

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ СЕМЕНА КУЗНЕЦЯ**

ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ

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

**Харків
ХНЕУ ім. С. Кузнеця
2024**

УДК 004.415.2(072.034)

О-13

Укладачі: Ю. Е. Парфьонов

Л. Е. Гризун

Затверджено на засіданні кафедри інформаційних систем.

Протокол № 7 від 21.12.2023 р.

Самостійне електронне текстове мережеве видання

Об'єктно-орієнтоване програмування [Електронний ресурс] :

О-13 методичні рекомендації до виконання курсового проекту для здобувачів вищої освіти спеціальності 121 «Інженерія програмного забезпечення» освітньої програми «Інженерія програмного забезпечення» першого (бакалаврського) рівня / уклад. Ю. Е. Парфьонов, Л. Е. Гризун. – Харків : ХНЕУ ім. С. Кузнеця, 2024. – 40 с.

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

Рекомендовано для здобувачів вищої освіти спеціальності 121 «Інженерія програмного забезпечення» освітньої програми «Інженерія програмного забезпечення» першого (бакалаврського) рівня.

УДК 004.415.2(072.034)

© Харківський національний економічний
університет імені Семена Кузнеця, 2024

Вступ

Сучасні умови господарювання потребують від фахівців з економічного управління всебічного використання новітніх інформаційних технологій. Широкі можливості комп'ютеризованих засобів щодо збирання, опрацювання та видачі потрібної інформації є здатними 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 зростає, оскільки тенденції розвитку комп'ютерної техніки свідчать про те, що, з одного боку, її складність і функціональні можливості швидко зростають, а, з іншого – це потребує більш досконалих програмних засобів для задоволення потреб користувачів.

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

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

Курсовий проєкт є важливим етапом підготовки здобувачів вищої освіти. Його спрямовано на більш глибоке осмислення й закріплення теоретичних знань, набутих під час вивчення вказаної навчальної дисципліни, та вдосконалення практичних навичок із розроблення програмного забезпечення та потрібної документації.

Ці методичні рекомендації встановлюють єдині правила та порядок виконання, оформлення й захисту курсового проєкту. Їх призначено для допомоги здобувачам вищої освіти спеціальності 121 «Інженерія програмного забезпечення» у виконанні курсового проєкту.

1. Мета й завдання курсового проєктування

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

Робота над курсовим проєктом сприяє систематизації, поглибленню й закріпленню знань, набутих здобувачами вищої освіти під час вивчення цієї навчальної дисципліни. У процесі курсового проєктування здобувачі вищої освіти розвивають навички із практичного застосування набутих знань шляхом розроблення застосунку з використанням сучасних інструментальних засобів. Крім цього, здобувач вищої освіти має показати вміння користуватися спеціальною літературою, державними стандартами, довідниками та іншими матеріалами з інформаційних технологій.

Під час розроблення курсового проєкту здобувач вищої освіти має показати **знання**:

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

Здобувач вищої освіти має **уміти**:

- формулювати постановку завдання та здійснювати 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 здобуття довідкової інформації;

- розробляти й оформлювати необхідну документацію.

Робота над курсовим проєктом певною мірою визначає загально-теоретичну та спеціальну підготовку здобувача вищої освіти і в остаточному підсумку готує його до майбутнього виконання більш складного й завершального етапу навчального процесу – дипломного проєктування. Здобувач вищої освіти має розглядати роботу над курсовим проєктом як своєрідну «репетицію» дипломного проєктування.

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

Таблиця 1

**Результати навчання та компетентності,
які формує курсове проєктування**

Результати навчання	Компетентності, якими має оволодіти здобувач вищої освіти
РН8	СК1, СК10, СК13
РН13	СК2, СК3

Примітка.

РН8. Вміти розробляти людино-машинний інтерфейс.

РН13. 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 проєктуванні 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 проєктування

Якісне виконання курсового проєкту потребує чіткої організації роботи здобувача вищої освіти з моменту вибору теми проєкту й до його захисту.

Керівництво курсовим проєктуванням здійснюють викладачі кафедр інформаційних систем. Науковий керівник курсового проєкту рекомендує здобувачу вищої освіти основну літературу, орієнтує його на розроблення відповідних проєктних рішень.

Здобувачу вищої освіти може бути призначено тему курсового проєкту з переліку рекомендованих тем (додаток А). Також йому надано право самостійного вибору теми проєкту. Якщо тему пропонує здобувач вищої освіти, то її має бути обговорено й погоджено з науковим керівником курсового проєкту.

Після затвердження вибраної теми здобувачу вищої освіти видають завдання на курсове проєктування. У завданні міститься тема курсового проєкту; вихідні дані до проєкту; його зміст; терміни початку й закінчення роботи над курсовим проєктом, обумовлені графіком навчального процесу; план-графік виконання етапів курсового проєктування. Зразок завдання на курсовий проєкт наведено в додатку Б.

Після вивчення наукових джерел здобувач вищої освіти складає попередній план виконання курсового проєкту, обговорює його з науковим керівником. У процесі обговорення уточнюють вихідні дані для проєктування й терміни, що регламентують роботу здобувачів вищої освіти над проєктом. Після цього здобувач вищої освіти складає уточнений план роботи над проєктом, погоджує його з науковим керівником та розпочинає проєктування. У процесі виконання курсового проєкту здобувач вищої освіти має регулярно відвідувати консультації наукового керівника, подавати йому на перевірку робочі матеріали, відповідно до плану-графіка виконання етапів курсового проєктування.

Матеріали виконаного курсового проєкту у друкованому й електронному вигляді, вихідний код програмного застосунку, програму-інсталятор, презентацію доповіді в електронному вигляді – здобувач вищої освіти має здати на перевірку науковому керівникові не пізніше за визначений термін.

Курсовий проєкт перевіряють на унікальність. Порядок перевірки та відсоток унікальності визначено протоколом засідання кафедри. Результати перевірки є задовільними, якщо значення оцінок подібності є нижчими за критичні значення метрик, визначених під час перевірки.

Захист курсових проєктів організовує кафедра інформаційних систем за затвердженням графіком. Під час захисту здобувач вищої освіти

коротко доповідає про поставлене йому завдання, знайдені проєктні рішення, визначені результати, відповідає на запитання. Здобувачі вищої освіти мають супроводжувати свою доповідь демонстрацією електронної презентації, яка містить як мінімум такі слайди:

1. Титульний слайд з вихідними даними щодо курсового проєкту: тема проєкту, ПІБ виконавця, ПІБ наукового керівника тощо.
2. Зміст презентації.
3. Актуальність теми та мета курсового проєкту.
4. Коротке формулювання 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 проєкту.
9. Заключний слайд.

3. Структура й обсяг курсового проєкту

Курсовий проєкт складається зокрема із програмного застосунку, що розробляють відповідно до завдання, а також електронної презентації, приготовленої здобувачем вищої освіти для захисту курсового проєкту.

Обсяг курсового проєкту – приблизно 20 – 30 друкованих сторінок формату А4 (без додатків).

Матеріали курсового проєкту мають бути зброшурованими.

У табл. 2 наведено структуру курсового проєкту.

Таблиця 2

Структура курсового проєкту

Структурні елементи курсового проєкту	Кількість сторінок
1	2
Титульний аркуш	1
Завдання на курсове проєктування	2

1	2
Реферат	1
Зміст	1
Вступ	1
1. Специфікація проєкту	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	1 – 10

4. Методичні рекомендації до розроблення структурних елементів курсового проєкту

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

відомості про виконавця роботи;

повна назва роботи;

підписи відповідальних осіб, серед них наукового керівника роботи;

рік складання курсового проєкту;

Приклад титульного аркуша наведено в додатку В.

Реферат – це короткий виклад змісту курсового проєкту, що містить основні фактичні відомості та висновки, потрібні для початкового

ознайомлення з нею. Реферат має бути стислим, інформативним і містити відомості, які дозволяють визначити доцільність читання курсового проєкту.

Реферат має містити:

текст реферату;

перелік ключових слів.

Текст реферату має відображати подану в курсовому проєкті інформацію в такій послідовності:

об'єкт дослідження або розроблення;

мета роботи;

методи дослідження й апаратура;

результати та їхня новизна;

основні технологічні й техніко-експлуатаційні характеристики та показники;

взаємозв'язок з іншими роботами;

рекомендації до використання результатів курсового проєкту;

сфера застосування;

значущість роботи та висновки;

прогнозні припущення про розвиток об'єкта дослідження або розроблення.

Реферат належить виконувати обсягом, не більшим як 500 слів, бажано його розміщувати на одній сторінці формату А4.

Ключові слова призначено для розкриття сутності проєкту та для поширення інформації про результат проєкту. 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 в додатку Г.

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

Приклад змісту курсового проєкту наведено в додатку Д.

У **вступі** слід зазначити важливість використання інформаційних систем у сучасних умовах; роль інформаційних систем у предметній галузі, що розглядають у курсовому проєкті; роль засобів збереження

даних у інформаційних системах; мету та завдання курсового проєкту; відомості щодо розробленої програми (призначення, що вона дозволяє автоматизувати, технології та мову програмування, що було використано, під час розроблення програми), заключна частина.

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

У підрозділі 1.1. «Постановка завдання» наводять умову завдання на розроблення програмного продукту, відповідно до варіанта, який був виданим здобувачу вищої освіти. Опис умови завдання за будь-якою темою як мінімум має містити інформацію, аналогічну наведеній в додатку Е.

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

Функціональні вимоги явно описують, що має 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 бути максимальний час відгуку системи на запит користувача, мінімальний час безперебійної роботи системи, наявність графічного інтерфейсу користувача й т. ін.

Функціональні та нефункціональні вимоги до програмної системи, що розробляють відповідно до будь-якої теми курсового проєкту з переліку рекомендованих тем, наведено далі. В обґрунтованих випадках деякі вимоги може бути змінено за погодженням з науковим керівником курсового проєкту.

У будь-якому разі ці вимоги мусять мати стосунок до певної предметної галузі та конкретної програми, що є результатом проєкту.

Функціональні вимоги

1. Створення сховища даних і запис до нього даних у певному форматі.
2. Читання всіх даних зі сховища даних та їхнє відображення.
3. Додавання нового елементу даних до сховища даних.
4. Оновлення будь-якого елементу даних у сховищі.
5. Видалення будь-якого елементу даних зі сховища.

6. Здобуття та відображення підсумкової інформації.
7. Перевірка допустимості основних даних, що вводяться користувачем.
8. Видача користувачу попереджальних та інформаційних повідомлень.

Нефункціональні вимоги

Мінімальні вимоги до інтерфейсу користувача:

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

Під час проєктування інтерфейсу користувача слід дотримуватися таких правил:

перед проєктуванням інтерфейсу слід чітко визначити, які завдання будуть виконувати 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 «некоректних» дій користувача і запобігання їм; (3) виводити зрозумілі лаконічні інформаційні повідомлення в ситуаціях, коли це дійсно потрібно;

слід дотримуватися 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 ніж одну структуру даних зі стандартної бібліотеки колекцій.
2. Використання потоків введення-виведення даних.
3. Використання механізму винятків для опрацювання помилок.

Вимоги до вихідного коду застосунку:

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

2. Вихідний код кожного із класів програми має міститися в окремому файлі.

3. Ідентифікатори класів, інтерфейсів, полів класів, властивостей, методів, локальних змінних та констант тощо мусять мати смислове навантаження в певній предметній галузі.

Якщо здобувач вищої освіти вибрав тему курсового проєкту самостійно, то вимоги до програми обговорюють з науковим керівником та вони можуть відрізнятися від наведених.

У підрозділі 1.3. «Математичний опис задачі» має бути наведено математичну формалізацію задачі, тобто наявні співвідношення між величинами, які визначають залежність показників, що розраховують, від вхідних даних. Водночас, залежно від специфіки розв'язуваної задачі, може бути використано різні розділи математики та інших навчальних дисциплін. Таким способом формують математичну модель явища з певною точністю, припущеннями й обмеженнями.

Математична модель має задовольняти принаймні двом вимогам: реалістичності та реалізованості.

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

Реалізованості досягають розумною абстракцією, відволіканням від другорядних деталей, щоб звести задачу до задачі з відомим розв'язком. Умовою реалізованості є можливість практичного виконання необхідних обчислень за відведений час за доступних витрат потрібних ресурсів.

Отже, у цьому підрозділі слід навести математичні формули або логічні співвідношення, які визначають залежність показників, що розраховують, від вхідних даних, а також потрібні пояснення.

Другий розділ «Програмна документація» складається із 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 розробляють. Це може бути колекція з багатьох компонентів програмного забезпечення, один застосунок або частина застосунку.

Для уявлення статичної структури моделі програмної системи призначено UML-діаграму класів. Вона може відображати, зокрема, різні взаємозв'язки між окремими сутностями предметної галузі, як-от об'єкти й підсистеми, а також описує їхню внутрішню структуру та типи відносин.

У цьому підрозділі курсового проєкту потрібно навести:

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

2. Посилання на лістинг програми із вихідним кодом, що відповідає класам, які реалізують основну бізнес-логіку програмної системи. Лістинг програми має міститися в одному з додатків до курсового проєкту.

Підрозділ 2.2. «Тестування програмної системи». Тестування програмної системи – процес виконання програмного коду, спрямований на виявлення наявних у ньому дефектів. Під дефектом розуміють ділянку програмного коду, виконання якої за певних умов призводить до несподіваного поведження системи (тобто поведження, що не відповідає вимогам).

Завдання тестування – визначення умов, за яких виявляють 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 тестують як на відповідність вимогам, так і на відсутність проблемних ділянок програмного коду, які можуть викликати відмови й перебої в роботі системи.

Інтеграційне тестування. Окремі модулі зрідка 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. Після завершення інтеграційного тестування всі модулі системи є погодженими за інтерфейсами та функціональністю. Починаючи із цього моменту можна переходити до системного тестування, тобто тестування поведження системи загалом як єдиного об'єкта.

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

Системне тестування проводять в кілька етапів, на кожному з яких використовують один із видів системного тестування.

Важливим видом системного тестування є функціональне тестування. Цей вид системного тестування призначено для підтвердження того, що вся система загалом поводить ся, відповідно до очікувань користувача, формалізованих у вигляді системних вимог. У ході функціонального тестування перевіряють усі функції системи з погляду її користувачів (як людей, так і інших програмних систем). Також потрібно перевірити функціональну повноту інтерфейсу користувача й коректність виведення інформації.

Документування процедури тестування

Основне призначення документації, створеної під час тестування, – забезпечення гарантій того, що процес тестування виконують із потрібною якістю та всі аспекти поведження системи протестовано.

Перелік потрібної документації:

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

Тест-вимоги розробляють на підставі системних і функціональних вимог до застосунку. У них докладно описано, які аспекти поведження системи має бути протестовано, щоб упевнитися у її коректному функціонуванні, і на підставі якого зовнішнього ефекту можна переконатися, що функціональність, яку перевіряють, реалізовано правильно.

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

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

Для кожної тест-вимоги має бути можливість перевірки: чи виконують цю вимогу в реалізованій системі, чи ні.

На підставі тест-вимог створюють тест-план – документ, що містить докладний покроковий опис того, як має бути протестовано тест-вимоги. На відміну від тест-вимог, у тест-плані описують конкретні способи перевірки функціональності системи.

Зазвичай, тест-план складається з окремих тестових прикладів, кожний із яких перевіряє деяку функцію або набір функцій системи. Для

кожного тестового прикладу однозначно визначено критерій успішного проходження, за допомогою якого можна судити про відповідність поводження системи заданому.

Структура тест-плану має відповідати структурі тест-вимог.

Кожний пункт тест-плану має містити:

1. Посилання на вимогу(и), що перевіряє цей пункт.
2. Конкретне значення вхідних даних.
3. Очікувану реакцію програми (тексти повідомлень, значення результатів).
4. Опис послідовності дій, потрібних для виконання пунктів тест-плану.

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

Іноді тест-план поєднують зі звітом про проведення тестування, додаючи до нього інформацію про досягнену реакцію системи та збіг (розбіжності) реальних результатів з очікуваними. Наприкінці опису кожного тестового прикладу додають інформацію про те, чи пройдений тестовий приклад загалом. Наприкінці всього тест-плану, поєднаного зі звітом, міститься графа «Тестових прикладів пройдено/усього», у яку заносять кількість пройдених тестових прикладів і загальна їхня кількість.

Приклад тест-плану, поєднаного зі звітом про проведення тестування наведено в додатку И.

У цьому пункті курсового проєкту має міститися опис процедур функціонального тестування та їхніх результатів, що належать до певної предметної галузі.

Для опису процедури тестування має бути укладено такі документи:

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

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

Однак загальний результат тестування застосунку має бути позитивним, бо інакше він не відповідає певним вимогам. Для цього в разі потреби проводять додаткові заходи щодо виправлення помилок і тестування. Їх також має бути описано в цьому пункті курсового проєкту.

У підрозділі 2.3. «Розгортання програмного продукту» слід навести опис вимог до апаратних та програмних засобів, потрібних для функціонування розробленого програмного продукту, та дій щодо його інсталяції на комп'ютері користувача.

Розгортання – це процес поширення готового застосунку або компонента для встановлення на інші комп'ютери. 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 в процесі роботи програми.

У пункті 2.4.1. «Призначення програмного продукту» вказують відомості про його призначення та інформацію, достатню для розуміння функцій програмного продукту.

У пункті 2.4.2. «Використання програмного продукту» має бути вказано 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 їхнього змісту.

Приклад керівництва користувача наведено в додатку Л.

У **висновках** наводять оцінку визначених 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 в роботі. Джерела можна розміщувати в списку одним із таких способів: у порядку появи посилань у тексті, алфавітному порядку прізвищ перших авторів або заголовків. Вимоги до оформлення списку використаних джерел наведено у [1].

У **додатках** уміщують матеріал, який є потрібним для повноти курсового проєкту, але не може бути послідовно розміщеним у її основній частині через великий обсяг або способи відтворення та з інших причин. Рисунки (діаграми бізнес-процесів, схеми алгоритмів, технологічних процесів, сценарії діалогів та ін.), таблиці, проміжні математичні докази,

формули й розрахунки, текст допоміжного характеру та інші матеріали можуть бути оформленими у вигляді додатків.

5. Вимоги до оформлення курсового проєкту

Важливе значення під час роботи над курсовим проєктом має його оформлення, до якого ставлять певні вимоги.

Для коректного оформлення тексту курсового проєкту слід керуватися державним стандартом України ДСТУ 3008-2015 «Документація. Звіти у сфері науки і техніки. Структура і правила оформлення» [2]. Далі наведено загальні вимоги до оформлення курсового проєкту.

Матеріали курсового проєкту друкують на аркушах формату А4. Їхній текст має відповідати правилам граматики й стилістики.

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

Текст документа має бути виконаним із використанням шрифту Times New Roman (розмір 14), міжрядковим інтервалом 1,2 (37 рядків на сторінці). Найменшим розміром шрифту може бути розмір 10 (його можна використовувати для подання вихідного тексту програм). Шрифт друкування має бути чітким, текст – чорного кольору середньої жирності. Кольорове друкування дозволяють використовувати лише для рисунків (екранні форми, діаграми тощо). Щільність тексту має бути однаковою. Вирівнювання основного тексту виконують за шириною сторінки.

Увесь текст курсового проєкту, а також назви її структурних елементів, виконують шрифтом однакової жирності. Не дозволено використання курсиву та підкреслення.

Абзацний відступ має бути однаковим упродовж усього тексту та дорівнювати 1,25 см.

Друкарські помилки, описки та графічні неточності, виявлені в процесі підготовки курсового проєкту, можна виправляти підчищенням або коректором і нанесенням на тому самому місці виправленого тексту. Допускають наявність не більше ніж два виправлення на одній сторінці.

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

Кожний структурний елемент курсового проєкту (за винятком підрозділів, пунктів, підпунктів) має 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 та друкувати маленькими літерами, крім першої великої, не підкреслюючи, без крапки в кінці.

Переноси в середині слова в заголовках не допускають.

Не дозволено 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 рядками заголовка, а також між двома заголовками беруть такою, як у тексті.

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

Сторінки курсового проєкту має бути пронумеровано арабськими цифрами в правому верхньому куті без крапки. Нумерація усіх сторінок (а також ілюстрацій, таблиць, графіків) є наскрізною від титульного аркуша до останнього аркуша тексту. На титульному аркуші, завданні на курсовий проєкт нумерацію сторінок не проставляють.

Викладені в тексті матеріали мають наочно доповнювати й підтверджувати рисунки (схеми, ілюстрації, графіки, діаграми). Рисунки мають відображати тему курсового проєкту. Здобувачеві вищої освіти потрібно продумати, який матеріал проілюструвати. Це діаграми класів, схеми алгоритмів, схеми інформаційних зв'язків тощо.

Усі ілюстрації називають рисунками, їм надають порядковий номер у межах номера розділу. Підпис рисунку складається із слова «Рисунок», його номера та 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 рисунки мають бути посилання у тексті.

Посилання на рисунки вказують його порядковим номером, наприклад, «рис. 1.2». У повторних посиланнях на рисунки треба вказувати скорочено слово «дивись», наприклад: (див. рис. 1.2).

У тому місці, де викладають матеріал, пов'язаний з рисунком, розміщують посилання у вигляді виразу в 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 нумерують у межах розділу.

Пояснення до умовних літерних позначень у формулі наводять зразу ж під формулою. Для цього після формули ставлять кому і записують пояснення до кожного символу з нового рядка в тій послідовності, у якій їх наведено у формулі, розділяючи крапкою з комою. Перший рядок має починатися з абзацу зі слова «де».

Наприклад,

$$I = UR, \tag{2.1}$$

де U – електрична напруга;

R – сила електричного струму.

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

Таблицю потрібно розташовувати безпосередньо після тексту, у якому її згадують уперше або на наступній сторінці. На всі таблиці мають бути посилання. Таблиці послідовно нумерують у межах розділу. Над лівим верхнім кутом таблиці міститься напис «Таблиця» зі вказівкою її порядкового номера та назви (наприклад, Таблиця 1.1. Структурні елементи курсового проєкту).

Переліки, за потреби, може бути наведено 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 курсового проекту мають бути посилання на літературу. Водночас наводять її порядковий номер, записаний у квадратних дужках (наприклад, «... у роботах [1 – 7]»).

Додатки потрібно оформляти як продовження курсового проекту на її наступних сторінках. Кожен додаток має починатися з нової сторінки. Додаток мусить мати заголовок, надрукований угорі малими літерами з першої великої симетрично щодо тексту сторінки. Посередині рядка над заголовком малими літерами з першої великої має бути надруковано слово «Додаток» і велика літера, що позначає додаток (наприклад, «Додаток А»).

Додатки позначають послідовно великими літерами української абетки за винятком букв Ґ, Є, З, І, Ї, Й, О, Ч, Ь. Один додаток позначають як додаток А.

Рекомендована література

Основна

1. ДСТУ ГОСТ 7.1 – 2006. Бібліографічний запис. Бібліографічний опис. Загальні вимоги та правила складання : чинний з 2007-07-01. – Київ : 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. ДСТУ 3008-2015. Інформація та документація. Звіти у сфері науки і техніки. Правила оформлення : чинний з 2017-07-01. – Київ : 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-Wesly, 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.

Інформаційні ресурси

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

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

8. C# documentation [Electronic resource]. – Access mode : <https://docs.microsoft.com/en-us/dotnet/csharp/>.

9. C# Tutorial [Electronic resource]. – Access mode : <https://www.tutorialspoint.com/csharp/index.htm>.

10. Java Tutorial [Electronic resource]. – Access mode : <https://www.w3schools.com/java/default.asp>.

11. JDK 20 Documentation [Electronic resource]. – Access mode : <https://docs.oracle.com/en/java/javase/20/>.

12. What are UML diagrams, and how can you use them [Electronic resource]. – Access mode : <https://miro.com/blog/uml-diagram/>.

Додатки

Додаток А

Перелік рекомендованих тем курсових проєктів

1. Розроблення програмного продукту для роботи з даними про надходження 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 рух 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 продаж книг у книгарні.

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

22. Розроблення програмного продукту для роботи з даними про витрати 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 здобувачами вищої освіти.

28. Розроблення програмного продукту для роботи з даними про постачання 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 здобувачів вищої освіти.

35. Розроблення програмного продукту для роботи з даними про облік оплати телефонних розмов.

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 здобувачами вищої освіти літератури в бібліотеці факультету.

40. Розроблення програмного продукту для роботи з даними про рух товарів на складі торгового підприємства.

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

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

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

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

45. Розроблення програмного продукту для роботи з даними про замовлення клієнтів.

46. Розроблення програмного продукту для роботи з даними про публікації здобувачів вищої освіти.

Зразок завдання на курсовий проєкт

**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ СЕМЕНА КУЗНЕЦЯ**

Науково-навчальний інститут інформаційних технологій
Кафедра інформаційних систем

ЗАВДАННЯ

на курсовий проєкт: Об'єктно-орієнтоване програмування
здобувачу вищої освіти II курсу групи 6.04.121.010.23.01
Вареникові Григорію Васильовичу

1. Тема проєкту: «Розроблення програмної системи для автоматизації роботи з даними про <Назва предметної галузі>».
2. Термін здавання здобувачем вищої освіти закінченого проєкту «__» _____ 202__ р.
3. Вхідні дані до проєкту: наукові джерела, технічна документація щодо розроблення програм, ДСТУ з оформлення документації.
4. Зміст курсового проєкту:
Вступ. Специфікація проєкту. Програмна документація. Висновки. Додатки.
5. Перелік графічного матеріалу: UML-діаграма класів програмної системи.

Календарний план
виконання курсового проєкту

№ з/п	Назви етапів роботи	Терміни виконання за планом	Позначка виконання
1	З'ясування загальної постановки завдання. Розроблення чернетки першого розділу курсового проєкту		
2	Деталізація завдань курсового проєктування. Уточнення архітектури програмної системи. Розроблення остаточного варіанта першого розділу курсового проєкту		
3	Програмна реалізація, налагодження та тестування застосунку		
4	Остаточне налагодження та тестування застосунку. Розроблення чернетки другого розділу курсового проєкту, висновків, списку використаних джерел, додатків, вступу, реферату		
5	Розроблення остаточного варіанта курсового проєкту та підготовка електронної презентації		

Дата видачі завдання «__» _____ 202_ р.

Науковий керівник канд. техн. наук, доцент кафедри ІС _____ О. І. Петренко
(підпис)

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

Зразок оформлення титульного аркуша

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ СЕМЕНА КУЗНЕЦЯ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ

**КУРСОВИЙ ПРОЄКТ: ОБ'ЄКТНО-ОРІЄНТОВАНЕ
ПРОГРАМУВАННЯ**

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

Здобувач вищої освіти II курсу
6.04.121.010.21.01 групи
Спеціальності 121 «Інженерія програмного
забезпечення»
першого (бакалаврського) рівня
Вареник Г. В.
Науковий керівник: доцент кафедри ІС,
канд. техн. наук, доцент
Петренко О. І.

Національна шкала _____
Кількість балів: _____ Оцінка: ECTS _____

Члени комісії

_____	_____
(підпис)	(прізвище та ініціали)
_____	_____
(підпис)	(прізвище та ініціали)

Харків – 202_

Зразок оформлення реферату

РЕФЕРАТ

Курсовий проєкт налічує 32 с., 20 рис., 3 табл., 6 джерел.

Об'єктами проєктування 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 проєктування – 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 використано на підприємствах, які надають послуги доступу до мережі «Інтернет».

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

Зразок оформлення змісту курсового проєкту**ЗМІСТ**

Вступ.....	6
1. Специфікація проєкту.....	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 умови завдання

Підприємство веде неавтоматизований облік яблук із використанням відомості в «паперовому» вигляді. Кількість записів відомості є не обмеженою.

Відомість має табличну структуру із 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 обчислюють за ф-лою (1.1):

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

де i – номер запису відомості.

Приклад заповненої відомості наведено у табл. 1.1.

Таблиця 1.1

Приклад заповненої відомості

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 через унесення до відповідних полів запису відомості початкових даних та даних, які було розраховано за ф-лою (1.1).

Наприкінці звітної періоду співробітник підприємства обчислює середній об'єм яблука V_{cp} за ф-лою (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 у відомості;

n – кількість яблук у відомості.

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

Приклад опису UML-діаграми класів

Діаграму класів програми наведено на рис. 2.1.

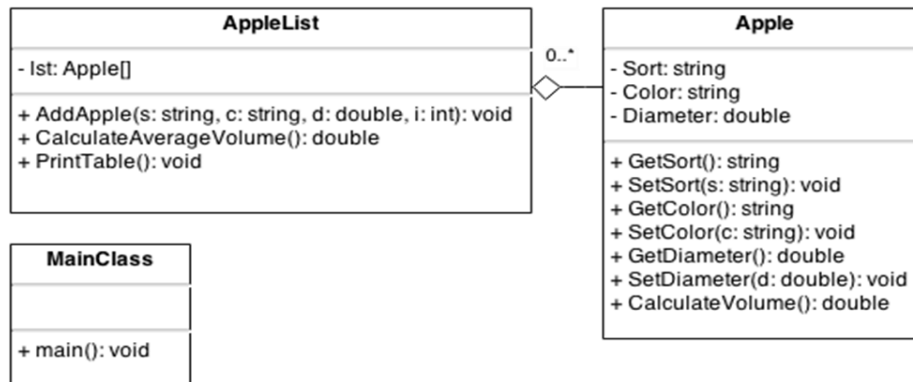


Рис. 2.1. UML-діаграма класів

Програма складається із 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 діаметра яблука, метод GetColor() повертає поточне значення його кольору. Метод SetSort() установлює 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 встановлено відношення агрегації: поле Ist класу AppleList є масивом 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() потрібний для додавання об'єкта-«яблука» в масив Ist. Метод CalculateAverageVolume() – обчислює середній об'єм яблука у відомості. Призначення методу PrintTable() – «друкування» значень полів об'єктів-«яблук», діаметра яблука і підсумкових даних відомості.

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

Приклад тест-плану, поєднаного 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 коректно відображати їх в інтерфейсі користувача.

Тест-вимоги, що перевіряє приклад: функціональна вимога № 2

Передумови для тесту: програмну систему має бути запущено, на диску комп'ютера має бути наявним файл із даними у визначеному форматі.

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

Звіт про виконання тестування представлено у таблиці 1.

Таблиця 1

Звіт про виконання тестування

№ з/п	Кроки сценарію	Очікуваний результат	Реальний результат	Позначка 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 коректно відображатися завантажені дані	Дані, що були завантаженими з файлу, коректно відображаються в консольному вікні програми	Так

Закінчення додатка И

Закінчення табл. 1

1	2	3	4	5
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 завантаженими	З'являється повідомлення про те, що дані не можуть бути завантаженими	Так

Позначка 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 – двоядерний, 2 ГГц або більшої швидкості.
2. Доступний простір ОЗУ – не менше ніж 4 Гб.
3. Вільний дисковий простір – не менше ніж 25 Гб.

Вимоги до програмних засобів:

1. Операційна система:
 - Windows 10, 11.
 - Більшість сучасних дистрибутивів Linux.
 - Mac OSX10.14 або більш нова версія.
2. Java Runtime Environment 17 або більш нова версія.

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

1. Створіть на цільовому диску каталог для застосунку, наприклад MyApp.
 2. У каталозі MyApp створіть новий каталог, наприклад dist.
 3. Скопіюйте виконуваний jar-файл застосунку (наприклад install.jar) у каталог dist.
 4. Завантажте з вебсайту компанії Oracle програмну платформу Java Runtime Environment потрібної версії та розпакуйте відповідний архів у деякий каталог, наприклад у папку appjre.
 5. Перемістіть каталог appjre у каталог MyApp.
 6. У каталозі MyApp створіть командний файл цільової операційної системи, наприклад start.bat (операційна система Windows).
 7. Додайте в start.bat такі команди (операційна система Windows):
@echo OFF
set PATH =.\appjre\bin
java -jar dist\install.jar
pause> NUL
- Для перевірки коректності запуску програми виконайте подвійне клацання лівою кнопкою миші на файлі start.bat (ОС Windows).

Приклад керівництва користувача (скорочено)

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 здійснюють одним зі 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 із подальшим вибором пункту «Усі програми» та подвійним клацанням лівою кнопкою миші на ярлику програми.

Основні елементи інтерфейсу користувача програми

Інтерфейс користувача 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 із програмою

Уведення початкових даних

Для початку роботи потрібно вибрати пункт головного меню Enter initial data через введення цифри 1 та подальшого натискання клавіші <Enter> на клавіатурі. Після цього на екран послідовно виводять повідомлення для введення кількості яблук в облікової відомості, а також сорту, кольору та діаметру кожного яблука (рис. 2.2). Після введення значення кожного з параметрів слід 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 через 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 відображають значення сорту, кольору, діаметра та обчисленого об'єму. У рядку Total користувач програми має змогу побачити значення середнього об'єму яблука з наявних у таблиці.

Слід зазначити, що після вибору цього пункту головного меню таблиця з'являється тільки в тому разі, якщо користувач програми заздалегідь увів початкові дані щонайменш як 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 із програмою

Для цього потрібно 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 вводив початкових даних, на екрані комп'ютера з'явиться таке повідомлення (рис. 2.4):

2
No data to print!

Рис. 2.4. Повідомлення за відсутності даних для відображення

Зміст

Вступ.....	3
1. Мета й завдання курсового проєктування	4
2. Організація курсового проєктування	5
3. Структура й обсяг курсового проєкту	7
4. Методичні рекомендації до розроблення структурних елементів курсвого проєкту.....	8
5. Вимоги до оформлення курсового проєкту.....	17
Рекомендована література.....	20
Основна	20
Додаткова	21
Інформаційні ресурси	21
Додатки.....	22

НАВЧАЛЬНЕ ВИДАННЯ

ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ

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

Самостійне електронне текстове мережеве видання

Укладачі: **Парфьонов** Юрій Едуардович
Гризун Людмила Едуардівна

Відповідальний за видання *Д. О. Бондаренко*

Редактор *О. Г. Доценко*

Коректор *В. О. Дмитрієва*

План 2024 р. Поз. № 78 ЕВ. Обсяг 40 с.

Видавець і виготовлювач – ХНЕУ ім. С. Кузнеця, 61165, м. Харків, просп. Науки, 9-А

*Свідоцтво про внесення суб'єкта видавничої справи до Державного реєстру
ДК № 4853 від 20.02.2015 р.*