого в методичних вказівках переліку. Самостійний вибір предметної області дозволяє зробити висновок щодо рівня творчої активності студента, його вміння самостійно здійснити попередній аналіз предметної області, поставити перед собою конкретну задачу та ефективно її реалізувати.

1.3 Вимоги до розробки програмного засобу

Результатом виконання курсової роботи повинен бути, як правило, повноцінний програмний додаток для конкретної операційної системи (Windows, Unix, Android, IOS), що реалізує замовлений функціонал.

Виконання проекту передбачає наступні етапи: *ескізний проект*, на якому приймаються загальні рішення щодо обрання інструментів розробки, побудови 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 й 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 (на 2017–2018 рік це – JDK8). 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 у роботі. Фактично, стандартні середовищами для мови програмування Java є NetBeans IDE – вільне безкоштовне середовище, та IntelliJ IDEA – багатофункціональне комерційне середовище, проте безкоштовне для використання в освітніх закладах.

2 ОФОРМЛЕННЯ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

2.1 Загальні вимоги до оформлення

При оформленні пояснювальної записки (ПЗ) необхідно дотримуватись наведених далі вимог ДСТУ 3008:2015. Текст пояснювальної записки повинен бути набраний на комп'ютері та роздрукований на принтері.

Шрифт і відступи

Текст пояснювальної записки повинен бути набраний у будь-якому текстовому редакторі шрифтом *Times New Roman* розміром 14 з інтервалом між рядками 1,15–1,5. Відступи: зліва – 2,5 см, справа – 1 см, решта – 1,5 см. Шрифт та міжрядковий інтервал у додатках можуть бути довільними, але такими, щоб можна було прочитати і зрозуміти.

Нумерація сторінок.

Сторінки повинні бути пронумеровані, починаючи зі змісту у правому верхньому кутку сторінки. Титульна сторінка, сторінки з анотацією, сторінка з індивідуальним завданням не нумеруються.

Нумерація додатків продовжує основну нумерацію.

Оформлення розділів і підрозділів

Структурними елементами основної частини пояснювальної записки є розділи, підрозділи, пункти, підпункти, переліки.

Розділ – головна ступінь поділу тексту, позначена номером і має заголовок. Підрозділ – частина розділу, позначена номером і має заголовок. Пункт (підпункт) – частина розділу чи підрозділу (пункту), позначена номером і може мати заголовок.

Розділи нумерують порядковими номерами (тільки арабськими числами) в межах всього документа (1, 2, і т. д.). Підрозділи нумерують в межах кожного розділу, пункти – в межах підрозділу і т. д. за формою (3.1, 3.2, 3.2.1, 3.2.2.1 і т. д.). Цифри, які вказують номер, не повинні виступати за абзац. Після номера крапку не ставлять, а пропускають один знак.

Кожен розділ рекомендується починати з нової сторінки. Заголовок розділу (заголовок 1-го рівня) записують посередині великими літерами з більш високою насиченістю.

Заголовки підрозділів, пунктів та підпунктів (при наявності заголовка) записують з абзацу малими літерами, починаючи з великої. Перед заголовком і після нього пропускають один рядок або роблять інтервал 18 пт. Заголовки 2-го рівня записують шрифтом з більш високою насиченістю.

Допускається розміщувати текст між заголовками розділу і підрозділу, між заголовками підрозділу і пункту.

Переліки

Переліки, за потреби, можуть бути наведені всередині пунктів або підпунктів. Перед переліком ставлять двокрапку. Перед кожною позицією переліку слід ставити малу літеру української абетки з дужкою, або, не нумеруючи – дефіс (перший рівень деталізації). Наприклад,

Основні проблеми такі:

- промислове шпигунство;
- крадіжка і копіювання;
- несанкціонована модифікація;
- піратство.

Для подальшої деталізації використовують арабські цифри з дужкою.

Оформлення таблиць

Таблицю розміщують симетрично до тексту після першого посилання на даній сторінці або на наступній, якщо на даній вона не уміщується і таким чином, щоб зручно було її розглядати без повороту або з поворотом на кут 90°. Таблиці у тексті пояснювальної записки набираються основним шрифтом, в деяких випадках розмір шрифту може бути зменшений до 10-12. Підписи таблиць розташовуються над таблицею з вказанням її номеру і назви, вирівнявши по лівому краю. Наприклад,

Таблиця 1.1 – Поняття ООП

Поняття	Опис
1. Клас	певна абстрактна сутність, наприклад, "Пес", "Кіт", "Автомобіль", "Ціна товару", що на програмному рівні представлена змінними (полями даних) та методами, що оперують над цими полями даних
2. Об'єкт	конкретний екземпляр класу. Наприклад, зелена "Тойота" вашого сусіда є екземпляром класу "Автомобіль". По суті, це клас, поля якого ініціалізовані і він завантажений у пам'ять комп'ютера. На основі одного класу, можна створити безліч об'єктів.

На всі таблиці мають бути посилання за формою “ ... в табл. 1.1 або в дужках по тексту (табл. 1.1). Посилання на раніше наведену таблицю дають зі скороченим словом ”дивись” (див. табл. 1) за ходом чи в кінці речення.

При перенесенні частин таблиці на інші сторінки, повторюють або продовжують найменування граф. Допускається виконувати нумерацію граф на початку таблиці і при перенесенні частин таблиці на наступні сторінки повторювати тільки нумерацію граф. У всіх випадках найменування (при його наявності) таблиці розміщують тільки над першою частиною, а над іншими частинами зліва пишуть “Продовження таблиці 1” без крапки в кінці, наприклад,

Продовження таблиці 1

2	3
3. Успадкування	утворення нових класів на основі інших
4. Інтерфейс і	посилальний тип даних. Інтерфейси схожі на класи, проте їхні поля даних є константами, а методи не реалізовані. Об'єкти на основі інтерфейсів не створюються, проте класи можуть реалізовувати певний інтерфейс. І через об'єктну змінну інтерфейсного типу можна викликати реалізації даних методів.

Оформлення рисунків

Розміщують рисунки в тексті або в додатках. В тексті ілюстрацію розміщують симетрично до тексту після першого посилання на неї або на наступній сторінці, якщо на даній вона не уміщується без повороту. На всі рисунки мають бути посилання за формою: “ ... на рис. 3.6”, або в дужках по тексту (рис. 3.6). Посилання на раніше наведений рисунок дають зі скороченим словом ”дивись” (див. рис. 3.4) за ходом чи в кінці речення.

Кожен рисунок повинен мати номер і підпис, розташовані під рисунком по центру. Крапку в кінці не ставлять, знак переносу не використовують. Якщо найменування рисунка довге, то його продовжують у наступному рядку підрисункового надпису, починаючи від найменування. Наприклад,

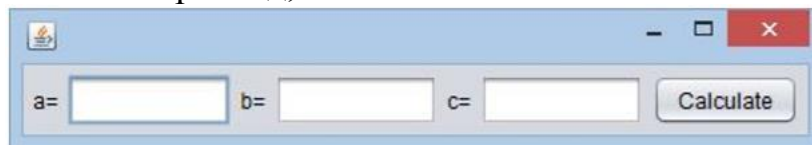


Рисунок 1.2 – Вікно визначення параметрів

Між ілюстрацією і текстом пропускають один рядок.

Нумерують ілюстрації в межах розділів, вказуючи номер розділу і порядковий номер ілюстрації в розділі, розділяючи крапкою. Дозволяється нумерувати в межах всього документа.

Оформлення формул

Кожну формулу записують з нового рядка, симетрично до тексту, курсивом. Між формулою і текстом пропускають один рядок.

Умовні літерні позначення (символи) в формулі повинні відповідати установленим. Їх пояснення наводять в тексті або зразу ж під формулою. Для цього після формули ставлять кому і записують пояснення до кожного символу з нового рядка в тій послідовності, в якій вони наведені у формулі, розділяючи крапкою з комою. Перший рядок повинен починатися з абзацу з слова “де” і без будь-якого знака після нього.

Всі формули нумерують в межах розділу арабськими числами. Номер вказують в круглих дужках з правої сторони, в кінці рядка, на рівні

закінчення формули. Номер формули складається з номера розділу і порядкового номера формули в розділі, розділених крапкою. Дозволяється виконувати нумерацію в межах всього документа.

2.2 Структура пояснювальної записки

Кожен етап виконання курсової роботи обов'язково має знайти своє відображення в пояснювальній записці. Пояснювальна записка повинна відповідати індивідуальному завданню, а її оформлення – чинним стандартам, які слід враховувати на момент виконання розробки з врахуванням всіх офіційних змін, введених в дію.

Пояснювальна записка до курсової роботи повинна мати таку структуру.

- вступну частину;
- основна частина;
- додатки;
- ілюстративна частина.

2.3 Вступна частина пояснювальної записки

Вступна частина повинна містити:

- титульний аркуш;
- індивідуальне завдання;
- анотації двома мовами;
- зміст.

Титульний аркуш

Титульний аркуш є першою сторінкою курсової роботи, яка не нумерується. Титульний аркуш виконується за встановленим зразком. Зразок титульного аркуша пропонується у додатку Б.

Система умовних позначень має таку структуру:

XX-XX.XXX.XXX.XX.XXX.XX
1 2 3 4 5 6

де 1 (XX-XX) – числовий шифр кафедри, прийнятий у ВНТУ;

2 (XXX) – умовне скорочення для дисципліни (ЗПЗ, КГКА, ПБД, ...);

3 (XXX) – перша цифра 0, якщо це проект або 1, якщо робота, друга і третя цифри означають рік, наприклад, 12 - 2012 рік;

4 (XX) – варіант завдання (наприклад, 01, 02, ..., 99);

5 (XXX) – перший символ – номер групи (1 або 2), наступні два символи задають номер студента за списком у журналі старости групи;

6 (XX) – код документа: ПЗ – пояснювальна записка.

Індивідуальне завдання

Конкретний зміст кожного курсової роботи, етапи його виконання визначає керівник на підставі індивідуального завдання, затвердженого

завідувачем кафедри.

Попередньо керівник видає індивідуальне завдання до курсової роботи. Індивідуальне завдання в перелік змісту не вноситься і має бути другою сторінкою після титульного аркуша. Зразок індивідуального завдання до курсової роботи наведено в додатку В.

Керівник курсової роботи пропонує перелік основних етапів реалізації задачі курсової роботи, який, як правило, є типовим і наводиться в розроблених методичних вказівках. Цей перелік повинен висвітлюватись в індивідуальному завданні. Крім того, в індивідуальному завданні повинні бути наведені вхідні і вихідні дані проекту.

В залежності від специфіки дисципліни керівник курсової роботи може пропонувати тему, яка підлягає конкретному обґрунтуванню та розробці специфічного індивідуального завдання. Індивідуальне завдання до курсової роботи має містити термін видачі, підписи керівника та студента.

Завдання на курсову роботу повинно бути підготовлено студентом не пізніше другого тижня з початку навчального семестру, підписано викладачем, що видав завдання і студентом, що прийняв його до виконання.

Анотація

Анотація призначена для ознайомлення з текстовим документом курсової роботи. Вона повинна коротко характеризувати мету роботи, засоби, використані для досягнення поставленої задачі, коротку інформацію про досягнуті результати. Розмір анотації повинен становити приблизно 1/3 частину сторінки.

Анотацію розміщують безпосередньо за аркушем з індивідуальним завданням, починаючи з нової сторінки, нумерація якої не зазначається. Заголовок (слово АНОТАЦІЯ) розміщується по центру сторінки, після нього пропускається один рядок.

Зміст

Зміст розташовують безпосередньо після анотації, починаючи з нової сторінки. До змісту включають: вступ; послідовно перелічені назви всіх розділів, підрозділів, пунктів і підпунктів (якщо вони мають заголовки) суті роботи; висновки; перелік використаних джерел; назви додатків і номери сторінок, які містять початок матеріалу.

Зміст не включає титульний лист, індивідуальне завдання на курсову роботу та анотацію. Нумерація у змісті починається зі ВСТУПУ (відповідно до нумерації у пояснювальній записці). Сам зміст за нумерацією пояснювальної записки є четвертою сторінкою. Нумерація сторінок повинна бути наскрізною, включаючи додатки.

Назви заголовків змісту повинні однозначно відповідати назвам заголовків пояснювальної записки за текстом. Формування змісту у текстовому документі бажано здійснювати автоматично, використовуючи

засоби обраного текстового редактора.

Приклад оформлення змісту:

ВСТУП

1 РОЗРОБКА ...

1.1 Варіанти ...

1.1.1 ...

.....

2 ЗАГОЛОВОК ДРУГОГО РОЗДІЛУ

2.1 Заголовки підрозділів

2.1.1 ...

.....

.....

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

Додаток А. Назва першого додатка

Додаток Б. Назва другого додатка

.....

2.4 Вміст і оформлення основної частини

Основна частина складається з таких складових:

- вступу;
- аналізу сучасного стану питання та обґрунтування теми;
- реалізації задачі курсової роботи;
- тестування розробки;
- висновків;
- переліку використаних джерел.

2.4.1 Вступ

Вступ пишуть з нової пронумерованої сторінки із заголовком «ВСТУП» посередині великими літерами.

Текст вступу повинен бути коротким і висвітлювати питання актуальності, значення, сучасний рівень і призначення курсової роботи. У вступі і далі за текстом не дозволяється використовувати скорочені слова, терміни, крім загальноприйнятих. Якщо ж в тексті є необхідність використовувати певні загальноприйняті скорочення (аббревіатури), то при введенні їх вперше в дужках слід вказати скорочення. І лише після цього дане скорочення можна використовувати по тексту. Наприклад, об'єктно-орієнтоване програмування (ООП). У назвах розділів, підрозділів, пунктів і підпунктів використовувати скорочення не рекомендується.

Вступ повинен стисло висвітлювати такі питання:

- стан розвитку проблеми в даній галузі, до якої має відношення розробка;

- галузь використання та призначення даної розробки;
 - *мету* та загальну постановку задачі;
 - *актуальність*, яка повинна подаватись в останньому абзаці вступу з метою стислого викладання суті обраної розробки.
- Обсяг вступу не повинен перевищувати 1-2 сторінок.

2.4.2 Аналіз сучасного стану питання та обґрунтування теми

Цей підрозділ є обов'язковим і передбачає огляд та дослідження відомих аналогів та існуючих програмних продуктів відповідної спрямованості тощо. Тут має бути наведений варіантний аналіз існуючих програм-аналогів, що використовуються для розв'язання аналогічних завдань, і здійснено короткий аналіз патентної науково-вітчизняної та зарубіжної літератури протягом декількох останніх років, електронних документів (електронних книг, Internet-сайтів, web-сторінок). Посилання на джерела по тексту пояснювальної записки є обов'язковими і наводяться у квадратних дужках, вказуючи порядковий номер за списком.

Рекомендований обсяг розділу – 4-5 сторінок.

2.4.3 Реалізація основної задачі курсової роботи

В даному розділі повинні бути детально описані кроки, які необхідно виконати для досягнення мети роботи, вимоги до виконуваної розробки, очікувані результати.

Таким чином, в цьому розділі повинні бути такі етапи:

1. Розробка інтерфейсу програми та прикладів використання. На цьому етапі створюється як загальний інтерфейс програми, так і приклади використання, які ілюструють основні способи взаємодії між об'єктами та користувачем.
2. Розробка діаграм класів, об'єктів та стану. На цьому етапі фіксуються результати декомпозиції предметної області на класи та об'єкти у вигляді діаграм наслідування, агрегації, композиції, взаємодії, послідовності та стану.
3. Розробка підсистеми графічного відображення. На цьому етапі створюються зображення мікро- та макро- об'єктів, при необхідності формуються графічні спец-ефекти.
4. Використання засобів програмування JavaFX. На цьому етапі розробляються специфічні питання синхронізації об'єктів, передачі повідомлень і т.і.
5. Розробка підсистеми серіалізації/десеріалізації даних. На цьому етапі розробляються способи збереження структури даних програми на зовнішніх носіях.
6. Керівництво користувача. На цьому етапі розробляється інструкції користувачу щодо того як встановлювати і запускати програму, як здійснювати керування об'єктами при роботі програми.

При викладенні тексту пояснювальної записки забороняється переписування матеріалів літературних джерел, сканування рисунків, які мають відношення до основної частини. Частина описового матеріалу (у вигляді таблиць, рисунків, графічної інформації) бажано виносити у додатки до пояснювальної записки, а у тексті посилатись на ці додатки.

У випадку, якщо результатом курсової роботи повинен бути програмний продукт, окремими підпунктами бажано описати розробку загальної схеми функціонування програми або схему роботи програми. Для окремих модулів, що входять до складу програмного засобу, необхідно розробити блок-схеми алгоритмів, структурні схеми з наведенням основних блоків, покрокове викладення алгоритму з розгалуженням і т. ін.

Наявність схем у пояснювальній записці є обов'язковою. Їх кількість і види повинні обиратися розробником таким чином, щоб вони допомагали зрозуміти роботу програми в цілому і її складових. Всі схеми повинні бути виконані згідно з вимогами чинних стандартів (підрозділ 3.6).

При описі програмної реалізації задачі проводиться покроковий опис реалізації основного алгоритму поставленої задачі. Описуються основні структури мови програмування, які використані в даній роботі. Наводяться фрагменти (5-10 операторів, а не програма цілком!). Описуються основні змінні, допоміжні модулі, класи, процедури і функції, основні і заголовні файли, які використовуються при розробці програмного забезпечення.

Якщо розроблено власні підпрограми або функції, вони повинні бути описані і мати посилання на ті додатки, де знаходиться лістинг програми. Якщо використовуються стандартні функції, навести доцільність та мету їх.

Даний розділ повинен бути одним з найконкретніших і змістовних і займати приблизно 2/3 обсягу пояснювальної записки (13-15 сторінок).

2.4.4 Використання схем і діаграм

Окремими підпунктами основного розділу або завершенням певних підрозділів бажано розробити і описати різні схеми (додаток Г):

- схеми даних;
- схеми програм;
- схеми роботи системи;
- схеми взаємодії програм;
- схеми ресурсів системи.

Розробник програми сам повинен вирішувати, які саме схеми доцільно розробляти у своїй роботі.

Схеми можуть бути виконані у вигляді UML-діаграм, серед яких можуть бути такі (додаток Є):

- діаграма варіантів використання;
- діаграма класів;

- діаграма станів;
- діаграма діяльності;
- діаграма послідовності;
- діаграма компонентів тощо.

Схеми і діаграми можуть використовуватися на різних рівнях деталізації, причому кількість рівнів залежить від розмірів і складності задачі оброблення даних.

Для побудови схем і діаграм можуть бути використані стандартні програмні засоби, спеціально призначені для цього або вбудовані засоби програмних середовищ, у яких здійснюється реалізація програм.

2.4.5 Тестування програми і розробка інструкцій

Цей розділ може бути як самостійним розділом, так і підрозділом основного розділу пояснювальної записки. Він повинен бути присвячений тестуванню розробленої програми на предмет коректності роботи програми, зручності подання матеріалів тощо, а також для формування інструкцій та рекомендацій щодо роботи з нею.

Тут необхідно продемонструвати весь хід виконання програми на всіх режимах її роботи. Рекомендується включати до пояснювальної записки підрозділ "Розробка тестових прикладів", в якому слід підготувати різні комплекти вхідних даних: правильні і такі, що призводять до помилок. При здійсненні і описі аналізу роботи програми можна наводити рисунки, що ілюструють роботу програми (screen-shorts) і демонструють основні режими і можливості її функціонування. Ілюстрації можна виносити у додатки, а в тексті пояснювальної записки посилається на ці додатки.

Якщо темою даного курсової роботи є розробка програмного засобу, то необхідно передбачити підготовку конкретних інструктивних матеріалів щодо роботи з програмою. Тобто, слід передбачити розробку рекомендацій щодо роботи з програмою: інструкцію програмісту, системному програмісту, оператору (користувачу) та інструкцію з технічного обслуговування. Інструкції для роботи з програмою виносяться у додатки.

Інструкція з технічного обслуговування

Інструкція з технічного обслуговування повинна містити такі підпункти :

- *вступ* (призначення інструкцій, перелік експлуатаційних документів, якими повинні додатково до інструкцій користуватися при технічному обслуговуванні і експлуатації);
- *загальні вказівки* (порядок технічного обслуговування, вказівки щодо організації і особливостей його проведення);
- *вимоги до технічних засобів* (вказують мінімальний склад технічних засобів, що забезпечують роботу програмного засобу);
- *опис функцій* (максимальний перелік функцій, що здійснюються цією

програмою; опис сумісного функціонування технічних засобів і програми з вказанням методу обробки помилок; опис організації вхідних і вихідних даних для перевірки роботоздатності; опис взаємодій пристроїв з програмою, результатів взаємодій, висновки з результатів роботи програми).

Інструкція системного програміста

Інструкція системного програміста повинна містити такі підпункти:

- *загальні відомості про програму* (призначення і функції програми і відомості про технічні і програмні засоби, що забезпечують виконання даної програми);
- *структуру програми* (відомості про структуру програмного продукту, її складові частини, про зв'язки між складовими частинами і про зв'язки з іншими програмами). Тут доречною є схема зв'язків між програмами;
- *настроювання програми* (опис дій щодо настроювання програми на умови конкретного застосування – настроювання на склад технічних засобів, вибір функцій і ін. При необхідності наводять пояснювальні приклади);
- *перевірку програми* (опис способів перевірки, що дозволяють дати загальний висновок про роботоздатність програми – контрольні приклади, методи прогону, результати);
- *повідомлення системному програмісту* (тут повинні бути вказані тексти повідомлень, які можуть з'являтися під час виконання настроювання системи захисту, перевірки правильності роботи програми, опис їх змісту і дій, які необхідно зробити у відповідь на ці повідомлення).

Інструкція програміста

Інструкція програміста повинна містити такі підпункти:

- *призначення і умови застосування програми* (тут слід вказати призначення і функції, виконувані програмою, умови, необхідні для виконання програми – об'єм оперативної пам'яті, вимоги до складу і параметрів пристроїв, вимоги до програмного забезпечення і т.п.);
- *характеристику програми* (опис основних характеристик і особливостей програми – тимчасові характеристики, режими роботи, засоби контролю правильності виконання і самовідновлення програми і т.п., середовище розробки, можливості внесення коректувань);
- *звернення до програми* (опис процедур виклику програми (способи передачі управління, параметрів і ін.);
- *вхідні і вихідні дані* (опис організації використовуваної вхідної і вихідної інформації і, при необхідності, її кодування);
- *повідомлення* (тексти повідомлень програмісту або оператору в ході виконання програми, опис їх змісту і дій, які слід зробити у відповідь на ці повідомлення).

Інструкція оператора

Інструкція оператора повинна містити такі:

- *призначення програми* (відомості про призначення програми і інформація, достатня для розуміння функцій програми і її експлуатації);
- *умови виконання програми* (умови, необхідні для роботи програми: мінімальний і/або максимальний склад апаратних і програмних засобів і т.п.);
- *виконання програми* (послідовність дій оператора, що забезпечують завантаження, запуск, виконання і завершення програми, опис функцій, формату і можливих варіантів команд, за допомогою яких оператор здійснює завантаження і управління ходом виконання програми);
- *повідомлення оператору* (тексти повідомлень в ході виконання програми, опис їх змісту і відповідні дії оператора: у разі збою, можливості повторного запуску програми і т.п.).

2.4.6 Висновки

Висновки оформляють з нової пронумерованої сторінки, починаючи зі слова “ВИСНОВКИ” посередині великими літерами, після чого пропускається один рядок. У висновках наводяться основні результати роботи над курсовою роботою. Коротко по основних розділах описуються етапи реалізації задачі курсової роботи. На основі проведених досліджень даються обґрунтовані висновки щодо виконаної об’єктно-класової декомпозиції, недоліки та переваги розробки, труднощі при розробці програми та причини, що їх обумовили, і можливі шляхи їх подолання. Обов’язково слід навести можливі рекомендації прикладного застосування, розробленого програмного засобу, та шляхи (перспективи) удосконалення розробленого програмного забезпечення.

2.4.7 Перелік використаних джерел

Перелік використаної літератури оформляють з нової пронумерованої сторінки, починаючи зі слів “ПЕРЕЛІК ПОСИЛАНЬ”, посередині великими літерами, після чого пропускається один рядок.

Перелік містить список використаних джерел, які було використано в процесі виконання роботи, і на які повинні бути обов’язкові посилання в тексті пояснювальної записки. Література (книги, статті, патенти, журнали, інтернет-сторінки) в загальний список записується в порядку посилання на неї в тексті. Посилання на літературу наводять в квадратних дужках [...], вказуючи порядковий номер за списком.

Кожне джерело повинно бути оформлено у відповідності до ДСТУ ГОСТ 7.1:2006 [9]. При цьому повинно бути вказано видавництво, рік видання, кількість сторінок. Літературу записують мовою оригіналу. У списку кожне джерело записують з абзацу, нумерують арабськими

цифрами, починаючи з одиниці (далі показано). Правильне оформлення певного джерела інформації можна переглянути у переліку літературних джерел у будь-якому навчальному посібнику. Якщо у списку використаних джерел є посилання на Інтернет-сторінки, слід наводити разом з назвою Інтернет-сторінки.

Приклад оформлення переліку використаних джерел різного характеру:

Приклад посилання на книги:

1. Мамаев М. Технология защиты информации в интернете [Специальный справочник] / Максим Мамаев, Сергей Петренко. - СПб. : Питер, 2002. – 848 с.
2. Савчин Володимир Павлович. Електронне перенесення в напівпровідникових структурах : навч. посіб. для студ. ВНЗ / В. П. Савчин, Р. Я. Шувар. – Львів : ВЦ ЛНУ ім. І. Франка, 2008. – 688 с. – ISBN 978-966-613-569-1

Приклад посилання на журнали:

4. Ершов А. А. Стабильные методы оценки параметров / А. А. Ершов, Б. Б. Петров // Автоматика и телемеханика. – 1978. – №8. – С. 86-91.

Приклад посилання на ГОСТ і ДСТУ:

5. Захист інформації. Технічний захист інформації. Основні поняття : ДСТУ 3396.0-96. – [Чинний від 1997—01—01]. – К. : Держспоживстандарт України, 1996. – 20 с. – (Національні стандарти України).

Приклад посилання на патенти:

6. Адаптивний метод ущільнення даних : патент 14709 : МКЗ 07-07 / Горін О. М., Волощенко О. В., Чуріп О. О. ; власник патенту Вінницький національний технічний університет. — № 200601173 ; заявл. 26.07.06 ; опубл. 10.08.07, Бюл. № 12 (кн. 2). — 2 с. : іл.

Приклад посилання на web-сторінки:

7. Windows Vista [Електронний ресурс]. – Режим доступу : URL : http://ru.wikipedia.org/wiki/Windows_Vista - Назва з екрану.
8. Новые технологии написания вирусов [Електронний ресурс]. – Режим доступу : URL : <http://bezpeka.com/ru/news/2008/10/30/virus-injection.html> - Назва з екрану.

2.5 Оформлення додатків

У додатках вміщують матеріал, який:

- є необхідним для повноти звіту, але включення його до основної частини звіту може змінити впорядковане й логічне уявлення про роботу;
- не може бути послідовно розміщений в основній частині звіту через великий обсяг або способи відтворення;

- може бути вилучений для широкого кола читачів, але є необхідним для фахівців даної галузі.

У додатки можуть бути включені:

- додаткові ілюстрації, схеми або таблиці;
- матеріали, які через великий обсяг, специфіку викладення або форму подання не можуть бути внесені до основної частини (оригінали фотографій, проміжні математичні доведення, лістинги дизасембльованого коду; протоколи випробувань; інструкції; методики; опис комп'ютерних програм, розроблених або використаних у процесі виконання роботи та ін.);
- інструкції по роботі з програмою.

Кожен додаток необхідно починати з нової сторінки, вказуючи зверху посередині рядка слово “Додаток” і через пропуск – його позначення. Додатки позначають послідовно великими українськими літерами, за винятком букв Г, Є, З, І, Ї, Й, О, ..., наприклад, Додаток А, Додаток Б і т. д. Якщо додатків більше ніж літер, то продовжують позначати арабськими цифрами. Дозволяється позначати додатки латинськими літерами, за винятком літер *I* і *O*.

Кожен додаток повинен мати тематичний (змістовний) заголовок, який записують посередині рядка малими літерами, починаючи з великої.

Сторінки додатків нумеруються, продовжуючи загальну нумерацію у пояснювальній записці.

Всі додатки включають у зміст, вказуючи номер, заголовок і сторінки, з яких вони починаються.

2.6 Оформлення ілюстративної частини

Ілюстративна частина призначена для того, щоб допомогти студенту вдало захистити свою роботу. В ілюстративний матеріал слід виносити усі ті матеріали, які допоможуть представити етапи розробки курсової роботи, основні схеми, алгоритми, математичну модель, фрагменти інтерфейсу розробленої програми.

Перед ілюстративною частиною у пояснювальну записку вставляється окремий аркуш, в центрі якого розташовується надпис «Ілюстративна частина», а далі слідує безпосередньо аркуші ілюстративного матеріалу. Кожний з них повинен бути оформлений у вигляді плакату: підпис матеріалу повинен бути вгорі по центру, далі по центру сутність плакату.

Аркуші ілюстративної частини не нумеруються і не вносяться у зміст пояснювальної записки.

3 ГРАФІК ВИКОНАННЯ КУРСОВОЇ РОБОТИ І ПОРЯДОК ЗАХИСТУ

3.1 Графік виконання

Рекомендується такий графік виконання курсової роботи, який враховує самостійну роботу студентів під час 2-го триместру (18 тижнів) для денної форми навчання та 6-го семестру для заочної форми навчання.

Зміст розділу	Термін виконання
Отримання завдання на курсова робота, розробка і оформлення індивідуального завдання	1-2 тижд.
Аналіз літературних джерел та сучасного стану проблеми, обґрунтування вибору мови об'єктно орієнтованого програмування	2-3 тижд.
Розробка структури програмного забезпечення: усвідомлення алгоритмів підзадач, виконання контрольних прикладів, розробка інтерфейсу, обґрунтування необхідності додаткових засобів, розробка структури вхідних і вихідних даних, підбір необхідних програмних засобів і т. д.	3-5 тижд.
Розробка програмного забезпечення і налагодження його: програмування та тестування основних процедур та функцій, програмна реалізація інтерфейсу, програмна реалізація роботи з файлами, з елементами керування, реалізація перевірки цілісності даних і т. д.	6-11 тижд.
Тестування розробки та виправлення виявлених недоліків. Підготовка контрольних прикладів, перевірка повноти реалізованого функціоналу	9 тижд.
Оформлення пояснювальної записки до курсової роботи, розробка рекомендацій щодо експлуатації розробленої програми	10-11 тижд.
Здача курсової роботи на попередню перевірку: демонстрація роботи програми та чернетки пояснювальної записки	12 тижд.
Корегування і доповнення (при необхідності) програми згідно із зауваженнями керівника курсової роботи, врахування і виправлення пояснювальної записки	13-16 тижд.
Захист курсової роботи	17-18 тижні

Готовність до захисту курсової роботи визначає керівник за результатами попередньої перевірки якості пояснювальної записки та дієздатності програми. Записка повинна бути здана керівнику на перевірку не менше, як за тиждень до визначеного терміну захисту роботи. Якщо робота виконана в повному обсязі і не має принципових помилок, керівник допускає студента до захисту. В іншому випадку робота повертається студенту на доопрацювання. Після позитивного висновку про готовність курсової роботи студент повинен захистити його перед комісією у складі двох викладачів, які призначені кафедрою.

3.2 Оцінювання виконання курсової роботи

Після позитивного висновку про готовність курсової роботи студент повинен захистити її перед комісією у складі двох викладачів, які призначені кафедрою.

Курсова робота оцінюється за 100-бальною шкалою:

Сума балів за всі види за КП	Оцінка ECTS	Оцінка за національною шкалою
90 – 100	A	відмінно
82 – 89	B	добре
74 – 81	C	
64 – 73	D	задовільно
60 – 63	E	
35 – 59	FX	незадовільно з можливістю повторного складання
0 – 34	F	незадовільно з обов'язковим повторним вивченням дисципліни

Розподіл бальної оцінки за виконання курсової роботи:

Пояснювальна записка	Практична реалізація	Ілюстративна частина	Інструкції	Захист роботи	Сума балів
до 20	до 45	до 10	10	до 15	100

Пояснювальна записка (ПЗ) оцінюється за такими параметрами:

- дотримання вимог щодо оформлення ПЗ згідно ДСТУ 3008:2015;
- відповідність ПЗ і програми індивідуальному завданню;
- відповідність ПЗ програмному забезпеченню, що виноситься на захист;
- логічна пов'язаність розділів пояснювальної записки.

Програмна частина оцінюється за такими параметрами:

- виконання основної задачі курсової роботи;
- зрозумілість і зручність інтерфейсу програми;
- здійснення захисту як такого, сумісність зі штатним програмним забезпеченням;
- доведення повноти реалізованого функціоналу програми за допомогою даної розробки;
- наявність у додатках інструкцій по роботі з програмою і їх відповідність програмному забезпеченню.

Захист роботи оцінюється за такими параметрами:

- демонстрація працездатного програмного засобу або інших результатів розробки;
- вміння пояснити основні моменти розробки;
- обґрунтованість відповідей на запитання членів комісії;
- представлення розробки (презентація, ілюстративний матеріал тощо).

ПЕРЕЛІК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Шилдт Герберт. Java. Полное руководство, 8-е изд. / Шилдт Герберт - М.: ООО "И.Д. Вильямс", 2012. - 1104 с.
2. Шилдт Герберт. Swing: руководство для начинающих / Шилдт Герберт - М.: ООО "И.Д. Вильямс", 2007. - 704 с.
3. Эккель Брюс Философия Java. Библиотека программиста, 4-е изд / Эккель Брюс - СПб.: Питер, 2009. - 640 с.
4. Хорстманн Кей С., Java. Библиотека профессионала / Том 1. Основы. 9-е изд. / Хорстманн Кей С., Корнелл Гари - М.: ООО "И.Д. Вильямс", 2014. - 864 с.
5. Хорстманн Кей С., Java. Библиотека профессионала / Том 2. Расширенные средства. 9-е изд. / Хорстманн Кей С., Корнелл Гари - М.: ООО "И.Д.Вильямс", 2014. - 1008 с.
6. Шилдт Г. Java 8: руководство для начинающих /Шилдт Г. – СПб.: Вильямс, 2015.- 712с.
- 7.Арлоу, Д. UML 2 и унифицированный процесс / Арлоу Д., Нейштадт А. – СПб.: Символ, 2007. - 624с.
8. UML Profile for XSD [Електронний ресурс]. – Режим доступу: http://www.sparxsystems.com/uml_tool_guide/xml_technologies/uml_profile_for_xs_d.htm
9. ДСТУ ГОСТ 7.1:2006. Система стандартів з інформації, бібліотечної та видавничої справи. Бібліографічний запис. Бібліографічний опис. Загальні вимоги та правила складання / Нац. стандарт України. – Вид.офіц. – [Чинний від 2007-07-01]. – Київ : Держспоживстандарт України, 2007. – 47 с.

Додаток А

Варіанти завдань на курсову роботу

1. Особиста бібліотека. Картотека домашньої бібліотеки: вихідні дані книги (Автори, назва, видавництво і т.д.), розділ бібліотеки (спеціальна література, хобі, домашнє господарство, белетристика і т.д.), походження і наявність книги в даний момент, суб'єктивна оцінка книги. Вибір книг по безпідставному замовленню; інвентаризація бібліотеки.

2. Картотека Інтерполу. Дані по кожному зареєстрованому злочинцю: прізвище, ім'я, кличка, зріст, колір волосся і очей, особливі прикмети, громадянство, місце і дата народження, останнє місце проживання, знання Зиков, злочинна професія, остання справа і т.д. Злочинні і мафіозні угруповання (дані про співучасників). Вибір по будь-якій підмножині ознак. Перенесення «Зав'язали в архів; видалення лише після смерті.

3. Бюро знайомств. База потенційних наречених: стаття, реєстраційний номер, дата реєстрації, відомості про себе, вимоги до партнера. Вибір підмножини підходящих кандидатур, підготовка зустрічей (формування запрошення для знайомства). Перенесення в архів пар, які вирішили свої сімейні проблеми, видалення клієнтів, що відмовилися від послуг.

4. Біржа праці. База безробітних: анкетні дані, професія, освіта, місце і посада останньої роботи, причина звільнення, сімейний стан, житлові умови, контактні координати, вимоги до майбутньої роботи. База вакансій: фірма, посада, умови праці та оплати, житлові умови, вимоги до фахівця. Пошук і реєстрація варіантів з того та іншого боку, формування оголошень для друку, видалення в архів після працевлаштування, повне видалення при відмові від послуг.

5. Записна книжка. Анкетні дані, адреси, телефони, місце роботи або навчання, посаду знайомих, колег і родичів, характер знайомства, ділові

якості і т.д. Автоматичне формування поздоровлення з днем народження (за поточної дати). Впорядкування за алфавітом і за датою останнього коригування. Пошук по довільному шаблону.

6. Каса Аерофлоту. Розклад: номер рейсу, маршрут, пункти проміжної посадки, час відправлення, дні польоту. Кількість вільних місць на кожному рейсі. Вибір найближчого рейсу до заданого пункту (при наявності вільних місць), оформлення заданого числа квитків за погодженням з пасажиром (з зменшенням числа вільних місць), оформлення посадкової відомості.

7. Довідник споживача (служба побуту). База підприємств побутового обслуговування міста: назва, розряд, адреса та телефони, спеціалізація, перелік надаваних послуг, форма власності, години і дні роботи. Пошук підприємств по заданій послугі та іншими ознаками.

8. Довідник покупця. Бази торгових точок міста: назва, адреса і телефони, спеціалізація, форма власності, час роботи. Вибір магазинів за довільним шаблоном.

9. Магазин з одним продавцем. Комп'ютер замість касового апарату. База наявності товарів: найменування, одиниця виміру, ціна одиниці, кількість, дата останнього завозу. Реєстрація надходження товару (як старих, так і нових найменувань). Оформлення покупки: виписка чека, коректування бази. Проблема уцінки і списання. Інвентаризація залишків товару з обчисленням сумарної вартості.

10. Відділ кадрів. База даних про співробітників фірми: паспортні дані, освіта, спеціальність, підрозділ, посада, оклад, дати надходження у фірму і останнього призначення і т.д. Вибір за довільним шаблоном. Скорочення штатів: вибір для звільнення осіб пенсійного та передпенсійного віку, підготовка наказу.

11. Генеалогічне дерево. Паспортні дані членів деякого родового клану; посилення на дітей (або на батьків). Пошук всіх нащадків або всіх

предків для вказаної особи.

12. Склад. База товарів, що зберігаються на складі: найменування, одиниця виміру, ціна одиниці, кількість, дата останнього завозу. Реєстрація надходження товару (формування прибуткової накладної) і відвантаження (видаткова накладна). Висновок інвентарної відомості.

13. Каса автовокзалу. Розклад автобусів: номер рейсу, кінцевий і проміжний пункти, час відправлення. Кількість вільних місць на кожному рейсі. Вибір найближчого рейсу до заданого пункту (при наявності вільних місць), оформлення квитків, оформлення посадкової відомості. Попередній продаж, повернення квитків.

14. Адміністратор готелю. Список номерів: клас, число місць. Список гостей: паспортні дані, дати приїзду і від'їзду, номер. Поселення гостей: вибір відповідного номера (за наявності вільних місць), реєстрація, оформлення квитанції. Від'їзд: вибір всіх постояльців, що від'їжджають сьогодні, звільнення місця або оформлення затримки з випискою додаткової квитанції. Можливість дострокового від'їзду з перерахунком. Пошук гостя по довільною ознакою.

15. Довідник меломана. База груп і виконавців: база пісень: база дисків з переліком пісень (у вигляді посилань). Вибір всіх пісень заданої групи; всіх дисків, де зустрічається задана пісня.

16. Щоденник. База намічуваних заходів - дата, час і протяжність, місце проведення. Автоматичне нагадування найближчої справи: за поточною датою і часу: видалення вчорашніх справ або перенесення на майбутнє. Аналіз "накладок" - перетинань планованих справ. Перегляд справ на завтра, післязавтра і т.д.

17. Термінологія. База визначень будь-якої науки: вводиться термін, його тлумачення (визначення), посилання на використовувані терміни. Можливість перегляду всього ланцюжка від заданого терміна до первинних понять.

18. Шеф-кухар. База рецептур страв: розкладка, рецепт приготування. База продуктів на складі: найменування, ціна, кількість. Формування меню на день (на задане число осіб); звану вечерю. Перевірка достатності запасів: формування видаткової накладної на склад, коригування запасів.

19. Довідник лікаря. База хвороб: назва, симптоми, процедури, перелік рекомендованих ліків із зазначенням необхідної кількості. База медикаментів на складі: назва, кількість, взаємозамінність. Формування рецепта після огляду хворого, перевірка наявності ліків, коригування запасів.

21. Довідник фірм. Назва, адреса та телефони, перша особа, статус (форма власності), сировина, продукція. Вибір за довільним шаблоном.

22. Обмін житла. База пропозицій по обміну: район, площа, планування і т.д.; вимоги до варіантів обміну. Реєстрація клієнтів, вибір відповідних варіантів, видалення при відбулося обміні або відмову.

23. Справочник абитуриента. База вузов; наименование, адрес, перечень специальностей, конкурс прошлого года по каждой специальности (дневной, вечерней, заочной форм), размер оплаты при договорном обучении. Выбор по разным критериям: все о данном вузе; все о данной специальности, поиск минимального конкурса по данной специальности или вообще. 20. Зарахування абітурієнтів. База абітурієнтів; анкетні дані, сукупність оцінок на вступних іспитах, готовність вчитися на договірній основі. Вибір для зарахування заданої кількості абітурієнтів; формування для співбесіди списку тих, хто набрав граничний прохідний бал, але не може платити за освіту. 24. Довідник поштової індексації. Республіка, область (край), район, населений пункт, поштовий індекс. Пошук по будь-якої сукупності полів (крім останнього); ієрархічна зв'язок між полями (зверніть увагу, що, наприклад, Новомосковськ є і в Тульській, і в Дніпропетровській областях).

25. Ощадбанк. Відомості про вкладників банку: номер особового рахунку, категорія вкладу, паспортні дані, поточна сума внеску, дата останньої операції. Операції прийому і видачі будь-якої суми, автоматичне нарахування відсотків. 26. Ломбард. База збережених товарів та нерухомості: анкетні дані клієнта, найменування товару, оціночна вартість; сума, видана під заставу, дата здачі, термін зберігання. Операції прийому товару, повернення, продажу - по закінченню терміну зберігання.

27. Довідник селекціонера. Найменування сорту будь-якої культури, автор, батьківські сорти, врожайність, характеристики плодів, морозостійкість, стійкість до шкідників і хвороб, наявність в тому чи іншому фонді. Вибір сортів, що володіють заданими властивостями.

28. Довідник власника відеотеки. База відеофільмів: назва, студія, жанр, рік випуску, режисер, виконавці головних ролей, короткий зміст, суб'єктна оцінка фільму. Факт наявності фільму у відеотеці. Оформлення видачі та повернення касети.

29. Купи - продай. База продавців: найменування товару, обсяг партії при оптовому продажі, ціна, умови продажу - відвантаження, форма оплати, контактну адресу або телефон, примітка (наприклад, «посередників прошу, не турбується»). База покупців: найменування товару, обсяг покупки, прийнятна ціна і форма оплати, контактну адресу або телефон, примітка. Пошук і реєстрація варіантів з тієї та іншої сторони; формування оголошень для друку, видалення в архів після купівлі - продажу (можливо, один із клієнтів залишається незадоволеним), повне вилучення при відмові від послуг.

30. Довідник фаната. База спортсменів: анкетні і антропологічні дані, громадянство, походження, вид спорту, клуб або команда, дані про особовий рекорді або перемоги і так далі. Вибір за довільним ознакою. Пошук рекордсмена в заданому виді спорту.

31. Довідник радіоаматора. Бази довідкових даних транзисторів,

діодів, тиристорів і так далі: марка, характеристики, гранично допустимі умови експлуатації, ціна, облік взаємозамінності і т.д. Підбірка по заданим вимогам.

32. Довідник комерційних банків. Найменування, адресу, статус (форма власності), умови зберігання коштів на особовому рахунку (річні відсотки на різних видів вкладів). Вибір банку з найбільшим відсотком для заданого типу вкладів.

33. Довідник начальника в'язниці. Анкетні дані ув'язнених, стаття, термін, дата взяття під варту, місце в тюремній ієрархії, камера, відомості про родичів, особливості характеру. Формування статистичних зведень про склад, вибір за довільною ознакою.

34. Довідник командира. Список підлеглих військовослужбовців: анкетні дані, адресу батьків, цивільна професія, освіта, звання та дата його отримання, посада, підрозділ, форма служби, особливості характеру і ставлення до служби. Формування списків: заданого підрозділи, офіцерського складу, новобранців і т.д. 35. Ріелтерська контора (купівля-продаж житла). База пропозицій: район та адреса, характеристика будинку і квартири, запитувана вартість, координати заявника. База попиту: вимоги покупця до житла (можливо кілька варіантів, допустимі діапазони), фінансові можливості, координати заявника. Підбір варіантів для того чи іншого боку, автоматизований пошук взаємоприйнятних варіантів. Приклад запиту покупця: однокімнатна, до 100 тис. руб., Інкерман не пропонувати.

36. Черга на житло. Список черговиків на отримання і поліпшення житла: дата надходження у фірму, дата подачі заяви, склад сім'ї, житлові умови на дату подання заяви, пільги на додаткову площу, позачергове і першочергове поліпшення, побажання на район і ін Реалізація розподілу одержуваного муніципального житла: задоволення підходящих черговиків, переселення наступних в звільнилися квартири. Додавання заявників та

видалення вибулих і задоволених. 37. Автосалон. База нових і старих вітчизняних та іноземних автомобілів: марка, рік випуску, технічні характеристики, особливості виконання, технічний стан, запитувана ціна. База покупців: контактні координати, вимоги до марки, технічним характеристикам і технічним станом, фінансові можливості. Автоматизація підбору варіантів для покупця, формування заявки для постачальників і перегонників.

38. Довідник туриста. Туристичні агенції та пропоновані послуги: країна, місто (або маршрут круїзу), умови проживання та проїзду, екскурсійне обслуговування, сервіс прийнятої боку, вартість путівки.

39. «Купи-продай». База оголошень (радіо-, газетних, в біжучому рядку) по всіх рубриках. Пошук для ... (наприклад, «репетиторство & математика»).

40. Крилаті фрази. Довідник прислів'їв, приказок, афоризмів, каламбурів, інших словесних курйозів. Класифікація за авторам і джерелам, пошук по темах і ключовим словам.

41. Каталог запчастин автомобіля. В автомобілі налічується кілька тисяч деталей; деякі використовуються в різних марках. Таблиці: країна, фірма-виробник, марка автомобіля, агрегат, вузол, деталь. Облік взаємозамінності. Користувачі: працівники автосервісу, магазинів запчастин; постачальники-оптовики. 42. Каталог радіодеталей (довідник телевізійного обладнання). Моделі побутової аудіо-, та відеотехніки; для кожної моделі - каталог радіодеталей, використаних в ній (резистори, конденсатори, котушки індуктивності, чіпи тощо). Багато деталей використовуються в різних моделях; деякі деталі однобічно (наприклад, замість резистора 10К 0,25 Вт, але не навпаки) або двосторонне (транзистори з близькими характеристиками) замінювані. Облік наявності деталей на складі, пошук відповідних деталей для ремонту.

43. Довідник нумізмата. Монети: країна, номінал, рік випуску,

кількість випущених монет, особливості. Колекціонери: країна, ім'я, контактні координати, наявність рідкісних монет в колекції. Власна колекція.

44. Довідник філателіста. Марки: країна, національна вартість, рік випуску, тираж, особливості. Філателісти: країна, ім'я, контактні координати, наявність рідкісних марок в колекції. Власна колекція.

45. Довідник любителя живопису. Художники з анкетними даними та стилями. Картини з посиланням на художників, датою створення, жанром. Колекціонери та музеї: наявність оригіналів картин і копій. Аукціонери і комісійки: дата проведення, список виставлених шедеврів і ціни на них. Власна колекція.

46. Так-тіклі. Ця гра схожа на шашки і на хрестики нулики. Ігрове поле - це 16 клітин (4x4). Кожен з гравців ставить свої чотири фішки по обидва боки поля, через одну з фішками противника, як показано на малюнку. За один хід можна пересунути свою фішку на одну вільну клітину вгору або вниз, вправо або вліво, але не по діагоналі. Знімати фішки супротивника не можна. Завдання полягає в тому, щоб розташувати три фішки свого кольору в один ряд по вертикалі, горизонталі або діагоналі швидше противника.

47. Хід конем. Мета гри: набрати більше очок, ніж суперник. Правила гри: Гравці по черзі рухають коня. Напрямок ходу довільно, але має відповідати правилам руху шахового коня. До суми очок гравця додається число, вказане на поле, на яке він поставив фігуру, і право ходу передається супротивникові. Гравець ходить з тієї клітини, на яку фігуру поставив противник.

48. Хід слоном. Мета гри: набрати більше очок, ніж суперник. Правила гри: Гравці по черзі рухають слона. Напрямок ходу довільно, але має відповідати правилам руху шахового слона. До суми очок гравця додається число, вказане на поле, на яке він поставив фігуру, і право ходу

передається супротивникові. Гравець ходить з тієї клітини, на яку фігуру поставив противник.

49. Хід королеви. Мета гри: набрати більше очок, ніж суперник. Правила гри: Гравці по черзі рухають ферзя. Напрямок ходу довільно, але має відповідати правилам руху шахового ферзя. До суми очок гравця додається число, вказане на поле, на яке він поставив фігуру, і право ходу передається супротивникові. Гравець ходить з тієї клітини, на яку фігуру поставив противник.

50. Хід тури. Мета гри: набрати більше очок, ніж супротивник. Правила гри: Гравці по черзі рухають туру. Напрямок ходу довільно, але має відповідати правилам руху шахової тури. До суми очок гравця додається число, вказане на поле, на яке він поставив фігуру, і право ходу передається супротивникові. Гравець ходить з тієї клітини, на яку фігуру поставив противник.

51. Хід амазонки. Мета гри: набрати більше очок, ніж суперник. Правила гри: Амазонка - уявна <шахова> фігура, яка об'єднує властивості ферзя і коня. Гравці по черзі рухають амазонку. Напрямок ходу довільно, але має відповідати зазначеним правилам. До суми очок гравця додається число, вказане на поле, на яке він поставив фігуру, і право ходу передається супротивникові. Гравець ходить з тієї клітини, на яку фігуру поставив противник.

52. Шибениця. Мета гри: вгадати слово, загадане комп'ютером. Правила гри: Можна вибрати будь-яку літеру. Якщо така буква є, всі аналогічні букви в слові будуть відкриті.

53. Решітка. Ігрове поле має розмір 10x10 з випадково розподіленими числами в них. Мета гри: Набрати більше очок, ніж супротивник. Правила гри: Перший гравець ходить по горизонталі, другий - по вертикалі. Перший гравець може вибрати будь-яку клітину з цифрою на поточному горизонталі. Якщо колір клітини червоний, то кількість очок, вказане на

ній, додається до рахунку гравця, якщо синій - віднімається. Після ходу першого гравця вертикаль, на якій знаходиться вибрана ним клітина, стає поточною і хід переходить до супротивника, який може вибрати будь-яку клітину з цифрою на поточному вертикалі і т.д.

54. Діагоналі. Ігрове поле має розмір 6x6. Мета гри: Зайняти якомога більше місця на дошці. Правила гри: Коли настає Ваша черга ходу, Ви можете поставити одну фішку на будь-яку клітину дошки. Якщо після Вашого ходу, один чи кілька фішок супротивника опинилися між вашими фішками, все етіфішки суперника переходять до Вас.

55. Double-dealing fellows. Ігрове поле має розмір 6x6. Мета гри: зайняти на дошці простір більше, ніж супротивник. Правила гри: Можна рухати яку свою фішку на наступну клітину дошки, при цьому фішка розділиться на двоє, тобто з'явиться в новій позиції і залишиться на старій одночасно. Також можна перестрибувати через одну клітку, але в цьому випадку <роздвоєння> фішки не відбувається. Якщо сусідні клітини зайняті суперником, вони переходять до Вас.

56. Сенегальські шашки. Розмір ігрового поля - 5x6 клітин. Кожен гравець отримує по 14 шашок (черепашки і каміння), які розставляються так, щоб заповнилися дві крайні верхні і нижні лінії шашками відповідного кольору. У центрі шашки розміщаються з країв, так щоб утворилося два порожніх поля, з яких починається гра. Рухатися можна тільки на порожнє поле попереду, праворуч або ліворуч, по одній клітці за хід. Не можна рухатися назад або по діагоналі. Якщо з'являється можливість перестрибнути через шашку супротивника, гравець зобов'язаний зробити це. У такому випадку, шашка супротивника вважається <убитою> і знімається з дошки. Гра закінчується, коли шашки одного з партнерів просунулися так далеко, що більше не можуть <з'їсти> фішок. Тоді підраховується кількість з'їдених шашок і визначається переможець.

57. Японські шашки. Ігрове поле має розмір 10x10. Мета гри:

поставити 5 фішок в ряд (по вертикалі, горизонталі, діагоналі). Правила гри: Фішку можна ставити на будь-який вільний місце ігрового поля.

58. Куточки. Ігрове поле має розмір 7x7. Спочатку фішки розташовуються двома блоками 3x3 в протилежних кутках дошки. Мета гри - перемістити всі фішки в протилежний кут. Шашку можна пересувати двома способами: на одну клітку вліво, вправо, вгору або вниз, або перестрибувати через інші фішки.

59. Блокада. Ігрове поле має розмір 7x7. Спочатку фішки розташовуються двома блоками 3x3 в протилежних кутках дошки. Мета гри: пересунути всі шашки в протилежний кут швидше противника. Правила гри: Через порожні сусідні поля можна просувати фішку до трьох клітин за хід. Можна перестрибувати, через фішку. У цьому випадку фішка, через яку гравець перестрибнув, відправляється <в дім> - в кут, з якого вона почала свій рух. Кожен гравець може зробити не більше 10 стрибків за гру.

Зауваження. Допускається індивідуальний вибір теми. У цьому випадку необхідно узгодити тему з викладачем.

Додаток Б

Приклад оформлення титульного аркуша

Міністерство освіти і науки України
Вінницький національний технічний університет
Кафедра програмного забезпечення

КУРСОВА РОБОТА

з дисципліни ” **Об’єктно-орієнтоване програмування** ”

на тему: РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ ДЛЯ МОДЕЛЮВАННЯ

ОБ’ЄКТІВ ” МУЗИКАНТИ, ГОТЕЛІ, МІСТО ”

З ВИКОРИСТАННЯМ UML ТА МОВИ ПРОГРАМУВАННЯ C#

Студента (ки) _____ курсу _ групи _ПІ-__б
спеціальності 121 Інженерія програмного
забезпечення” Копейкіна А.І.

Керівник: доцент кафедри ПЗ, к.т.н., Д.І. Кательніков

Національна шкала _____

Кількість балів: _____ Оцінка: ECTS _____

Члени комісії

(підпис)

(прізвище та ініціали)

(підпис)

(прізвище та ініціали)

м. Вінниця - 2018 рік

Додаток В

Приклад оформлення індивідуального завдання

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії

ЗАТВЕРДЖУЮ
Зав. кафедри ПЗ, проф., д.т.н.

А.М. Петух

(підпис)

” ” _____ р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на курсову роботу з дисципліни ”Об'єктно-орієнтоване програмування”

студенту групи ІПІ-176 Копейкіну А.І.

РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ ДЛЯ МОДЕЛЮВАННЯ ОБ'ЄКТІВ ” МУЗИКАНТИ, ГОТЕЛІ, МІСТО ” З ВИКОРИСТАННЯМ UML ТА МОВИ ПРОГРАМУВАННЯ C#

Вихідні дані:

- MS Visual Studio 2015, довідники з мови C# та Windows Forms.

Вимоги до курсової роботи:

В програмі повинно бути реалізовано: додаток Windows з графічним інтерфейсом. Клас макрооб'єкта повинен містити не менше 4 елементів змінних і не менше 4 методів. Як мінімум одна змінна повинна бути типу int, одна – типу double і як мінімум одна – рядок (будь якої реалізації). При використанні мови C++ створити деструктор. Додати до класу макрооб'єкта в якості елемента масива або посилання на нескладний службовий об'єкт, щоб зробити необхідним глибоке копіювання. При використанні мови C++ реалізувати глибоке копіювання шляхом створення копіювального конструктора і перевантаження оператора=. При використанні мови C# реалізувати глибоке копіювання шляхом реалізації інтерфейсу IClosable. При використанні мови Java реалізувати глибоке копіювання шляхом реалізації інтерфейсу Cloneable. Повинне бути продемонстроване використання глибокого копіювання. При використанні мов C# C++ перевантаження використання перевантаженого унарного та бінарного операторів. При використанні мови C# продемонструвати використання як мінімум двох функцій-статичних елементів класу Array. При використанні мови Java продемонструвати використання як мінімум двох функцій-статичних елементів класу Arrays. При використанні мови C# продемонструвати використання інтерфейсів Comparable/Comparable та IClosable. При використанні мови Java продемонструвати використання інтерфейсів Comparable/Comparable та Cloneable. Макрооб'єкт: три графічних примітива (один з яких - текст). Більший макрооб'єкт: три графічних примітива(один з яких - текст). Менший макрооб'єкт: три графічних примітива(один з яких - текст). При натисканні двох кнопок миші на макрооб'єкт він повинен ставати активним/неактивним. При натисканні клавіш-стрілок активні об'єкти/об'єкти повинні рухатись у вікні програми. При натисканні клавіш Delete активні об'єкти повинні зникати. Якщо активного об'єкта нема – клавіша Delete ігнорується. Реалізувати в проєкті: Клавіша Esc повинна відмінати активацію об'єкта. Повина бути побудована ієрархія класів макрооб'єктів, яка містить як мінімум три рівня наслідування, які можуть інстанціюватись і зображатись на екрані. Зображення кожного рівня має відрізнятись від всіх інших хоча б одним графічним примітивом. При натисканні клавіш Insert повинно з'являтись діалогове вікно, яке повинно визначати параметри створюваного макрооб'єкта. В нього повинна бути додана можливість обирати до якого з класів нащадків у ієрархії наслідування він належить. Крім кружочкового елемента Win32 у діалоговому вікні також повинні бути використані як мінімум три з наступного списку: TextBox, CheckBox, ListBox, RadioButton. Повинна бути реалізована подвійна буферизація, яка повинна уможливити миттєвий зміни зображення у вікні програми. Повинно бути продемонстровано використання статичного поліморфізму. Повинно бути продемонстровано використання динамічного поліморфізму, упаковка/розпаковка (boxing/unboxing), типу доступу protected. Додати у проєкт клас/класи більшого макрооб'єкта. Більший макрооб'єкт призначений містити всі макрооб'єкти та всі менші макрооб'єкти. Графічне зображення більшого макрооб'єкта відображає лише певну його частину, яка не повинна перевищувати 25% його загального розміру. Мінімальний розмір більшого макрооб'єкта 1600 позицій для макрооб'єкта (тобто на екрані одночасно можна побачити не більше квадрата 20*20 позицій для макрооб'єкта). Менший макрооб'єкт повинен мати можливість містити більше одного макрооб'єкта. Не обов'язково мати можливість додавати/видаляти менші макрооб'єкти в програму. Цілоком достатньо, якщо програма буде мати деяку кількість менших макрооб'єктів створених при запуску програми і яка буде незмінною протягом роботи програми. Менші макрооб'єкти повинні мати певні позиції в більшому макрооб'єкті і повинні відображатись на екрані. Позиції менших макрооб'єктів в середині більшого макрооб'єкта можуть залишатись незмінними протягом роботи програми. В більшому макрооб'єкті повинно міститись більше одного екземпляра меншого макрооб'єкта. Макрооб'єкти можуть або належати одному (або більшій кількості) меншому макрооб'єкту, або не належати жодному. Повинна обов'язково бути реалізована можливість виводити макрооб'єкт з всіх менших макрооб'єктів, в результаті чого він не буде належати жодному меншому макрооб'єкту. Всі макрооб'єкти та менші макрооб'єкти повинні належати більшому макрооб'єкту і не повинні мати змоги вийти за його межі. Повинні бути реалізовані автоматичний рух деяких макрооб'єктів у програмі: А) при русі деякі макрооб'єкти взаємодіють між собою (це має бути помітно візуально в тому сенсі, що або поєднюються у їх зображенні або у характері їх руху); Б) деякі макрооб'єкти заходять в менші макрооб'єкти та деякі виходять.Вхарактер руху макрооб'єктів повинні змінюватись при натисканні певної клавіші клавіатури або при виборі команди меню. Повинні бути продемонстровано використання асинхронний виклик делегата. Повинні бути продемонстровано використання виклик multi-сая делегата. Повинна бути запрограмована серіалізація/де-серіалізація всіх об'єктів у файл. Серіалізація/де-серіалізація обов'язково повинна зберігати не тільки власне інформацію про стан макро- та мікро-об'єктів, але й про їх позицію на екрані. При серіалізації/де-серіалізації обов'язково повинні використовуватись діалогові вікна, щоб запитати у користувача ім'я файлу та його розташування на диску (наприклад можна використовувати системні діалогові вікна OpenFileDialog та SaveFileDialog). В програмі повинно бути продемонстровано використання інтерфейсів, створеного користувачем. В програмі повинно бути продемонстровано використання механізмів динамічного приведення типу: ключові слова as та is.

Зміст ПЗ до курсової роботи

Індивідуальне завдання

АНОТАЦІЯ

ВСТУП

1 АНАЛІЗ СУЧАСНОГО СТАНУ ПИТАННЯ ТА ОБГРУНТУВАННЯ ЗАВДАННЯ НА РОБОТУ

2 РОЗРОБКА ІНТЕРФЕЙСУ ПРОГРАМИ ТА ПРИКЛАДІВ ВИКОРИСТАННЯ

3 РОЗРОБКА ПІДСИСТЕМИ ГРАФІЧНОГО ВІДОБРАЖЕННЯ

4 РОЗРОБКА ДІАГРАМ КЛАСІВ, ОБ'ЄКТІВ ТА СТАНУ

5 ВИКОРИСТАННЯ ЗАСОБІВ ПРОГРАМУВАННЯ WINDOWS FORMS

6 РОЗРОБКА ПІДСИСТЕМИ СЕРІАЛІЗАЦІЇ/ДЕСЕРІАЛІЗАЦІЇ ДАНИХ

7 КЕРІВНИЦТВО КОРИСТУВАЧА

ВИСНОВКИ

ПЕРЕЛІК ПОСИЛАНЬ

Додатки (за необхідністю)

Дата видачі ” _____ ” _____ р. Керівник _____

(підпис)

Завдання отримав _____

(підпис)

Додаток Г

Використання схем алгоритмів, програм, даних і систем

Схеми алгоритмів, програм, даних і систем складаються з символів, що мають задане значення, короткого тексту пояснення і з'єднувальних ліній. Усі символи поділяються на такі підгрупи:

- символи даних (табл. Г.1);
- символи процесів (табл. Г.2). Приклад наведено на рис. Г.1.;
- спеціальні символи (табл. Г.3). Приклад наведено на рис. Г.2.;
- символи ліній (табл. Г.4). Приклад наведено на рис. Г.3.

Символи поділяються на:

- основні, для випадків, коли точний вигляд процесу або носія даних невідомий або відсутня необхідність в описі фактичного носія даних;
- специфічні, використовувані тоді, коли відомий точний вигляд процесу або носія даних або коли необхідно описати фактичний носій даних.

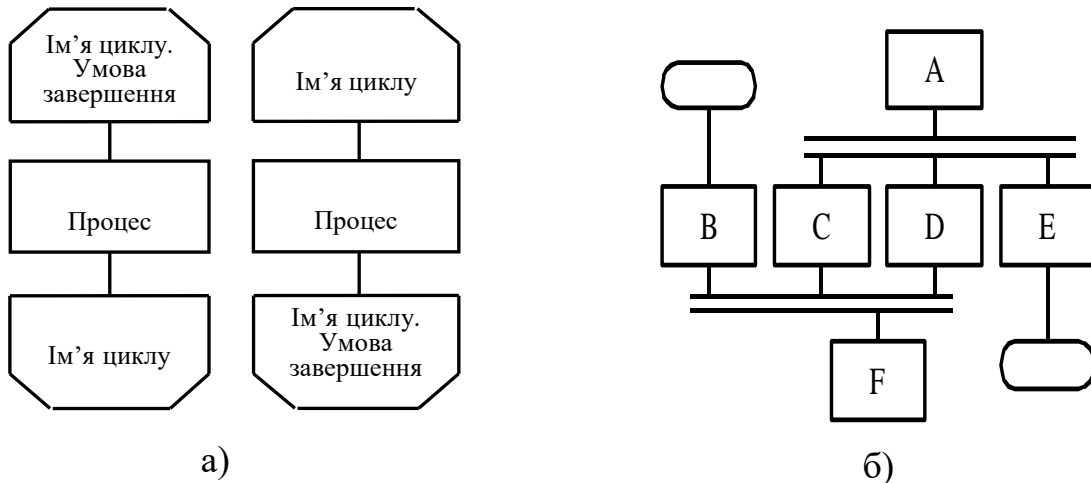


Рисунок Г.1 – Приклад застосування символів процесу (а – використання символів меж циклу; б – використання символів паралельних дій)



Рисунок Г.2 – Приклади використання спеціальних символів

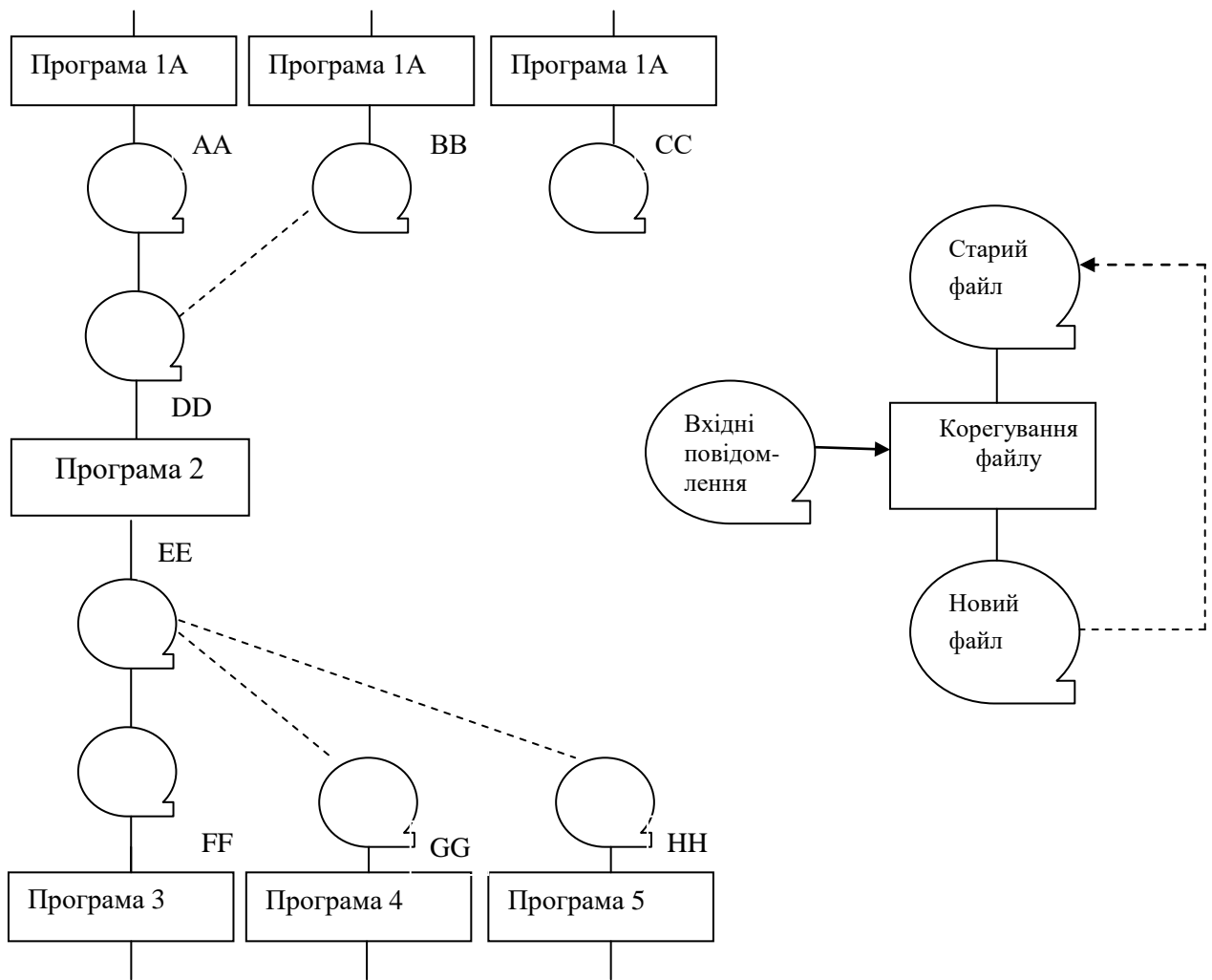


Рисунок Г.3 – Приклади використання символів ліній

Схеми даних відображають шлях даних при розв’язанні задач і визначають етапи обробки, а також різні використовувані носії даних. Схема даних складається з таких символів: символів даних, які можуть також вказувати вид носія даних; символів процесу, який виконується над даними (можуть також вказувати функції, виконувані обчислювальною машиною); символів ліній, які вказують потоки даних між процесами і (або) носіями даних; спеціальних символів для полегшення написання і читання схеми.

Символи даних чергуються з символами процесу. Схема даних починається і закінчується символами даних (за винятком спец. символів).

Схеми програм відображають послідовність операцій в програмі. Схема програми складається з: символів процесу, що вказують фактичні операції обробки даних (включаючи символи, що визначають шлях, якого слід дотримуватися з урахуванням логічних умов); лінійних символів, що вказують потік управління; спеціальних символів, використовуваних для полегшення написання і читання схеми.

Схеми роботи системи відображають управління операціями і потік даних в системі. Схема роботи системи складається з: символів даних, що вказують на наявність даних (символи даних можуть також вказувати на вид носія даних); символів процесу, що вказують операції, які слід виконати над даними, а також визначають логічний шлях, якого слід дотримуватися; лінійних символів, що вказують на потоки

даних між процесами і (або) носіями даних, а також потік управління між процесами; спеціальних символів, використовуваних для полегшення написання і читання блок-схеми.

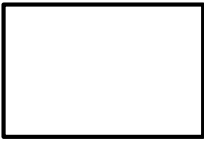



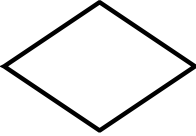

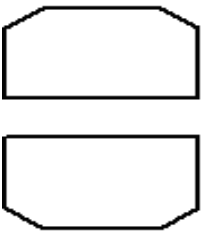
Схема взаємодії програм відображає шлях активацій програм і взаємодій з відповідними даними. Кожна програма в схемі взаємодії програм показується тільки один раз (у схемі роботи системи програма може відображатися більше, ніж в одному потоці управління). Схема взаємодії програм складається з: символів даних, що вказують на наявність даних; символів процесу, що вказують на операції, які виконують над даними; лінійних символів, що відображають потік між процесами і даними, а також ініціації процесів; спеціальних символів – для полегшення написання і читання схеми.

Схема ресурсів системи відображає конфігурацію блоків даних, блоків обробки цих даних, яка потрібна для розв'язання задачі або набору задач. Схема ресурсів системи складається з: символів даних, що відображають вхідні, вихідні і запам'ятовують пристрої обчислювальної машини; символів процесу, що відображають процесори, канали і т. д.); лінійних символів, що відображають передачу даних між пристроями введення-виведення і процесорами, а також передачу управління між процесорами; спеціальних символів, використовуваних для полегшення написання і читання схеми.

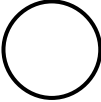

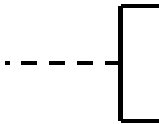


Таблиця Г.1 – Символи даних

<i>Основні символи даних</i>		
Дані		Символ відображає дані, носій даних невизначений
Дані, що запам'ятовуються		Символ відображає дані, що зберігаються, у вигляді, придатному для обробки, носій даних невизначений
<i>Специфічні символи даних</i>		
Оперативний запам'ятовувальний пристрій		Символ відображає дані, що зберігаються в оперативному запам'ятовувальному пристрої
Запам'ятовувальний пристрій з послідовним доступом		Символ відображає дані, що зберігаються в запам'ятовувальному пристрої з послідовним доступом (магнітна стрічка, касета з магнітною стрічкою, магнітофонна касета)
Запам'ятовувальний пристрій з прямим доступом		Символ відображає дані, що зберігаються в запам'ятовувальному пристрої з прямим доступом (магнітний диск тощо)
Документ		Символ відображає дані, подані на носії в легкій для читання формі (документ для оптичного або магнітного зчитування, мікрофільм, бланки даних)
Ручне введення		Символ відображає дані, що вводяться вручну під час оброблення з будь-яких пристроїв (клавіатура, перемикачі, кнопкисмужки зі штриховим кодом)
Дисплей		Символ відображає дані, подані у візуальній людиночитабельній формі на носії у вигляді пристрою відображення (екран для візуального спостереження, індикатори введення інформації)


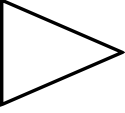


Таблиця Г. 2 – Символи процесу

<i>Основні символи, процесу</i>		
Процес		Символ відображає функцію обробки даних будь-якого вигляду (виконання певної операції або їх групи, що приводить до зміни значення, форми інформації).
<i>Специфічні символи процесу</i>		
Підпорядкований процес		Символ відображає підпорядкований процес, що складається з однієї або декількох операцій або кроків програми, які визначені у іншому місці
Ручна операція		Символ відображає будь-який процес, виконуваний людиною
Підготовка		Символ відображає модифікацію команди або групи команд з метою дії на деяку подальшу функцію (установлення перемикача, модифікація індексного реєстра або ініціалізація програми)
Умова або вибір		Символ відображає умову, вибір або функцію типу перемикача, що має один вхід і ряд альтернативних виходів, один і лише один з яких може бути активізований після обчислення умов, визначених усередині цього символу
Паралельні дії		Символ відображає синхронізацію двох або більше паралельних операцій
Межа циклу		Символ, що складається з двох частин, відображає початок і кінець циклу. Обидві частини символу мають один і той самий ідентифікатор. Умови для ініціалізації, прирости, завершення поміщаються усередині символу на початку або в кінці залежно від розташування операції, що перевіряє умову

Таблиця Г. 3 – Спеціальні символи

З'єднувач 	Символ відображає вихід в частину схеми і вхід з іншої частини цієї схеми і використовується для обривання лінії і продовження її у іншому місці. Відповідні символи-з'єднувачі повинні містити одне і те ж унікальне позначення
Термінатор 	Символ відображає вихід в зовнішнє середовище і вхід із зовнішнього середовища (початок або кінець схеми програми, зовнішнє використання і джерело або пункт призначення даних)
Коментар 	Символ використовують для додавання описових коментарів або записів пояснень з метою пояснення або приміток. Пунктирні лінії в символі коментаря пов'язані з відповідним символом або можуть окреслювати групу символів. Текст коментарів або приміток повинен бути поміщений біля обмежуючої фігури
Пропуск  	Символ (три крапки) використовують в схемах для відображення пропуску символу або групи символів, в яких не визначені ні тип, ні число символів. Символ використовують тільки в символах лінії або між ними. Він застосовується головним чином в схемах, що зображають загальні результати вибору з невідомим числом повторень

Таблиця Г. 4 – Символи ліній

<i>Основний символ ліній</i>	
Лінія 	Символ відображає потік даних або управління. У разі необхідності або для підвищення легкості читання можуть бути додані стрілки-покажчики
<i>Специфічні символи ліній</i>	
Передача управління 	Символ відображає безпосередню передачу управління від одного процесу до іншого, іноді з можливістю прямого повернення до ініціювального процесу після того, як ініційований процес завершить свої функції. Тип передачі управління повинен бути названий усередині символу (наприклад, запит, виклик, подія)
Канал зв'язку 	Символ відображає передачу даних по каналу зв'язку
Пунктирна лінія 	Символ відображає альтернативний зв'язок між двома або більшою кількістю символів, а також використовується для обведення ділянки

Додаток Д Приклади UML-діаграм

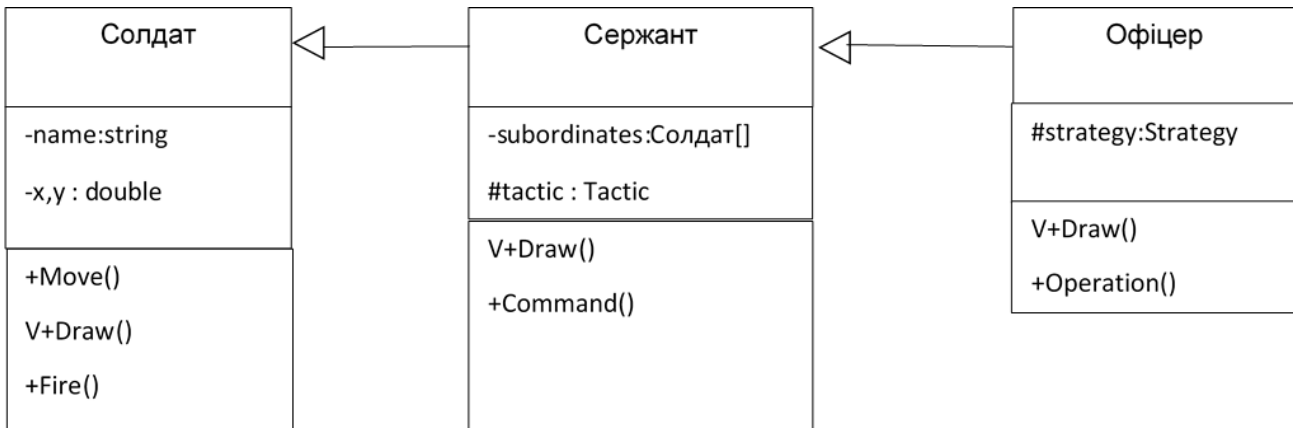


Рисунок Д.1 – Приклад діаграми класів

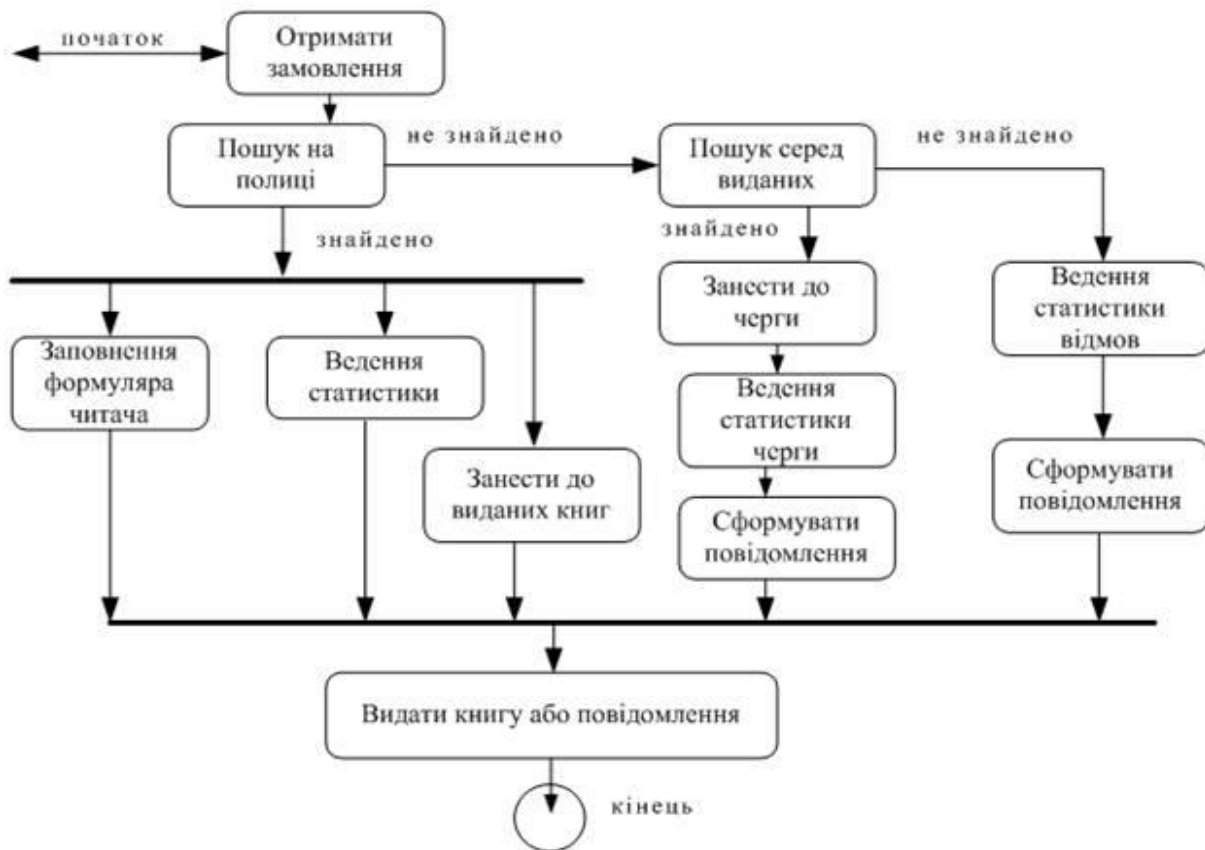


Рисунок Д.2 – Приклад діаграми діяльності

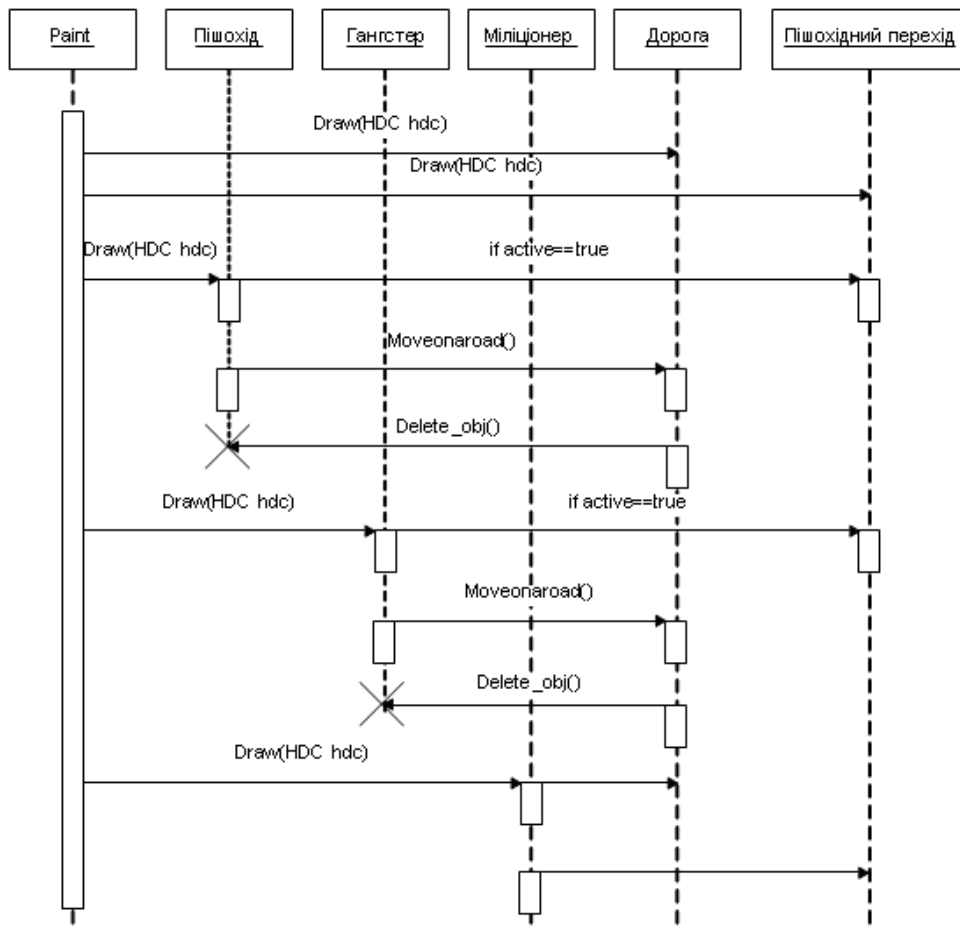


Рисунок Д.3 – Приклад діаграми послідовності

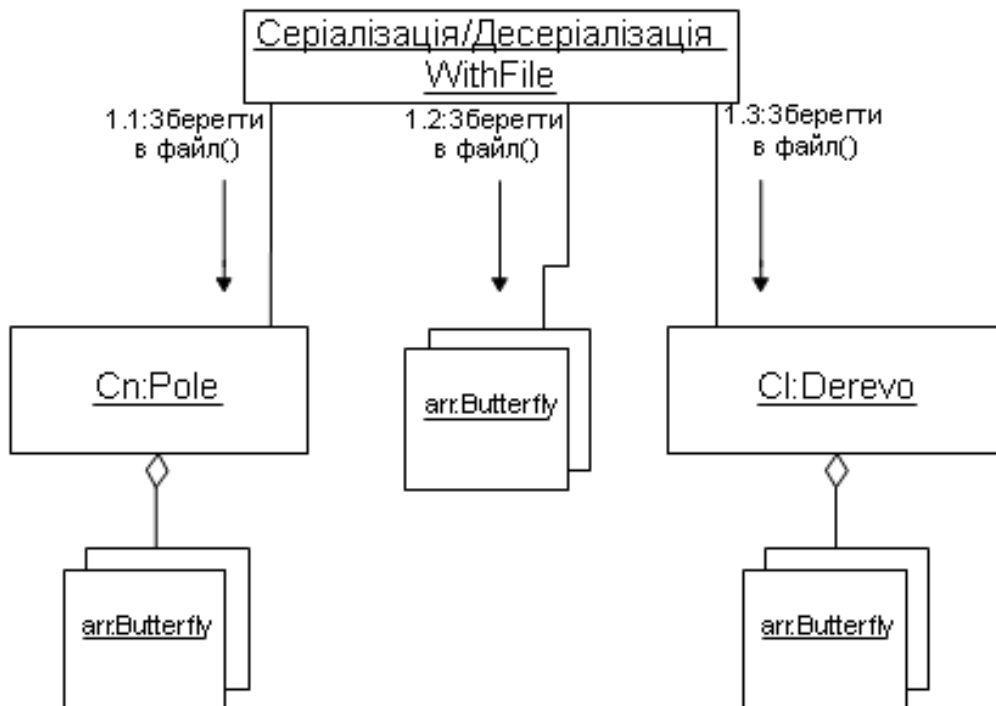


Рисунок Д.4 – Приклад діаграми кооперації

Навчальне видання

Денис Іванович Кательніков
Алла Василівна Денисюк

МЕТОДИЧНІ ВКАЗІВКИ
до виконання курсової роботи з дисципліни
"ОБ'ЄКТНО – ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ"
для студентів галузі знань 12 – Інформаційні технології
спеціальності 121 – Інженерія програмного забезпечення

Редактор В. Дружиніна

Укладачі: Денис Іванович Кательніков
Алла Василівна Денисюк

Оригінал-макет підготовлено Д.І. Кательніков

Підписано до друку
Формат 29,7×42¼. Папір офсетний.
Гарнітура Times New Roman.
Друк різнографічний. Ум.друк. арк.
Наклад ... пр. Зам. № 2016-

Вінницький національний технічний університет,
навчально-методичний відділ ВНТУ.
21021, м. Вінниця, Хмельницьке шосе, 95,
ВНТУ, к. 2201.
Тел. (0432) 59-87-36.
Свідоцтво суб'єкта видавничої справи
серія ДК № 3516 від 01.07.2009 р.

Віддруковано у Вінницькому національному технічному університеті
в комп'ютерному інформаційно-видавничому центрі.
21021, м. Вінниця, Хмельницьке шосе, 95,
ВНТУ, ГНК, к. 114.
Тел. (0432) 59-85-32,
publish.vntu.edu.ua; email: kivc.vntu@gmail.com.
Свідоцтво суб'єкта видавничої справи
серія ДК № 3516 від 01.07.2009 р.