

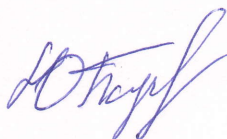
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ СЕМЕНА КУЗНЕЦЯ**

**ОБ'ЄКТНО-ОРІЄНТОВАНЕ
ПРОГРАМУВАННЯ**

**Методичні рекомендації
до виконання курсового проекту
для здобувачів вищої освіти спеціальності 121
"Інженерія програмного забезпечення" освітньої
програми "Інженерія програмного забезпечення"
першого (бакалаврського) рівня**

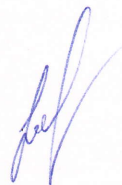
Укладачі: Парфьонов Ю.Е.



Гризун Л.Е.



Відповідальний за видання Бондаренко Д.О.



Харків
ХНЕУ ім. С. Кузнеця
2024

УДК 004.415 (07)

ББК 32.973-018р

Укладачі: Парфьонов Ю.Е.,
Гризун Л.Е.

Затверджено на засіданні кафедри інформаційних систем.
Протокол № 7 від 21.12.2023 р.

Самостійне електронне текстове мережеве видання

075 Об'єктно-орієнтоване програмування : методичні рекомендації до виконання курсового проекту для здобувачів вищої освіти спеціальності 121 "Інженерія програмного забезпечення" освітньої програми "Інженерія програмного забезпечення" першого (бакалаврського) рівня / укл. Парфьонов Ю.Е., Гризун Л.Е. – Х.: Вид. ХНЕУ ім. С. Кузнеця, 2024. - 40 с. (Укр. мов.)

Викладено питання організації курсового проектування, наведено вимоги до структури курсового проекту та оформлення пояснювальної записки, методичні рекомендації до розроблення його структурних елементів.

Рекомендовано для здобувачів вищої освіти спеціальності 121 "Інженерія програмного забезпечення".

Вступ

Сучасні умови господарювання вимагають від фахівців з економічного управління всебічного використання новітніх інформаційних технологій. Широкі можливості комп'ютеризованих засобів щодо збору, обробки та видачі необхідної інформації здатні значно підвищити якість економічних розрахунків, зробити більш ефективним процес обґрунтування економічних рішень.

Проте використання потужних комп'ютеризованих засобів неможливо без програмного забезпечення. Важливість галузі розроблення програмного забезпечення збільшується, оскільки тенденції розвитку комп'ютерної техніки свідчать про те, що з одного боку її складність та функціональні можливості швидко зростають, а з другого боку, це потребує більш досконалих програмних засобів для задоволення потреб користувачів.

Для створення таких програмних систем відповідні фахівці мають володіти як концепціями алгоритмізації та програмування узагалі, так і різними парадигмами програмування, зокрема, структурною, функціональною та об'єктно-орієнтованою. Це дозволяє їм розробляти програмні системи з використанням широкого кола мов програмування, оскільки більшість сучасних мов програмування тією чи іншою мірою допускають використання різних парадигм.

Необхідним елементом успішного засвоєння навчального матеріалу базової навчальної дисципліни «Об'єктно-орієнтоване програмування» є самостійна робота здобувачів із технічною літературою, сучасними мовами, технологіями та середовищами програмування. Підвищенню її ефективності покликане сприяти курсове проектування.

Курсовий проект є важливим етапом підготовки здобувачів. Він спрямований на більш глибоке осмислення і закріплення теоретичних знань, отриманих при вивченні вказаної дисципліни, та на вдосконалення практичних навичок щодо розроблення програмного забезпечення та необхідної документації.

Ці методичні рекомендації встановлюють єдині правила та порядок виконання, оформлення і захисту курсового проекту. Вони призначені для допомоги студентам спеціальності 121 "Інженерія програмного забезпечення" у виконанні курсового проекту.

1. Мета й завдання курсового проектування

Метою курсового проектування є закріплення та поглиблення знань з дисципліни «Об'єктно-орієнтоване програмування».

Робота над курсовим проектом сприяє систематизації, поглибленню й закріпленню знань, отриманих студентами під час вивчення цієї дисципліни. В процесі курсового проектування студенти розвивають навички практичного застосування отриманих знань шляхом розроблення застосунку з використанням сучасних інструментальних засобів розробки. При цьому студент повинен показати вміння користуватися спеціальною літературою, державними стандартами, довідниками та іншими матеріалами з інформаційних технологій.

Під час розроблення курсового проекту студент має показати знання:

- предметної області відповідно до постановки завдання;
- основних концепцій алгоритмізації та програмування;
- об'єктно-орієнтованого підходу до програмування;
- сучасного стану та перспектив розвитку технологій програмування;
- сучасних інструментальних засобів, призначених для розробки програмних систем.

Здобувач має уміти:

- виконувати постановку завдання та її аналіз;
- виконувати декомпозицію програмної системи на підсистеми;
- розробляти загальну архітектуру програмної системи;
- здійснювати деталізацію загальної архітектури програмної системи;
- розробляти програмну реалізацію системи певною мовою програмування (відповідно до робочої програми навчальної дисципліни «Об'єктно-орієнтоване програмування»);

- використовувати стандартні бібліотеки мови програмування або програмної платформи;

- документувати вихідний код програми;

- використовувати засоби розроблення програм та отримання довідкової інформації;

- розробляти та оформлювати необхідну документацію.

Робота над курсовим проектом певною мірою визначає загальнотеоретичну та спеціальну підготовку студента і в остаточному підсумку готує його до майбутнього виконання більш складного й завершального етапу навчального процесу – дипломного проектування. Студент має розглядати роботу над курсовим проектом як своєрідну «репетицію» дипломного проектування.

Результати навчання та компетентності, які формує освітня компонента визначено в табл. 1.

Таблиця 1

Результати навчання та компетентності, які формує курсове проектування

Результати навчання	Компетентності, якими повинен оволодіти здобувач вищої освіти
PH8	СК1, СК10, СК13
PH13	СК2, СК3

де, PH7. Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.

PH8. Вміти розробляти людино-машинний інтерфейс.

PH13. Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.

СК1. Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення.

СК2. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.

СК3. Здатність розробляти архітектури, модулі та компоненти програмних систем.

СК10. Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя.

СК13. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення.

2. Організація курсового проектування

Якісне виконання курсового проекту вимагає чіткої організації роботи здобувача з моменту вибору теми проекту й до його захисту.

Керівництво курсовим проектуванням здійснюється викладачами кафедри інформаційних систем. Керівник курсового проекту рекомендує студенту основну літературу, орієнтує його на розроблення необхідних проектних рішень.

Студенту може бути призначена тема курсового проекту з переліку рекомендованих тем (див. додаток А). Також йому надається право самостійного вибору теми проекту. Якщо тема пропонується студентом, то вона має бути обговорена й погоджена з керівником курсового проекту.

Після затвердження вибраної теми студентові видається завдання на курсове проектування. У завданні міститься тема курсового проекту, вихідні дані до проекту, зміст пояснювальної записки, строки початку й закінчення роботи над курсовим проектом, обумовлені графіком навчального процесу, план-графік виконання етапів курсового проектування. Зразок завдання на курсовий проект наведено у додатку Б.

Після вивчення літературних джерел здобувач складає попередній план виконання курсового проекту, обговорює його з керівником. У процесі обговорення уточнюються вихідні дані для проектування й строки, що регламентують роботу здобувачів над проектом. Після цього студент складає уточнений план роботи над проектом, узгоджує його з керівником та приступає до проектування. У процесі виконання курсового проекту студент повинен регулярно відвідувати консультації керівника, подавати йому на перевірку робочі матеріали відповідно до плану-графіка виконання етапів курсового проектування.

Матеріали виконаного курсового проекту: пояснювальну записку в друкованому та електронному вигляді, вихідний код програмного застосунку, програму-інсталлятор, презентацію доповіді в електронному вигляді студент має здати на перевірку керівникові не пізніше визначеного строку.

Захист курсових проектів організовується кафедрою інформаційних систем за затвердженим графіком.

Під час захисту студент коротко доповідає про поставлене йому завдання, проектні рішення, що були прийняті, одержані результати, відповідає на запитання. Доповідь має супроводжуватися демонстрацією електронної презентації, яка має містити як мінімум такі слайди:

1. Титульний слайд з вихідними даними щодо курсового проекту: тема проекту, П.І.Б. виконавця, П.І.Б. керівника тощо.
2. Зміст презентації.
3. Актуальність теми та мета курсового проекту.
4. Коротка постановка завдання.
5. Математичний опис задачі (декілька слайдів).
6. UML-діаграма класів, що реалізують основну бізнес-логіку програмної системи.
7. Використані інструментальні засоби та технології.
8. Висновки за результатами виконання курсового проекту.
9. Заключний слайд.

3. Структура та обсяг курсового проекту

Курсовий проект складається з пояснювальної записки та інших матеріалів, зокрема програмного застосунку, що розробляється відповідно до завдання.

Обсяг пояснювальної записки – приблизно 20 – 30 друкованих сторінок формату А4 (без додатків). Матеріали пояснювальної записки мають бути зброшурованими.

У табл. 1 наведено структура пояснювальної записки курсового проекту.

4. Методичні рекомендації до розроблення структурних елементів пояснювальної записки курсового проекту

Титульний аркуш є першою сторінкою пояснювальної записки. Він містить дані, які подають у такій послідовності:

- відомості про виконавця роботи;
- повна назва документа;

підписи відповідальних осіб, включаючи керівника роботи;
рік складення пояснювальної записки;

Приклад титульного аркушу наведено у додатку В.

Реферат – це короткий виклад змісту пояснювальної записки, що містить основні фактичні відомості і висновки, необхідні для початкового ознайомлення з нею.

Реферат має бути стислим, інформативним і містити відомості, які дозволяють прийняти рішення про доцільність читання пояснювальної записки.

Таблиця 1

Структура пояснювальної записки курсового проекту

Структурні елементи пояснювальної записки	Кількість сторінок
Титульний аркуш	1
Завдання на курсове проектування	2 (на одному аркуші)
Реферат	1
Зміст	1
Вступ	1
1 Специфікація проекту	3 - 6
1.1 Постановка завдання	1 - 2
1.2 Вимоги до програмного забезпечення	1 - 2
1.3 Математичний опис задачі	1 - 2
2 Програмна документація	10 - 17
2.1 Архітектура програмної системи	1 - 3
2.2 Тестування програмної системи	3 - 5
2.3 Розгортання програмного продукту	1 - 2
2.4 Керівництво користувача	4 - 7
2.4.1 Призначення програмного продукту	1
2.4.2 Використання програмного продукту	2 - 4
2.4.3 Повідомлення користувачеві	1 - 2
Висновки	1
Список використаних джерел	1
Додатки	

Реферат повинен містити:

текст реферату;
перелік ключових слів.

Текст реферату повинен відбивати подану в пояснювальній записці інформацію у такій послідовності:

об'єкт дослідження або розроблення;

мета роботи;

методи дослідження та апаратура;

результати та їх новизна;

основні технологічні й техніко-експлуатаційні характеристики та показники;

взаємозв'язок з іншими роботами;

рекомендації щодо використання результатів курсового проекту;

галузь застосування;

значущість роботи та висновки;

прогнози припущення про розвиток об'єкту дослідження або розроблення.

Реферат належить виконувати обсягом не більш, як 500 слів, і, бажано, щоб він уміщувався на одній сторінці формату А4.

Ключові слова призначені для розкриття суті проекту та для розповсюдження інформації про розробку. Їх розміщують після тексту реферату. Перелік ключових слів містить від 5 до 15 слів (словосполучень), надрукованих великими літерами в називному відмінку в рядок через коми.

Приклад реферату наведено у додатку Г.

До змісту включають: вступ; послідовно перелічені назви всіх розділів, підрозділів та пунктів основної частини пояснювальної записки; висновки; перелік посилань; назви додатків і номери сторінок, які містять початок матеріалу.

Приклад змісту пояснювальної записки наведено у додатку Д.

У вступі слід зазначити важливість використання інформаційних систем у сучасних умовах, роль інформаційних систем у предметної галузі, що розглядається в курсовому проекті, роль засобів збереження даних у інформаційних системах, мету та задачі курсового проекту, відомості щодо розробленої програми (призначення, що вона дозволяє автоматизувати, технології та мова програмування, що були використані, при розробленні програми), заключна частина.

Перший розділ «Специфікація проекту» містить постановку завдання, вимоги до програмного забезпечення та математичний опис задачі.

У підрозділі 1.1 «Постановка завдання» наводиться постановка завдання на розроблення програмного продукту відповідно до варіанту, який був виданий здобувачу. Опис постановки завдання за будь-якою темою як мінімум має містити інформацію аналогічну наведеної в додатку Ж.

У підрозділі 1.2 «Вимоги до програмного забезпечення» містяться функціональні та нефункціональні вимоги до програмної системи.

Функціональні вимоги явно описують, що повинна робити програмна система і які перетворення вхідних даних виконувати.

Нефункціональні вимоги визначають властивості програмної системи, прямо не пов'язані з її функціональністю. Прикладом таких властивостей може служити максимальний час відгуку системи на запит користувача, мінімальний час безперебійної роботи системи, наявність графічного інтерфейсу користувача й таке інше.

Функціональні та нефункціональні вимоги до програмної системи, що розробляється відповідно до будь-якої теми курсового проекту з переліку рекомендованих тем, наведено нижче. У обґрунтованих випадках деякі з цих вимог можуть бути змінені за узгодженням з керівником курсового проекту.

В будь-якому випадку ці вимоги повинні мати відношення до певної предметної області та конкретної розроблюваної програми.

Функціональні вимоги

1. Створення сховища даних та запис до нього даних у певному форматі.
2. Читання всіх даних зі сховища даних та їх відображення.
3. Додавання нового елемента даних до сховища даних.
4. Оновлення будь-якого елемента даних у сховищі.
5. Видалення будь-якого елемента даних зі сховища.
6. Отримання та відображення підсумкової інформації.
7. Перевірка допустимості основних даних, що вводяться користувачем.
8. Видача користувачу попереджуючих та інформаційних повідомлень.

Нефункціональні вимоги

Мінімальні вимоги до інтерфейсу користувача:

1. Інтерфейс користувача має складатися з текстових повідомлень українською (англійською) мовою, які відносяться до введення та виведення основних та допоміжних даних. Використання транслітерації заборонено.
2. Наявність текстового меню для вибору варіантів дій користувача.
3. Дані, що зберігаються в сховищі даних, мають відображатися у табличному вигляді.

Під час проєктування інтерфейсу користувача слід дотримуватись таких правил:

перед проєктуванням інтерфейсу слід чітко визначити, які завдання будуть вирішувати користувачі за допомогою програмного продукту і які вимоги до інтерфейсу можуть у них виникнути;

інтерфейс має бути інтуїтивно-зрозумілим;

інтерфейс має допомагати (а не перешкоджати) користувачу виконувати основні завдання, для яких розроблено програмний продукт, а саме:

(1) надавати користувачеві чіткі зрозумілі інструкції; (2) забезпечувати перевірку і попередження «некоректних» дій користувача; (3) виводити зрозумілі лаконічні інформаційні повідомлення в ситуаціях, коли це дійсно необхідно;

слід дотримуватись стандартних угод про розташування пунктів меню, прийнятих в операційній системі, і групувати пункти меню в логічному порядку.

Вимоги до архітектури програми:

1. Використання не менше однієї структури даних із стандартної бібліотеки колекцій.
2. Використання потоків введення-виведення даних.
3. Використання механізму винятків для обробки помилок.

Вимоги до вихідного коду застосунку:

1. Додержання принципу інкапсуляції щодо рівнів доступу до полів та методів класів.

2. Вихідний код кожного з класів програми повинен міститися в окремому файлі.

3. Імена класів, інтерфейсів, полів класів, властивостей, методів, локальних змінних та констант тощо повинні мати смислове навантаження в певній предметній області.

Якщо здобувач вибрав тему курсового проекту самостійно, то вимоги до програми обговорюються з керівником та можуть відрізнятися від наведених вище.

У підрозділі 1.3 «Математичний опис задачі» має бути наведена математична формалізація задачі, тобто існуючі співвідношення між величинами, які виражають залежність показників, що розраховуються, від вхідних даних. При цьому залежно від специфіки розв'язуваної задачі можуть бути використані різні розділи математики та інших дисциплін. Таким чином формується математична модель явища з певною точністю, припущеннями і обмеженнями.

Математична модель повинна задовольняти принаймні двом вимогам: реалістичності і реалізованості.

Під реалістичністю розуміється правильне відображення моделлю найбільш істотних рис досліджуваного явища.

Реалізованість досягається розумною абстракцією, відволіканням від другорядних деталей, щоб звести задачу до задачі з відомим рішенням. Умовою реалізованим є можливість практичного виконання необхідних обчислень за відведений час при доступних витратах необхідних ресурсів.

Отже, у цьому підрозділі необхідно навести математичні формули або логічні співвідношення, які виражають залежність показників, що розраховуються, від вхідних даних та необхідні пояснення.

Другий розділ «Програмна документація» складається з чотирьох підрозділів: 2.1 – 2.4. Цей розділ містить опис архітектури розробленої програмної системи, відомості про тестування програмної системи та розгортання програмного продукту, а також керівництво користувача.

Підрозділ 2.1 «Архітектура програмної системи».

Програмна система означає програмне забезпечення, що розроблюється. Це може бути колекція з багатьох компонентів програмного забезпечення, один застосунок або частина застосунку.

Для уявлення статичної структури моделі програмної системи призначена UML-діаграма класів. Вона може відбивати, зокрема, різні

взаємозв'язки між окремими сутностями предметної області, такими як об'єкти й підсистеми, а також описує їхню внутрішню структуру й типи відносин.

У даному підрозділі пояснювальної записки необхідно навести:

1. UML-діаграму класів, що реалізують основну бізнес-логіку програмної системи, та її опис. Приклад опису UML-діаграми класів наведено у додатку 3.

2. Посилання на лістинг програми з вихідним кодом, що відповідає класам, які реалізують основну бізнес-логіку програмної системи. Лістинг програми повинен знаходитися в одному з додатків до пояснювальної записки.

Підрозділ 2.2 «Тестування програмної системи»

Тестування програмної системи – процес виконання програмного коду, спрямований на виявлення існуючих у ньому дефектів. Під дефектом розуміється ділянка програмного коду, виконання якої за певних умов приводить до несподіваного поведження системи (тобто поведження, що не відповідає вимогам).

Завдання тестування – визначення умов, при яких проявляються дефекти системи, і протоколювання цих умов.

Мета застосування процедури тестування програмного коду – мінімізація кількості дефектів у кінцевому продукті.

Види тестування

Модульне тестування

У ході модульного тестування кожний модуль тестується як на відповідність вимогам, так і на відсутність проблемних ділянок програмного коду, які можуть викликати відмови й збої в роботі системи.

Інтеграційне тестування

Окремі модулі рідко функціонують самі по собі, тому наступне завдання після тестування окремих модулів – тестування коректності взаємодії декількох модулів, об'єднаних у єдине ціле. Таке тестування називають інтеграційним.

Системне тестування

По завершенню інтеграційного тестування всі модулі системи є погодженими по інтерфейсах і функціональності. Починаючи із цього моменту можна переходити до системного тестування, тобто тестування поведження системи в цілому як єдиного об'єкта.

Вхідною інформацією для проведення системного тестування є два класи вимог: функціональні й не функціональні.

Системне тестування проводиться в кілька етапів, на кожному з яких використовується один з видів системного тестування.

Важливим видом системного тестування є функціональне тестування. Даний вид системного тестування призначений для підтвердження того, що вся система в цілому поводить ся відповідно до очікувань користувача, формалізованих у вигляді системних вимог. У ході функціонального тестування перевіряються всі функції системи з погляду її користувачів (як людей, так і інших програмних систем). Також необхідно перевірити функціональну повноту користувальницького інтерфейсу й коректність виведення інформації.

Документування процедури тестування

Основне призначення документації, створеної при тестуванні, - забезпечення гарантій того, що процес тестування виконується з необхідною якістю й всі аспекти поведження системи протестовані.

Перелік необхідної документації:

1. Тест-вимоги.
2. Тест-план.
3. Звіт про тестування.

Тест-вимоги розробляються на підставі системних і функціональних вимог до застосунку. У них докладно описується, які аспекти поведження системи повинні бути протестовані, щоб упевнитися в її коректному функціонуванні, і на підставі якого зовнішнього ефекту можна переконатися, що функціональність, яка перевіряється, реалізована правильно.

Тест-вимоги повинні бути достатніми для побудови тест-плану перевірки програмної системи без знайомства з її програмним кодом.

Структура тест-вимог повинна дотримуватися структури функціональних вимог на систему. Як правило, одній системній або функціональній вимозі відповідає мінімум одна тест-вимога.

Для кожної тест-вимоги повинна існувати можливість перевірки - виконується ця вимога в реалізованій системі чи ні.

На підставі тест-вимог створюються тест-план - документ, що містить докладний покроковий опис того, як повинні бути протестовані тест-вимоги. На відміну від тест-вимог у тест-плані описуються конкретні способи перевірки функціональності системи.

Як правило, тест-план складається з окремих тестових прикладів, кожний з яких перевіряє деяку функцію або набір функцій системи. Для кожного тестового прикладу однозначно визначається критерій успішного проходження, за допомогою якого можна судити про відповідність поведінки системи заданому.

Структура тест-плану повинна відповідати структурі тест-вимог.

Кожний пункт тест-плану повинен містити:

1. Посилання на вимогу(и), що перевіряється цим пунктом;
2. Конкретне значення вхідних даних.
3. Очікувану реакцію програми (тексти повідомлень, значення результатів).
4. Опис послідовності дій, необхідних для виконання пунктів тест-плану.

За результатами виконання тестів створюється звіт про виконання тестування. Він є основним джерелом для висновку про ступінь відповідності протестованої системи вимогам. Такий звіт як мінімум повинен містити інформацію про кожний виконаний тестовий приклад і результат його виконання (успіх або невдача).

Іноді тест-план сполучають зі звітом про проведення тестування, додаючи до нього інформацію про отриману реакцію системи й збіг (розбіжності) отриманих результатів з очікуваними. Наприкінці опису кожного тестового прикладу додається інформація про те, чи пройдений тестовий приклад у цілому. Наприкінці всього тест-плану, сполученого зі звітом, міститься графа "Тестових прикладів пройдено/усього", у яку заноситься число пройдених тестових прикладів і загальна їхня кількість.

Приклад тест-плану, сполученого зі звітом про проведення тестування наведено у додатку И.

У цьому пункті пояснювальної записки має знаходитися опис процедур функціонального тестування та їхніх результатів, що відносяться до певної предметної області.

Для опису процедури тестування повинні бути складені наступні документи:

1. Тест-вимоги.
2. Тест-плани, сполучені зі звітами про проведення тестування.

У звіті про проведення тестування вказуються як позитивні так і негативні результати виконання окремих тестів.

Однак, загальний результат тестування застосунку повинен бути позитивним, бо в протилежному випадку він не відповідає певним вимогам. Для цього в разі необхідності проводяться додаткові заходи щодо виправлення помилок та тестування. Вони також повинні бути описані у даному пункті пояснювальної записки.

Розгортання - це процес поширення готового застосунку або компонента для установки на інші комп'ютери. У підрозділі 2.3 «Розгортання програмного продукту» необхідно навести опис вимог до апаратних та програмних засобів, необхідних для функціонування розробленого програмного продукту, та дій щодо його інсталяції на комп'ютері користувача.

Приклад опису процедури розгортання програмного продукту, створеного на платформі Java SE наведено у додатку К.

Підрозділ 2.4 «Керівництво користувача» містить призначення програми, інструкцію щодо її використання за призначенням, опис повідомлень користувачу, що можуть з'явитися у процесі роботи програми.

У пункті 2.4.1 «Призначення програмного продукту» вказуються відомості про його призначення та інформація, достатня для розуміння функцій програмного продукту.

У пункті 2.4.2 «Використання програмного продукту» має бути вказана послідовність дій користувача, яка забезпечує запуск, виконання та завершення програми, наведено опис функцій, формату та можливих варіантів дій, за допомогою яких користувач керує виконанням програми, а також реакцію програми на ці дії.

У пункті 2.4.3 «Повідомлення користувачеві» наводяться екранні форми повідомлень, що можуть з'являтися в ході виконання програми, та опис їх змісту.

Приклад керівництва користувача наведено у додатку Л.

У висновках наводять оцінку одержаних результатів роботи (негативних також); можливі галузі використання результатів роботи; економічну, наукову, соціальну значущість роботи.

Список використаних джерел – це перелік джерел інформації, які було цитовано, згадано або розглянуто у роботі. Джерела можна розміщувати в списку одним із таких способів: в порядку появи посилань у тексті, в алфавітному порядку прізвищ перших авторів або заголовків. Вимоги до оформлення списку використаних джерел наведено у [1].

У додатках вміщують матеріал, який є необхідним для повноти пояснювальної записки, але не може бути послідовно розміщений в її основній частині через великий обсяг або способи відтворення та з інших причин.

Ілюстрації (діаграми бізнес-процесів, схеми алгоритмів, технологічних процесів, сценарії діалогів та ін.), таблиці, проміжні математичні докази, формули та розрахунки, текст допоміжного характеру та інші матеріали можуть бути оформлені у вигляді додатків.

5. Вимоги до оформлення пояснювальної записки курсового проекту

Важливе значення при роботі над курсовим проектом має його оформлення, до якого пред'являються певні вимоги.

При оформленні тексту пояснювальної записки слід керуватися державним стандартом України ДСТУ 3008-2015 "Документація. Звіти у сфері науки і техніки. Структура і правила оформлення" [2]. Далі наведені загальні вимоги до оформлення пояснювальної записки курсового проекту.

Матеріали пояснювальної записки друкуються на аркушах формату А4. Їх текст повинен відповідати правилам граматики й стилістики.

Текст роботи необхідно форматувати, залишаючи на аркушах поля таких розмірів: ліве – 30 мм, праве – 10 мм, верхнє – 20 мм, нижнє – 20 мм.

Текст документу повинен бути виконаний з використанням шрифту Times New Roman (розмір 14), з міжрядковим інтервалом 1,2 (37 рядків на сторінці). Найменшим розміром шрифту може бути розмір 10 (його можна використовувати при поданні вихідного тексту програм). Шрифт друку повинен бути чітким, текст – чорного кольору середньої жирності. Кольоровий друк дозволяється використовувати лише для рисунків (екранні форми, діаграми тощо). Щільність тексту повинна бути однаковою. Вирівнювання основного тексту проводиться «за шириною» сторінки.

Весь текст пояснювальної записки, включаючи назви її структурних елементів, виконується шрифтом однакової жирності. Не дозволяється використання курсиву та підкреслення.

Абзацний відступ повинен бути однаковим впродовж усього тексту та дорівнювати 1,25 см.

Друкарські помилки, описки і графічні неточності, які виявилися в процесі виконання документу, можна виправляти підчищенням або зафарбовуванням білою фарбою і нанесенням на тому ж місці виправленого тексту. Допускається наявність не більше двох виправлень на одній сторінці.

При скороченні слів і словосполучень потрібно спочатку навести повну назву, а після цього в дужках – її скорочення.

Кожний структурний елемент пояснювальної записки (крім підрозділів, пунктів, підпунктів) повинен починатися з нової сторінки.

Назви елементів «РЕФЕРАТ», «ЗМІСТ», «ВСТУП», «ВИСНОВКИ», «СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ», «ДОДАТКИ» розміщують симетрично до тексту (від центру), без абзацного відступу, не нумерують (не можна друкувати «1. ВСТУП» або «РОЗДІЛ 6. ВИСНОВКИ»), виконують великими буквами без крапки наприкінці та не підкреслюють.

Заголовки підрозділів, пунктів і підпунктів слід починати з абзацного відступу і друкувати маленькими літерами, крім першої великої, не підкреслюючи, без крапки в кінці.

Переноси в середині слова в заголовках не допускаються.

Не дозволяється розміщати заголовки й підзаголовки в нижній частині сторінки, якщо на ній тільки один рядок наступного тексту.

Відстань між заголовком і подальшим чи попереднім текстом має бути два рядки. Відстань між основами рядків заголовку, а також між двома заголовками приймають такою, як у тексті.

Розділи, підрозділи, пункти, підпункти треба нумерувати арабськими цифрами. Розділи мають порядкову нумерацію, наприклад: 1, 2, 3. Підрозділи повинні мати порядкову нумерацію в межах розділу. Номер підрозділу включає номер розділу й порядковий номер підрозділу, які розділяються крапкою, наприклад: 1.1, 1.2, 1.3. Номер пункту включає номер розділу, підрозділу, порядковий номер параграфа, які розділяються крапкою, наприклад: 1.1.1, 1.1.2, 1.1.3.

Сторінки пояснювальної записки повинні бути пронумеровані арабськими цифрами в правому верхньому куті без крапки. Нумерація сторінок наскрізна від титульного аркуша до останнього аркуша тексту, включаючи ілюстрації, таблиці, графіки. На титульному аркуші, завданні на курсовий проект нумерація сторінок не проставляється.

Викладені у тексті матеріали повинні наочно доповнювати й підтверджувати ілюстрації (схеми, рисунки, графіки, діаграми). Ілюстрації повинні відбивати тему курсового проекту. Студентові необхідно продумати, який матеріал проілюструвати. Це діаграми класів, схеми алгоритмів, схеми інформаційних зв'язків тощо.

Усі ілюстрації іменуються рисунками, їм привласнюється порядковий номер у межах номера розділу. Підпис ілюстрації складається із слова «Рисунок», номера ілюстрації та її назви (наприклад, "Рисунок 3.1 - Схема розміщення"). Рисунки потрібно виконувати на одній сторінці й розташовувати відразу після першого згадування в тексті, або на наступній сторінці. На всі ілюстрації мають бути посилання у тексті.

Посилання на ілюстрації роботи вказують порядковим номером ілюстрації, наприклад, «рис. 1.2».

У повторних посиланнях на ілюстрації треба вказувати скорочено слово «дивись», наприклад: «(див. рис. 1.2)».

У тому місті, де викладається тема, пов'язана з ілюстрацією, розміщують посилання у вигляді виразу у круглих дужках «(рис. 3.1)» або зворот типу: «... як показано на рис. 3.1».

Кожну формулу записують після першого згадування в тексті з нового рядка, симетрично до тексту. Між формулою і текстом пропускають один рядок. Формули нумеруються в межах розділу.

Пояснення до умовних літерних позначень в формулі наводять 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 із слова «де».

Наприклад,

$$I = UR, \quad (2.1)$$

де U - електрична напруга;

R – сила електричного струму.

Якщо формула займає декілька рядків, то вона може бути розірвана тільки на математичних знаках: додавання, віднімання, множення, ділення та інших, які повторюють на початку наступного рядка.

Таблицю необхідно розташовувати безпосередньо після тексту, у якому вона згадується вперше або на наступній сторінці. На всі таблиці повинні бути посилання. Таблиці послідовно нумеруються в межах розділу. Над лівим верхнім кутом таблиці міститься напис «Таблиця» із вказівкою її порядкового номера та назви (наприклад, Таблиця 1.1 - Структурні елементи пояснювальної записки).

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).

Для подальшої деталізації переліку слід використовувати арабські цифри з дужкою (другий рівень деталізації).

Переліки першого рівня деталізації друкують малими літерами з абзацного відступу, другого рівня – відступом відносно місця розташування переліків першого рівня.

У тексті пояснювальної записки повинні бути посилання на літературу. При цьому наводиться її порядковий номер, записаний у квадратних дужках (наприклад, «...у роботах [1-7]»).

Додатки потрібно оформляти як продовження пояснювальної записки на її наступних сторінках. Кожен додаток повинен починатися з нової сторінки. Додаток повинен мати заголовок, надрукований вгорі малими літерами з першої великої симетрично відносно тексту сторінки. Посередині рядка над заголовком малими літерами з першої великої повинно бути надруковано слово "Додаток" і велика літера, що позначає додаток (наприклад, «Додаток А»).

Додатки позначаються послідовно великими літерами української абетки за винятком букв Г, Є, І, Ї, Й, О, Ч, Ь. Один додаток позначається як додаток А.

Рекомендована література

Основна

1. Бібліографічний запис. Бібліографічний опис. Загальні вимоги та правила складання ДСТУ ГОСТ 7.1:2006. - К. : Держстандарт України, 2007. — 57 с. [Електронний ресурс]. - Режим доступу:http://lib.khnu.km.ua/storinka_vykladacha/oforml_spysku/7.1-2006%20%28Ukr%29.pdf
2. Інформація та документація. Звіти у сфері науки і техніки. Правила оформлення: ДСТУ 3008-2015. — К. : ДП “УкрНДНЦ”, 2016. — 31 с. [Електронний ресурс]. - Режим доступу: https://science.kname.edu.ua/images/dok/derzhstandart_3008_2015.pdf
3. Booch G. Object-Oriented Analysis and Design with Applications / G.Booch, R.A.Maksimchuk et. al. – Boston : Addison-Wesley, 2019. – 717 p.
4. Troelsen A. Pro C# 9 with .NET 5 : Foundational Principles and Practices in Programming / A. Troelsen, P. Japikse – Berkly : Apress, 2021. – 1353 p.

Додаткова

5. Lewis J. Java Foundations. Introduction to Program Design and Data Structures / J. Lewis, P. DePasquale, J. Chase – Lancing : Pearson, 2020. – 1104 p.

Ресурси мережі Internet

1. What are UML diagrams, and how can you use them. [Electronic resource]. – Access mode: <https://miro.com/blog/uml-diagram/>
2. C# documentation [Electronic resource]. – Access mode: <https://docs.microsoft.com/en-us/dotnet/csharp/>
3. C# Tutorial [Electronic resource]. – Access mode: <https://www.tutorialspoint.com/csharp/index.htm>
4. Java Tutorial [Electronic resource]. – Access mode: <https://www.w3schools.com/java/default.asp>
5. JDK 20 Documentation [Electronic resource]. – Access mode: <https://docs.oracle.com/en/java/javase/20/>

6. Щербаков, О. В. Основи об'єктно-орієнтованого програмування : навч. посіб. / О. В. Щербаков, Ю. Е. Парфьонов, В. М. Федорченко - Харків : ХНЕУ ім. С. Кузнеця, 2019. - 236 с. [Електронний ресурс]. - Режим доступу: <http://repository.hneu.edu.ua/handle/123456789/23847>

7. Об'єктно-орієнтоване програмування. Методичні рекомендації до лабораторних робіт для студентів спеціальності 121 "Інженерія програмного забезпечення" першого (бакалаврського) рівня / уклад. Ю. Е. Парфьонов, О. В. Щербаков - Харків : ХНЕУ ім. С. Кузнеця, 2021. - 92 с. [Електронний ресурс]. - Режим доступу: <http://repository.hneu.edu.ua/handle/123456789/27252>

8. Персональна навчальна система "Курсовий проєкт: Програмування" [Електронний ресурс]. - Режим доступу: <https://pns.hneu.edu.ua/course/view.php?id=10374>

Додатки

Додаток А

Перелік рекомендованих тем курсових проектів

1. Розроблення програмного продукту для роботи з даними про прихід товарів на склад підприємства.
2. Розроблення програмного продукту для роботи з даними про фізичних осіб.
3. Розроблення програмного продукту для роботи з даними про рух пасажирських поїздів по станції Харків.
4. Розроблення програмного продукту для роботи з даними про тривалість розмов абонентів АТС.
5. Розроблення програмного продукту для роботи з даними про товари магазину побутової техніки.
6. Розроблення програмного продукту для роботи з даними про банківські операції.
7. Розроблення програмного продукту для роботи з даними результатів моніторингу якості палива на автозаправній станції.
8. Розроблення програмного продукту для роботи з даними про поштові операції.
9. Розроблення програмного продукту для роботи з даними заявок на ремонт мобільних телефонів у сервісному центрі.
10. Розроблення програмного продукту для роботи з даними контактів.
11. Розроблення програмного продукту для роботи з даними про поштову індексацію м. Харкова та населених пунктів районів Харківської області.
12. Розроблення програмного продукту - довідника куратора студентської групи.
13. Розроблення програмного продукту для роботи з даними про продаж палива на автозаправній станції.
14. Розроблення програмного продукту для роботи з даними про час використання комп'ютерів підприємства.

15. Розроблення програмного продукту для роботи з даними про надання послуг абонентам Інтернет-провайдера.

16. Розроблення програмного продукту для роботи з даними про хід виконання робіт на задану дату.

17. Розроблення програмного продукту для роботи з даними про русі транспортних засобів.

18. Розроблення програмного продукту для роботи з даними штатного розкладу підприємства.

19. Розроблення програмного продукту для роботи з даними про залізничні пасажирські перевезення.

20. Розроблення програмного продукту для роботи з даними про продажі книг в книгарні.

21. Розроблення програмного продукту для роботи з даними про нарахування зарплати співробітникам підприємства.

22. Розроблення програмного продукту для роботи з даними про витрат палива на автобазах міста.

23. Розроблення програмного продукту для роботи з даними про використання машинного часу в обчислювальному центрі.

24. Розроблення програмного продукту для роботи з даними про споживання електроенергії на заводах міста.

25. Розроблення програмного продукту для роботи з даними про рух матеріалів на складі підприємства.

26. Розроблення програмного продукту для роботи з даними про прибуток підприємства за звітний період.

27. Розроблення програмного продукту для роботи з даними про відвідування занять студентами.

28. Розроблення програмного продукту для роботи з даними про поставки продукції.

29. Розроблення програмного продукту для роботи з даними про перевезення авіапасажирів.

30. Розроблення програмного продукту для роботи з даними про час роботи верстатів підприємства.

31. Розроблення програмного продукту для роботи з даними про випуск деталей робітниками цеху.

32. Розроблення програмного продукту для роботи з даними про рух основних фондів підприємства.

33. Розроблення програмного продукту для роботи з даними про облік запчастин на складі підприємства.

34. Розроблення програмного продукту для роботи з даними про успішність студентів.

35. Розроблення програмного продукту для роботи з даними про облік оплати за телефонні розмови.

36. Розроблення програмного продукту для роботи з даними про облік автомобілів в автосалоні.

37. Розроблення програмного продукту для роботи з даними про відвідування Інтернет-ресурсів користувачами.

38. Розроблення програмного продукту для роботи з даними про автомобільні запчастини на складі автомагазину.

39. Розроблення програмного продукту для роботи з даними про отримання студентами літератури в бібліотеці факультету.

40. Розроблення програмного продукту для роботи з даними про рух матеріалів на складі.

41. Розроблення програмного продукту для роботи з даними про абітурієнтів, які подали документи для вступу до вищого навчального закладу.

42. Розроблення програмного продукту для роботи з даними про ціни на основні продукти харчування у місті.

43. Розроблення програмного продукту для роботи з даними про нарахуванні зарплати співробітникам.

44. Розроблення програмного продукту для роботи з даними про випуск продукції підприємством.

45. Розроблення програмного продукту для роботи з даними про співробітників підприємства.

46. Розроблення програмного продукту для роботи з даними про заклази клієнтів.

47. Розроблення програмного продукту для роботи з даними про публікації студентів.

Зразок завдання на курсовий проект

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ СЕМЕНА КУЗНЕЦЯ

Факультет інформаційних технологій
Кафедра інформаційних систем

ЗАВДАННЯ

на курсовий проект: Об'єктно-орієнтоване програмування
студенту 2-го курсу групи 6.04.121.010.21.01
Варенику Григорію Васильовичу

1. Тема проекту: «Розроблення програмної системи для автоматизації роботи з даними про <Назва предметної області>»
2. Термін здачі студентом закінченого проекту «__» _____ 202_ р.
3. Вхідні дані до проекту: літературні джерела, технічна документація щодо розроблення програм, ДСТУ з оформлення документації
4. Зміст пояснювальної записки:
Вступ. Специфікація проекту. Програмна документація. Висновки.
Додатки.
5. Перелік графічного матеріалу: UML-діаграма класів програмної системи.

Календарний план
виконання курсового проекту

№ з/п	Назва етапу роботи	Строк виконання за планом	Відмітка про виконання
1.	З'ясування загальної постановки завдання. Розроблення чернетки першого розділу пояснювальної записки.		
2.	Деталізація завдань курсового проектування. Уточнення архітектури програмної системи. Розроблення остаточного варіанту першого розділу пояснювальної записки.		
3.	Програмна реалізація, налагодження та тестування застосунку.		
4.	Остаточне налагодження та тестування застосунку. Розроблення чернетки другого розділу пояснювальної записки, висновків, списку використаних джерел, додатків, вступу, реферату).		
5.	Розроблення остаточного варіанту пояснювальної записки та підготовка електронної презентації		

Дата видачі завдання " __ " _____ 202_ р.

Керівник к.т.н., доц. кафедри ІС _____ О.І. Петренко
(підпис)

Завдання прийняв до виконання _____ Г.В. Вареник
(підпис)

Зразок оформлення титульного аркуша

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ СЕМЕНА КУЗНЕЦЯ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ

КУРСОВИЙ Проект: ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ

на тему: «Розроблення програмної системи для автоматизації
роботи з даними про <Назва предметної області>»

Студента(ки) 2 курсу 6.04.121.010.21.01 групи
Спеціальності 121 "Інженерія програмного
забезпечення"

першого (бакалаврського) рівня

Вареник Г.В.

Керівник: доцент кафедри ІС, к.т.н., доцент,
Петренко О.І.

Національна шкала _____
Кількість балів: _____ Оцінка: ECTS _____

Члени комісії

_____ (підпис) _____ (прізвище та ініціали)

_____ (підпис) _____ (прізвище та ініціали)

м. Харків – 202_ рік

Зразок оформлення реферату

РЕФЕРАТ

Пояснювальна записка до курсового проекту: 32 с., 20 рис., 3 табл., 6 джерел.

Об'єктами проектування є функціональні елементи, архітектура та код програмної системи для роботи з даними про надання послуг абонентам інтернет-провайдера.

Метою роботи є розроблення програмної системи для роботи з даними про надання послуг абонентам інтернет-провайдера.

Метод проектування – використання програмних продуктів `app.diagrams.net` та `Microsoft Visual Studio`.

Створена програмна система дозволяє підвищити достовірність обліку послуг, обґрунтованість та оперативність прийняття рішень інтернет-провайдером.

Програмний продукт може бути використаний на підприємствах, які надають послуги доступу до мережі Інтернет.

АБОНЕНТ, МЕРЕЖЕВИЙ ТРАФІК, ШВИДКІСТЬ ПЕРЕДАЧІ ДАНИХ, UML-ДІАГРАМА КЛАСІВ, СХОВИЩЕ ДАНИХ, .NET, C#

Зразок оформлення змісту пояснювальної записки

ЗМІСТ

Вступ.....	6
1 Специфікація проекту.....	8
1.1 Постановка завдання.....	8
1.2 Вимоги до програмного забезпечення.....	10
1.3 Математичний опис задачі.....	12
2 Програмна документація.....	14
2.1 Архітектура програмної системи.....	14
2.2 Тестування програмної системи.....	16
2.3 Розгортання програмного продукту.....	21
2.4 Керівництво користувача.....	23
2.4.1 Призначення програмного продукту.....	23
2.4.2 Використання програмного продукту.....	24
2.4.3 Повідомлення користувачеві.....	28
Висновки.....	30
Список використаних джерел.....	31
Додатки.....	32

Приклад опису постановки завдання

Підприємство веде неавтоматизований облік яблук з використанням відомості в «паперовому» вигляді. Кількість записів відомості не обмежена.

Відомість має табличну структуру з чотирма стовпцями, кожний з яких має назву, яка відповідає одному з параметрів будь-якого яблука: сорт (S), колір (C), діаметр (D), об'єм (V). Об'єм будь-якого яблука обчислюється за формулою 1.1:

$$V_i = (1/6) \cdot \pi \cdot D^3, \quad (1.1)$$

де i – номер запису відомості.

Приклад заповненої відомості – див. табл. 1.1.

Таблиця 1.1

Сорт	Колір	Діаметр	Об'єм
S	C	D	V
Антонівка	жовтий	6	113,04
Джонатан	червоний	10	523,33
...
Середній об'єм яблука			120,99

Кожен запис відомості призначено для зберігання даних деякого яблука, наприклад:

Антонівка	жовтий	6	113,04
-----------	--------	---	--------

Співробітник підприємства може додавати до відомості дані про нове яблуко шляхом внесення до відповідних полів запису відомості початкових даних та даних, які були розраховані за формулою 1.1.

Наприкінці звітної періоду співробітник підприємства обчислює середній об'єм яблука V_{cp} за формулою 1.2 і заносить його в графу «Середній об'єм яблука» відомості:

$$V_{cp} = (\sum V_i) / n, \quad (1.2)$$

де V_i – діаметр i -го яблука в відомості;

n - кількість яблук у відомості.

Розробити об'єктно-орієнтовану програмну систему, яка призначена для автоматизації обліку яблук на підприємстві.

Приклад опису UML-діаграми класів

Діаграму класів програми наведено на рис. 2.1.

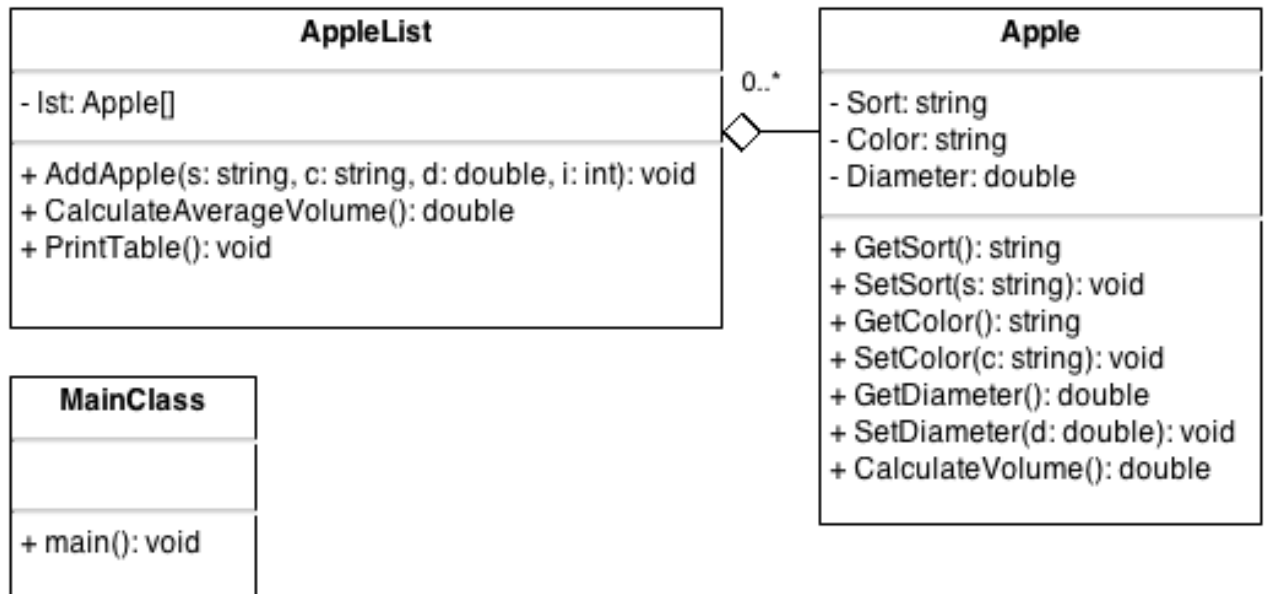


Рис. 2.1 - UML-діаграма класів

Програма складається з трьох класів: Apple, AppleList, MainClass.

Клас Apple призначено для опису будь-якого яблука. Поле Sort - сорт яблука, поле Color - колір яблука, поле Diameter - діаметр яблука. Метод GetSort() повертає поточне значення сорту яблука, метод GetDiameter() повертає поточне значення діаметру яблука, метод GetColor() повертає поточне значення його кольору. Метод SetSort() встановлює нове значення сорту яблука, методи SetDiameter() та SetColor() - його діаметру та кольору відповідно. Метод CalculateVolume() - обчислює об'єм яблука.

Клас AppleList призначено для опису відомості обліку яблук. Між класами AppleList і Apple існує відношення агрегації: поле lst класу AppleList являє собою масив об'єктів типу Apple і призначене для зберігання об'єктів-«яблук» (тобто один об'єкт класу AppleList може містити кілька об'єктів класу Apple).

Метод `AddApple()` необхідний для додавання об'єкта-«яблука» в масив `Ist`. Метод `CalculateAverageVolume()` - обчислює середній об'єм яблука у відомості. Призначення методу `PrintTable()` - «друк» значень полів об'єктів-«яблук», діаметру яблука і підсумкових даних відомості.

Клас `MainClass` - головний клас. Метод `main()` - точка входу в програму. Цей метод призначено для введення бажаної кількості яблук у відомості, створення об'єкта класу `AppleList`, введення параметрів яблук, створення на основі введених параметрів об'єкту класу `Apple`, для кожного з яблук, додавання цих об'єктів у відомість за допомогою виклику методу `AddApple()` об'єкту класу `AppleList`, виведення даних за допомогою виклику методу `PrintTable()` класу `AppleList`.

Приклад тест-плану, сполученого зі звітом про
проведення тестування

Тестовий приклад: № 1

Призначення: перевірка того, що програмна система дозволяє виконувати читання даних про послуги інтернет-провайдера з файлу та коректна відобразити їх у інтерфейсі користувача.

Тест-вимоги, що перевіряються: функціональна вимога № 2

Передумови для тесту: програмна система повинна бути запущена, на диску комп'ютера має знаходитися файл з даними у визначеному форматі.

Критерій проходження тесту: реальна поведінка програмної системи збігається з очікуваною.

Таблиця 3.1

№ з/п	Крок сценарію	Очікуваний результат	Отриманий результат	Відмітка про проходження кроку сценарію (Так/Ні)
1	2	3	4	5
1.	У меню програми вибрати пункт «Відкрити»	Має з'явитися повідомлення про введення шляху до файлу, який треба відкрити	З'являється повідомлення про введення шляху до файлу, який треба відкрити	Так
2.	У командному рядку ввести шлях до файлу визначеного формату та натиснути кнопку «Enter» клавіатури	У консольному вікні програми мають коректна відобразитися завантажені дані	Дані, що були завантажені з файлу, коректна відображуються у консольному вікні програми	Так
3.	У командному рядку ввести шлях до файлу неприпустимого формату та натиснути кнопку «Enter» клавіатури	Має з'явитися повідомлення про те, що дані не можуть бути завантажені	З'являється повідомлення про те, що дані не можуть бути завантажені	Так

Відмітка про проходження тесту (пройдений/не пройдений):
пройдений

Тестовий приклад: № 2

.....

Тестовий приклад: № 3

.....

Тестових прикладів виконано: 3

Тестових прикладів пройдено: 1

Опис процедури розгортання програмного продукту, створеного на платформі Java SE

Вимоги до апаратних засобів:

1. Процесор – двох ядерний, 2 ГГц або більш швидкий
2. Доступний простір ОЗУ - не менше 4 Гб
3. Вільний дисковий простір - не менше 25 Гб

Вимоги до програмних засобів:

1. Операційна система:
 - Windows 10, 11
 - Більшість сучасних дистрибутивів Linux
 - Mac OS X 10.14 або більш нова
2. Java Runtime Environment 17 або більш нове

Розгортання програмного продукту на комп'ютері користувача у вигляді автономного застосунку:

1. Створіть на цільовому диску каталог для застосунку, наприклад, MyApp.
2. В каталозі MyApp створіть новий каталог, наприклад, dist.
3. Скопіюйте виконуваний jar-файл застосунку (наприклад, install.jar) в каталог dist.
4. Завантажте з веб-сайту компанії Oracle програмну платформу Java Runtime Environment потрібної версії та розпакуйте відповідний архів у деякий каталог, наприклад у папку appjre.
5. Перемістите каталог appjre у каталог MyApp.
6. У каталозі MyApp створіть командний файл цільової операційної системи, наприклад, start.bat (операційна система Windows).
7. Додайте у start.bat наступні команди (операційна система Windows):

```
@echo OFF
set PATH =.\appjre\bin
java -jar dist\install.jar
pause> NUL
```

1. Для перевірки коректності запуску програми виконайте подвійне клацання лівою кнопкою миші на файлі start.bat (операційна система Windows).

Приклад керівництва користувача (у скороченні)

2.4 Керівництво користувача

2.4.1 Призначення програмного продукту

Програмний продукт призначено для автоматизації обліку яблук у компанії. Він дозволяє вести облік за такими параметрами кожного яблука: сорт, колір, діаметр та об'єм; додавати дані про нові яблука, обчислювати середній об'єм кожного яблука, зберігати дані про визначену кількість яблук.

2.4.2 Використання програмного продукту

Запуск програми

Запуск програми в операційній системі сімейства Windows здійснюється одним з стандартних способів:

- а) подвійним клацанням лівою кнопкою миші на ярлику програми;
- б) викликом контекстного меню з вибором його пункту «Відкрити»;
- в) натисканням кнопки «Пуск» панелі завдань з подальшим вибором пункту «Усі програми» та подвійним клацанням лівою кнопкою миші на ярлику програми.

Основні елементи користувацького інтерфейсу програми

Користувацький інтерфейс програми складається з головного меню (рис. 2.1), яке з'являється після її запуску на виконання.

Choose the menu item:

1. Enter initial data
 2. Print inventory
 3. Quit the program
-

Рис. 2.1 - Головне меню програми

Це меню містить усі команди для керування її виконанням:

1. «Enter initial data» призначено для початку введення даних.
2. «Print inventory» - для виведення списку яблук на консоль у табличному вигляді.
3. «Quit the program» - для завершення програми.

Робота з програмою

Введення початкових даних

Для початку роботи необхідно вибрати пункт головного меню «Enter initial data» шляхом введення цифри 1 та подальшого натискання клавіші <Enter> на клавіатурі. Після цього на екран послідовно виводяться повідомлення, для введення кількості яблук в облікової відомості, а також сорту, коліру та діаметру кожного яблука (рис. 2.2). Після введення значення кожного з параметрів необхідно натиснути <Enter>.

```
1
Enter number of apples: 2
Enter apple sort: Антонівка
Enter apple color: Жовтий
Enter apple diameter: 4.6
Enter apple sort: Джонатан
Enter apple color: Червоний
Enter apple diameter: 5
```

Рис. 2.2 - Інтерфейс введення даних

Виведення списку яблук на консоль

Для цього треба вибрати пункт головного меню «Print inventory» шляхом введення цифри 2 та натиснути клавішу <Enter> на клавіатурі.

Після виконання зазначених дій на екрані комп'ютера з'явиться список яблук у табличній формі (рис. 2.3). Для кожного з них у таблиці відображаються значення сорту, коліру, діаметру та обчисленого

об'єму. В рядку «Total» користувач програми має змогу побачити значення середнього об'єму яблука з наявних у таблиці.

Слід зазначити, що після вибору цього пункта головного меню таблиця з'являється тільки в тому разі, якщо користувач програми заздалегідь увів початкові дані щонайменш одного яблука.

2

Sort	Color	Diameter	Volume
Антонівка	Жовтий	4.60	50.97
Джонатан	Червоний	5.00	65.45
Total:			58.21

Рис. 2.3 - Список яблук у табличній формі

Завершення роботи з програмою

Для цього необхідно вибрати пункт меню «Quit the program» (цифра 3) та натиснути <Enter>.

2.4.3 Повідомлення користувачеві

Якщо користувач програми вибрав пункт «Print inventory» головного меню, але не ввів початкових даних, на екрані комп'ютера з'явиться таке повідомлення (рис. 2.4):

2
No data to print!

Рис. 2.4 - Повідомлення при відсутності даних для відображення

Зміст

Вступ	3
1. Мета й завдання курсового проектування	4
2. Організація курсового проектування	5
3. Структура та обсяг курсового проекту	6
4. Методичні рекомендації до розроблення структурних елементів пояснювальної записки курсового проекту	6
5. Вимоги до оформлення пояснювальної записки курсового проекту	16
Рекомендована література	19
Додатки	21

НАВЧАЛЬНЕ ВИДАННЯ

ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ

**Методичні рекомендації
до виконання курсового проекту
для здобувачів вищої освіти спеціальності 121
"Інженерія програмного забезпечення" освітньої
програми "Інженерія програмного забезпечення"
першого (бакалаврського) рівня**

Укладачі: Парфьонов Юрій Едуардович
Гризун Людмила Едуардівна

Відповідальний за випуск Бондаренко Д. О.

Редактор
Коректор

План 2024 р. Поз. № 78 ЕВ .

Підп. до друку Формат 60 x 90 1/16. Папір MultiCopy. Друк Riso.

Ум.-друк. арк. Обл.-вид. арк. Тираж прим. Зам. №

Видавець і виготівник — видавництво ХНЕУ ім. С. Кузнеця, 61166, м. Харків, пр.
Науки, 9-А

*Свідоцтво про внесення до Державного реєстру суб'єктів видавничої справи
Дк № 481 від 13.06.2001 р.*