

МЕТОДИЧНІ ВКАЗІВКИ

до виконання курсової роботи з дисципліни

"ЗАХИСТ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ"

для студентів спеціальності 125 «Кібербезпека»
спеціалізації «Безпека інформаційних і комунікаційних систем»

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

МЕТОДИЧНІ ВКАЗІВКИ

до виконання курсової роботи з дисципліни

"ЗАХИСТ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ"

для студентів спеціальності 125 «Кібербезпека»
спеціалізації «Безпека інформаційних і комунікаційних систем»

Вінниця
ВНТУ
2018

Рекомендовано до друку Методичною радою Вінницького національного технічного університету Міністерства освіти і науки України (Протокол № 7 від 28.03.2018 р.)

Рецензенти:

В. П. Майданюк, кандидат технічних наук, доцент

Ю. В. Булига, кандидат технічних наук, доцент

Методичні вказівки до виконання курсової роботи з дисципліни «Захист програмного забезпечення» для студентів спеціальності 125 «Кібербезпека» спеціалізації «Безпека інформаційних і комунікаційних систем» / Уклад.: В. А. Каплун, В. В. Лукічов. – Вінниця: ВНТУ, 2018. – 36 с.

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

Навчальне самостійне електронне мережне видання

Методичні вказівки
до виконання курсової роботи
з дисципліни «Захист програмного забезпечення»
для студентів спеціальності 125 «Кібербезпека»
спеціалізації «Безпека інформаційних і комунікаційних систем»

Укладачі:

Каплун Валентина Аполінаріївна

Лукічов Віталій Володимирович

Електронний ресурс PDF.

Підписано до видання 25.07.2018 р. Зам. № P2018-014

Видавець та виготовлювач - Вінницький національний технічний університет,

Інформаційний редакційно-видавничий центр. ВНТУ, ГНК, к.114,

Хмельницьке шосе, 95, м. Вінниця, 21021,

тел. (0432) 65-18-06.

press.vntu.edu.ua;

Email: irvc.vntu@gmail.com.

Свідоцтво суб'єкта видавничої справи

серія ДК № 3516 від 01.07.2009 р.

ЗМІСТ

1	ОСНОВНІ ВИМОГИ ДО КУРСОВОЇ РОБОТИ.....	4
1.1	Мета і задачі курсової роботи.....	4
1.2	Тематика курсової роботи.....	5
1.3	Вимоги до розробки програмного засобу.....	6
3	ОФОРМЛЕННЯ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ.....	8
3.1	Загальні вимоги до оформлення.....	8
3.2	Структура пояснювальної записки.....	10
3.3	Вступна частина пояснювальної записки.....	11
3.4	Вміст і оформлення основної частини.....	13
3.5	Оформлення додатків.....	19
3.6	Оформлення ілюстративної частини.....	20
4	ГРАФІК ВИКОНАННЯ КУРСОВОЇ РОБОТИ І ПОРЯДОК ЗАХИСТУ.....	21
4.1	Графік виконання.....	21
4.2	Оцінювання виконання курсової роботи.....	22
	ПЕРЕЛІК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ.....	23
	Додаток А. Варіанти завдань на курсову роботу.....	24
	Додаток Б. Приклад оформлення титульного аркуша.....	26
	Додаток В. Приклад оформлення індивідуального завдання.....	27
	Додаток Д. Приклади UML-діаграм.....	33

1 ОСНОВНІ ВИМОГИ ДО КУРСОВОЇ РОБОТИ

1.1 Мета і задачі курсової роботи

Курсова робота (КР) – навчальна робота з дисципліни, яка містить елементи ескізного і технічного проектів та робочої документації.

В курсовій роботі з дисципліни «Захист програмного забезпечення» студент повинен показати свої знання в галузі захисту програмних продуктів від нелегального копіювання і використання програм, несанкціонованого використання і дослідження, знання нових технологій захисту, вміння самостійно розробити систему захисту від атак певного виду, підібрати засоби для його реалізації та вміння подати розробку у вигляді, зручному для його використання стороннім користувачем.

Під час виконання курсової роботи студенти повинні використати знання, отримані ними під час вивчення дисциплін “Технологія програмування”, “Захист операційних систем”, “Основи інформаційної безпеки”, “Прикладна криптологія”.

Для успішного виконання курсової роботи студенти повинні вміти:

- коректно сформулювати задачу для розв’язання її на певному типі обчислювальних машин, у певному операційному середовищі;
- правильно визначити механізми захисту (вбудовані або навісні) та виконувані функції;
- визначити конкретні методи захисту для можливості їх застосування відповідно до умов використання програмного забезпечення і придатні для конкретного типу задач (захист інформації від нелегального копіювання, від несанкціонованого вивчення або дослідження програм);
- звести поставлення задачі до розробки алгоритму і визначити структури даних, які дозволяють перейти від абстрактного формулювання алгоритму до конструювання структурної схеми;
- здійснювати програмну реалізацію винайденого типу захисту, використовуючи як мови високого рівня, так і мову асемблера;
- довести доцільність використання розробленого захисту, визначити коло програм і задач, для яких даний метод може бути застосований;
- довести ефективність обраного способу захисту, використовуючи при цьому дизасемблери та налагоджувачі;
- подати власну розробку таким чином, щоб нею могли користуватись інші, підготувавши для цього відповідний методичний матеріал у вигляді детальних інструкцій та рекомендацій.

Завдання для курсових робіт визначаються викладачем із загального списку завдань, затверджених кафедрою захисту інформації. Заохочуються пропозиції студентів щодо самостійного, за узгодженням з викладачем, вибору теми курсового проекту поза межами запропонованого в методичних вказівках переліку. Самостійний вибір предметної області, в

якій доцільно використовувати сучасні методи захисту програм та оригінальні алгоритми, дозволяє зробити висновок щодо рівня творчої активності студента, його вміння самостійно здійснити попередній аналіз предметної області і розробити технічне завдання.

Метою індивідуальних завдань є закріплення теоретичних та практичних навичок у роботі з використання сучасних технологій захисту програм шляхом використання сучасних мов і технологій програмування.

1.2 Тематика курсової роботи

Зміст курсової роботи відповідає навчальній програмі та робочому плану дисципліни “Захист програмного забезпечення” і визначається завданням, яке видається не пізніше шести днів з початку семестру на консультації викладачем кожному студенту.

Курсове проектування включає декілька послідовних етапів, які повинні бути пов'язані зі змістовним постановленням задачі, розробкою індивідуального та технічного завдання, варіантним аналізом методів захисту, вибором форми подання результатів, математичною моделлю рішення, вибором оптимального алгоритму для реалізації захисту, проведенням досліджень створеної програми та формулюванням обґрунтованих висновків щодо ефективності розробленого захисту.

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

Тематика курсової роботи пов'язана з майбутньою спеціальністю студентів. Для програмної реалізації даного курсової роботи пропонується ряд основних методів, які реалізують деякий вид захисту:

- захист програм від несанкціонованого копіювання;
- захист від статичного та динамічного дослідження;
- захист від дампінгу;
- захист від несанкціонованого доступу і використання;
- стеганографічні методи захисту;
- захист форм web-ресурсів і захист контенту web-сторінок;
- інші методи захисту.

Крім того об'єктом для курсової роботи можуть бути дослідження та впровадження відомих систем захисту, реалізація тестових програм, розробка лабораторних практикумів для даної дисципліни та для інших дисциплін, пов'язаних із захистом програмного забезпечення, тощо.

Метою проектування повинен бути вибір такого методу захисту, що забезпечить неможливість несанкціонованого копіювання для заздальгідь

визначеного кола осіб і на обмежений час. Таким чином, для вибору способу організації захисту необхідним є індивідуальний підхід у кожному конкретному випадку.

Студент, вибравши собі завдання для курсового проектування, повинен чітко уявляти собі, що, як і від кого він збирається захищати, тобто визначитись з метою і об'єктом захисту, механізмом, принципом і методом захисту.

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

1.3 Вимоги до розробки програмного засобу

Результатом виконання курсової роботи повинен бути, як правило, повноцінний програмний додаток для конкретної операційної системи (Windows, Unix, Android, IOS), призначений для здійснення захисту деякого об'єкта.

Крім того, курсова робота може бути систематизацією наукових досягнень і програмних розробок у певній області захисту програмного забезпечення і подання їх у вигляді методичної документації для проведення лабораторних робіт з дисципліни "Захист програмного забезпечення" або інших дисциплін. Але і в цьому випадку результатом розробки повинен бути повноцінний цілісний програмний засіб або методичні рекомендації по певній темі.

Розроблена програма повинна відповідати таким вимогам.

1. *Реалізація алгоритму захисту інформації* для будь-якого відомого методу захисту або власного алгоритму захисту повинна бути здійснена на основі глибокого і ретельного аналізу літературних джерел з вибраної теми. Можливість використання та доцільність застосування вибраного методу захисту повинні бути всебічно обґрунтованими. Розроблений алгоритм повинен бути порівняний з іншими методами захисту, наведено його переваги та недоліки.
2. *Об'єктом захисту* може бути вихідний код програми певною мовою програмування, виконуваний код програми (exe-файл, функції DLL-бібліотеки). Тобто, захист може бути як вбудований (внутрішній), так і навісний (зовнішній). Механізм захисту повинен бути обумовлений заздалегідь і визначений в індивідуальному та технічному завданнях.
3. *При програмній реалізації задачі* допускається застосування будь-яких

мов програмування (Pascal, C++, Java, C#, Assembler, Delphi тощо) та будь-якого візуального середовища програмування, причому вибір тієї або іншої мови програмування і середовища програмування повинні бути обґрунтовані.

4. *Реалізація дружнього інтерфейсу є обов'язковою належністю грамотно розробленої програми: меню, елементів керування, попередження про можливі помилки при введенні інформації або під час інтерактивного режиму роботи і т. д. Всі результати вхідних, проміжних, результуючих дій повинні бути виведені на екран у вигляді, зручному для аналізу і розуміння стороннім користувачем.*
5. *В результаті реалізації поставленої задачі необхідно виконати аналіз роботи розробленої програми, здійснити тестування її роботи у різних передбачених програмою режимах, навести висновки щодо ефективності розробленого захисту, можливих об'єктів захисту, умов використання вибраного методу захисту, його недоліків і переваг.*
6. *В ілюстративній частині окрім схем функціонування системи захисту, схем взаємодії програм, блок-схем конкретних алгоритмів, схем даних тощо необхідно подати вигляд фрагментів інтерфейсу як доведення працездатності розробленого програмного засобу.*
7. *Розробка інструктивних документів для роботи з програмою є обов'язковою складовою пояснювальної записки до курсової роботи .*

3 ОФОРМЛЕННЯ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

3.1 Загальні вимоги до оформлення

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

Шрифт і відступи

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

Нумерація сторінок.

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

Нумерація додатків продовжує основну нумерацію.

Оформлення розділів і підрозділів

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

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

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

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

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

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

Переліки

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

Основні проблеми такі:

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

Для подальшої деталізації використовують арабські цифри з дужкою.

Оформлення таблиць

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

Таблиця 1.1 – Основні алгоритми захисту

Алгоритм захисту	Опис
1. Алгоритм перемішування	Використання хаотичних переходів в різні частини коду, впровадження помилкових процедур-"пустишок", холостих циклів тощо
2. Алгоритми мутації	Створення таблиць відповідності операндів-синонімів і заміна їх між при кожному запуску програми за певною схемою

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

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

Продовження таблиці 1

2	3
3. Алгоритм перемішування	Програма упаковується, а потім розпаковується під час виконання
4. Алгоритми мутації	Програма шифрується, а потім розшифровується під час виконання

Оформлення рисунків

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

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

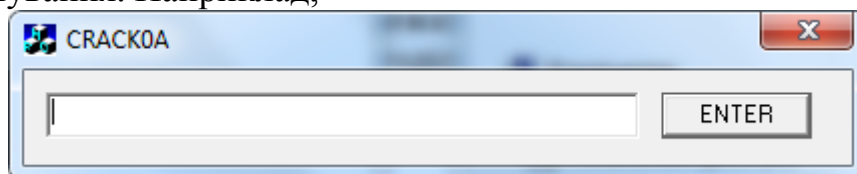


Рисунок 1.2 – Вигляд екрана при перевірці пароля

Між ілюстрацією і текстом пропускають один рядок.

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

Оформлення формул

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

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

Всі формули нумерують в межах розділу арабськими числами. Номер вказують в круглих дужках з правої сторони, в кінці рядка, на рівні закінчення формули. Номер формули складається з номера розділу і порядкового номера формули в розділі, розділених крапкою. Дозволяється виконувати нумерацію в межах всього документа.

3.2 Структура пояснювальної записки

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

врахуванням всіх офіційних змін, введених в дію.

Пояснювальна записка до курсової роботи повинна мати таку структуру.

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

3.3 Вступна частина пояснювальної записки

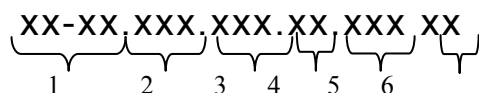
Вступна частина повинна містити:

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

Титульний аркуш

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

Система умовних позначень має таку структуру:



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

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

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

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

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

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

Індивідуальне завдання

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

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

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

бути наведені вхідні і вихідні дані проекту.

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

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

Анотація

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

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

Далі розміщують анотація англійською мовою з заголовком ABSTRACT. Вона може розташовуватись на тому самому або на окремому аркуші.

Зміст

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

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

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

Приклад оформлення змісту:

ВСТУП

1 РОЗРОБКА ...

1.1 Варіанти ...

1.1.1 ...

.....
2 ЗАГОЛОВОК ДРУГОГО РОЗДІЛУ

2.1 Заголовки підрозділів

2.1.1 ...

.....

.....

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

Додаток А. Назва першого додатка

Додаток Б. Назва другого додатка

.....

3.4 Вміст і оформлення основної частини

Основна частина складається з таких складових:

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

3.4.1 Вступ

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

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

Вступ повинен стисло висвітлювати такі питання:

- стан розвитку проблеми в даній галузі, до якої має відношення розробка;
- галузь використання та призначення даної розробки;
- мету та загальну постановку задачі;
- актуальність, яка повинна подаватись в останньому абзаці вступу з метою стислого викладання суті обраної розробки.

Обсяг вступу не повинен перевищувати 1-2 сторінок.

3.4.2 Аналіз сучасного стану питання та обґрунтування теми

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

Рекомендований обсяг розділу – 4-5 сторінок (не більше!).

3.4.3 Реалізація основної задачі курсової роботи

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

Таким чином, в цьому розділі рекомендується висвітити такі питання:

- формулювання основних вимог до показників розроблюваної системи захисту, до інтерфейсу;
- визначення кола програмних, технічних та інших засобів, необхідних для реалізації поставленої задачі і обґрунтування доцільності використання саме цих засобів;
- наведення і пояснення математичної моделі (якщо в цьому є необхідність),
- визначення послідовності основних кроків, необхідних для реалізації обраної розробки, і детальний опис дій при реалізації кожного кроку;
- програмна реалізація задачі (у випадку, якщо метою проекту є розробка програмного засобу);
- підготовка інформаційного наповнення та підбору досліджуваних об'єктів (якщо метою проекту є підготовка матеріалів для лабораторних робіт);
- здійснення процесу тестування розробки і розробка рекомендацій щодо роботи з нею.

Якщо використовуються деякі особливі методи або логічні рішення поставленої задачі, їх теж потрібно описати, але детальніше.

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

У випадку, якщо результатом курсової роботи повинен бути програмний продукт, окремими підпунктами бажано описати розробку

загальної схеми функціонування програми або схему роботи програми. Для окремих модулів, що входять до складу програмного засобу, необхідно розробити блок-схеми алгоритмів, структурні схеми з наведенням основних блоків, покрокове викладення алгоритму з розглядуванням і т. ін.

У випадку, якщо курсова робота присвячений дослідженню методів захисту, необхідно навести алгоритм дослідження, схему зв'язків між частинами інформаційного наповнення, схеми подання матеріалу, схеми даних, таблиці з основними характеристиками і функціональними особливостями досліджуваних об'єктів, порівняльними характеристиками, перевагами, недоліками і т. д.

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

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

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

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

3.4.4 Використання схем і діаграм

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

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

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

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

- діаграма варіантів використання;
- діаграма класів;
- діаграма станів;
- діаграма діяльності;

- діаграма послідовності;
- діаграма компонентів тощо.

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

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

3.4.5 Тестування програми і розробка інструкцій

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

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

Якщо темою курсової роботи є проведення досліджень, то цей розділ повинен продемонструвати повноцінну роботу досліджуваних програм, порівняльний аналіз повинен бути зрозумілим, обґрунтованим. При цьому вигляд вмісту вікон досліджуваних об'єктів та інших складових інтерфейсу може бути винесений у додатки.

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

Інструкція з технічного обслуговування

Інструкція з технічного обслуговування повинна містити такі підпункти :

- *вступ* (призначення інструкцій, перелік експлуатаційних документів, якими повинні додатково до інструкцій користуватися при технічному обслуговуванні і експлуатації);
- *загальні вказівки* (порядок технічного обслуговування, вказівки щодо організації і особливостей його проведення);

- *вимоги до технічних засобів* (вказують мінімальний склад технічних засобів, що забезпечують роботу програмного засобу);
- *опис функцій* (максимальний перелік функцій, що здійснюються цією програмою; опис сумісного функціонування технічних засобів і програми з вказанням методу обробки помилок; опис організації вхідних і вихідних даних для перевірки роботоздатності; опис взаємодій пристроїв з програмою, результатів взаємодій, висновки з результатів роботи програми).

Інструкція системного програміста

Інструкція системного програміста повинна містити такі підпункти:

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

Інструкція програміста

Інструкція програміста повинна містити такі підпункти:

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

виконання програми, опис їх змісту і дій, які слід зробити у відповідь на ці повідомлення).

Інструкція оператора

Інструкція оператора повинна містити такі:

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

3.4.6 Висновки

Висновки оформляють з нової пронумерованої сторінки, починаючи зі слова “ВИСНОВКИ” посередині великими літерами, після чого пропускається один рядок. У висновках наводяться основні результати роботи над курсовим проектом. Коротко по основних розділах описуються етапи реалізації задачі курсової роботи. На основі проведених досліджень даються обґрунтовані висновки щодо переваг та недоліків застосування того чи іншого методу захисту, того чи іншого засобу при здійсненні реалізації задачі, недоліки та переваги розробки, труднощі при розробці програми та причини, що їх обумовили, і можливі шляхи їх подолання. Обов’язково слід навести можливі рекомендації прикладного застосування, об’єкти захисту, умови використання даного способу захисту програм та шляхи (перспективи) удосконалення розробленого програмного забезпечення.

3.4.7 Перелік використаних джерел

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

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

Кожне джерело повинно бути вказано разом з видавництвом, роком видання, кількістю сторінок. Літературу записують мовою оригіналу. У списку кожне джерело записують з абзацу, нумерують арабськими цифрами, починаючи з одиниці (далі показано). Правильне оформлення певного джерела інформації можна переглянути у переліку літературних джерел у будь-якому навчальному посібнику. Якщо у списку використаних джерел є посилання на Інтернет-сторінки, слід наводити разом з назвою Інтернет-сторінки.

Приклад оформлення переліку використаних джерел різного характеру:

Приклад посилання на книги:

1. Мамаев М., Технология защиты информации в интернете : [Специальный справочник] / Максим Мамаев, Сергей Петренко. - СПб. : Питер, 2002. – 848 с.
2. Лужецький В. А. Інформаційна безпека : навчальний посібник / В. А. Лужецький, О. П. Войтович, А. В. Дудатьєв – Вінниця : УНІВЕРСУМ-Вінниця, 2009. – 240 с.

Приклад посилання на журнали:

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

Приклад посилання на ГОСТ і ДСТУ:

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

Приклад посилання на патенти:

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

Приклад посилання на web-сторінки:

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

3.5 Оформлення додатків

У додатках вміщують матеріал, який:

- є необхідним для повноти звіту, але включення його до основної частини звіту може змінити впорядковане й логічне уявлення про

роботу;

- не може бути послідовно розміщений в основній частині звіту через великий обсяг або способи відтворення;
- може бути вилучений для широкого кола читачів, але є необхідним для фахівців даної галузі.

У додатки можуть бути включені:

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

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

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

Сторінки додатків нумеруються, продовжуючи загальну нумерацію у пояснювальній записці.

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

3.6 Оформлення ілюстративної частини

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

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

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

4 ГРАФІК ВИКОНАННЯ КУРСОВОЇ РОБОТИ І ПОРЯДОК ЗАХИСТУ

4.1 Графік виконання

Рекомендується такий графік виконання курсової роботи, який враховує самостійну роботу студентів під час 7-го триместру (16 тижнів) для денної форми навчання та 6-го семестру для заочної форми навчання.

Зміст розділу	Термін виконання
Отримання завдання на курсова робота, розробка і оформлення індивідуального завдання	1-2 тижд.
Аналіз літературних джерел та сучасного стану проблеми, обґрунтування вибору методів захисту	2-3 тижд.
Розробка структури програмного забезпечення: усвідомлення алгоритмів підзадач, виконання контрольних прикладів, розробка інтерфейсу, обґрунтування необхідності додаткових засобів, розробка структури вхідних і вихідних даних, підбір необхідних програмних засобів і т. д.	3-5 тижд.
Розробка програмного забезпечення і налагодження його: програмування та тестування основних процедур та функцій, програмна реалізація інтерфейсу, програмна реалізація роботи з файлами, з елементами керування, реалізація захисту та перевірки цілісності даних і т. д.	6-11 тижд.
Тестування розробки та виправлення виявлених недоліків. Підготовка контрольних прикладів, перевірка ефективності захисту	11 тижд.
Оформлення пояснювальної записки до курсової роботи, розробка рекомендацій щодо експлуатації розробленої програми	12-13 тижд.
Здача курсової роботи на попередню перевірку: демонстрація роботи програми та чернетки пояснювальної записки	13 тижд.
Корегування і доповнення (при необхідності) програми згідно із зауваженнями керівника курсової роботи, врахування і виправлення пояснювальної записки	14-15 тижд.
Захист курсової роботи	15-16 тижні

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

4.2 Оцінювання виконання курсової роботи

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

Курсова робота оцінюється за 100-бальною шкалою:

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

Розподіл бальної оцінки за виконання курсової роботи:

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

Пояснювальна записка (ПЗ) оцінюється за такими параметрами:

- дотримання вимог щодо оформлення ПЗ згідно ДСТУ 3008-95;
- відповідність ПЗ і програми індивідуальному завданню;
- відповідність ПЗ програмному забезпеченню;
- логічна пов'язаність розділів пояснювальної записки.

Програмна частина оцінюється за такими параметрами:

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

Захист роботи оцінюється за такими параметрами:

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

ПЕРЕЛІК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Дудатьєв А.В., Каплун В.А., Семеренко В.П. Захист програмного забезпечення. Навчальний посібник. Частина 1. – Вінниця: ВНТУ, 2005. – 140 с.
2. Каплун В.А., Баришев Ю. В., Дмитришин О. В.. Захист програмного забезпечення. Навчальний посібник. Частина 2. – Вінниця: ВНТУ, 2013. – 151 с.
3. Каплун В. А., Дмитришин О. В., Баришев Ю. В. Захист програмного забезпечення : лабораторний практикум – Вінниця : ВНТУ, 2016. – 75 с.
4. Казарин О. В. Безопасность программного обеспечения компьютерных систем. Монография. – М.: МГУЛ, 2003. – 212 с.
5. Казарин О. В. Теория и практика защиты программ. – М.: МГУЛ, 2004. – 450 с.
6. Соколов А., Степанюк О. Защита от компьютерного терроризма. Справочное пособие. – БХВ-Петербург: Арлит, 2002. – 496 с.
7. Лукічов, В. В. Методи та засоби стеганографічного захисту інформації на основі вейвлет-перетворень : монографія. – Вінниця: ВНТУ, 2014. – 160 с.
8. Щеглов А. Ю. Защита компьютерной информации от несанкционированного доступа. – Санкт-Петербург: Наука и техника, 2004. – 384 с.
9. Румянцев П. В. Исследование программ Win32: до дизассемблера и отладчика – М.: Горячая линия-Телеком, 2004. – 367 с.
10. Румянцев П. В. Работа с файлами в Win 32 API. – М.: Горячая линия-Телеком, 2002. – 216 с.
11. Чернов А. В. Интегрированная среда для исследования "обфускации" программ. Доклад на конференции, посвящённой 90-летию со дня рождения А. А. Ляпунова. Россия, Новосибирск, 8-11 октября 2001 года. <http://www.ict.nsc.ru/ws/Lyap2001>.
12. Касперски К. Техника и философия хакерских атак. – М.: Солон-Р, 1999. – 272 с.
13. Касперски К. Компьютерные вирусы изнутри и снаружи. – СПб.: Питер, 2006. – 527 с.
14. Дубовой В. М., Москвіна С. М., Никитенко О. Д. Моделювання процесів і систем керування з використанням UML
15. Матеріали з мережі INTERNET.

Додаток А

Варіанти завдань на курсову роботу

1. Розробка захисту інформації користувача за допомогою використання flash-носіїв.
2. Розробка захисту програмного забезпечення за допомогою використання системного реєстру.
3. Розробка елементів захисту web-сторінок.
4. Розробка елементів захисту від шкідливого програмного забезпечення (вірусів).
5. Розробка захисту програмного забезпечення від клавіатурних шпигунів (шляхом встановлення хуків).
6. Розробка захисту програмного забезпечення команд роботи з вікнами (шляхом встановлення хуків).
7. Розробка програми для ідентифікації користувача за клавіатурним почерком на основі статистичних методів обробки інформації.
8. Розробка програми для ідентифікації користувача за особливостями роботи з маніпулятором миші на основі статистичних методів обробки інформації.
9. Розробка програми для ідентифікації користувача за особливостями роботи з маніпулятором миші на основі статистичних методів обробки інформації.
10. Розробка захисту програм від несанкціонованого копіювання за допомогою нестандартного форматування дистрибутивних носіїв.
11. Розробка засобу захисту програмного забезпечення від несанкціонованого копіювання шляхом прив'язки до архітектури ПЕОМ.
12. Розробка захисту програмного забезпечення від несанкціонованого копіювання шляхом прив'язки до файлової системи.
13. Розробка захисту програмного забезпечення від несанкціонованого використання шляхом прив'язки до вінчестера.
14. Розробка захисту програмного забезпечення від несанкціонованого копіювання шляхом прив'язки до BIOS.
15. Розробка захисту програмного забезпечення від несанкціонованого копіювання шляхом прив'язки до CMOS.
16. Розробка захисту програмного забезпечення від несанкціонованого використання за допомогою опитування довідників.
17. Розробка захисту програмного забезпечення від статичного дослідження шляхом внесення недосяжного коду.
18. Розробка захисту програмного забезпечення від статичного дослідження шляхом внесення "мертвого" коду.
19. Розробка захисту програмного забезпечення від статичного дослідження шляхом внесення надлишкового коду.
20. Розробка захисту програмного забезпечення від статичного дослідження шляхом переплутування і клонування функцій.
21. Розробка лабораторної роботи: "Дослідження роботи програм-обфускаторів".
22. Розробка захисту програмного забезпечення від статичного дослідження шляхом лексичної обфускації.
23. Розробка лабораторної роботи: "Дослідження роботи програм-декомпіляторів з різних мов програмування".
24. Розробка лабораторної роботи: "Дослідження роботи програм-дизасемблерів".

25. Розробка лабораторної роботи: "Дослідження роботи програм-скремблерів".
26. Розробка лабораторної роботи: "Дослідження роботи програм для редагування ресурсів".
27. Розробка захисту програмного забезпечення від статичного дослідження шляхом методу перемішування коду (скремблювання).
28. Розробка захисту програмного забезпечення шляхом обмеження на його використання (за кількістю запусків, за кількістю копій, за лімітом часу використання і т.д.).
29. Розробка лабораторної роботи: "Динамічне дослідження роботи захищених програмних продуктів за допомогою налагоджувача SoftIce".
30. Розробка лабораторної роботи: "Динамічне дослідження роботи захищених програмних продуктів за допомогою відомих налагоджувачів" (окрім SoftIce).
31. Розробка елементів захисту програмного забезпечення від налагоджувачів реального режиму.
32. Розробка елементів захисту програмного забезпечення від налагоджувачів захищеного режиму.
33. Розробка навісного захисту виконуваних файлів програмного забезпечення шляхом модифікації заголовку виконуваних файлів.
34. Розробка захисту програмного забезпечення шляхом модифікації структури виконуваних файлів.
35. Розробка лабораторної роботи: "Дослідження вмісту, структури і формату виконуваних файлів в операційних системах сімейства Windows".
36. Розробка лабораторної роботи: "Дослідження можливостей зняття дампу пам'яті".
37. Системи захисту програмного забезпечення в операційних системах сімейства Unix (Linux) та аналіз засобів зламу цих захистів.
38. Розробка програми дослідження дескрипторів безпеки і списків контролю доступу (ACL) та керування ними.
39. Розробка програми дослідження стану процесів і потоків та керування ними.
40. Розробка лабораторної роботи: "Захист програмного забезпечення за допомогою програм-пакувальників".
41. Розробити програму для приховування інформації у графічних файлах.
42. Розробити програму для приховування інформації у музичних файлах.
43. Розробити програму для приховування інформації у мультимедійних файлах.
44. Розробка програми для блокування роботи комп'ютера.
45. Розробка захисту програмного забезпечення за допомогою електронних ключів.

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

Додаток Б

Приклад оформлення титульного аркуша

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

КУРСОВА РОБОТА

з дисципліни "Захист програмного забезпечення"
на тему:

**«ПРОГРАМНИЙ ЗАСІБ ДЛЯ ЗАХИСТУ ВИКОНУВАНИХ
ФАЙЛІВ»**

08-20.ЗПЗ.118.13.107 ПЗ

Студента (ки) 2 курсу групи ІБС-15 б
спеціальності 125 – Кібербезпека

_____ Петрова П. П.

Керівник: ст. викл. кафедри ЗІ

_____ Каплун В. А.

Національна шкала _____

Кількість балів: ___ Оцінка ECTS ___

Члени комісії : _____ (підпис) _____ (прізвище та ініціали)

_____ (підпис) _____ (прізвище та ініціали)

м. Вінниця – 2018 рік

Додаток В

Приклад оформлення індивідуального завдання

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії

ЗАТВЕРДЖУЮ

Зав. кафедри ЗІ, д. т. н., проф.

_____ В. А. Лужецький

” ____ ” _____ 2018

р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на курсову роботу з дисципліни ”Захист програмного забезпечення”
студенту групи 1 БС-15 б Холодову І. В.

Тема: «Розробка програми захисту від несанкціонованого копіювання шляхом прив'язки до конфігурації комп'ютерної системи»

Вхідні дані: принцип захисту – від копіювання; механізм захисту – навісний; метод захисту – прив'язка до комп'ютера; параметри прив'язки – конфігурація дискової системи; об'єкт захисту – виконуваний модуль програми; середовище функціонування – ОС Windows.

Вихідні дані: програмний засіб для захисту, інструкції для застосування.

Короткий зміст частин проекту

- Текстова** (пояснювальна записка): Вступ. Аналіз літературних джерел і обґрунтування методу захисту. Розробка системи. Тестування програмного засобу. Висновки. Список використаних джерел. Додатки.
- Ілюстративна:** Схема функціонування програми. Схема алгоритму здійснення прив'язки до ПК. Схема ресурсів системи захисту. Візуальне подання результатів програми.

Дата видачі _____ 2018 р.

Керівник _____ Каплун В. А.

Додаток Г

Використання схем алгоритмів, програм, даних і систем

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

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

Символи поділяються на:

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

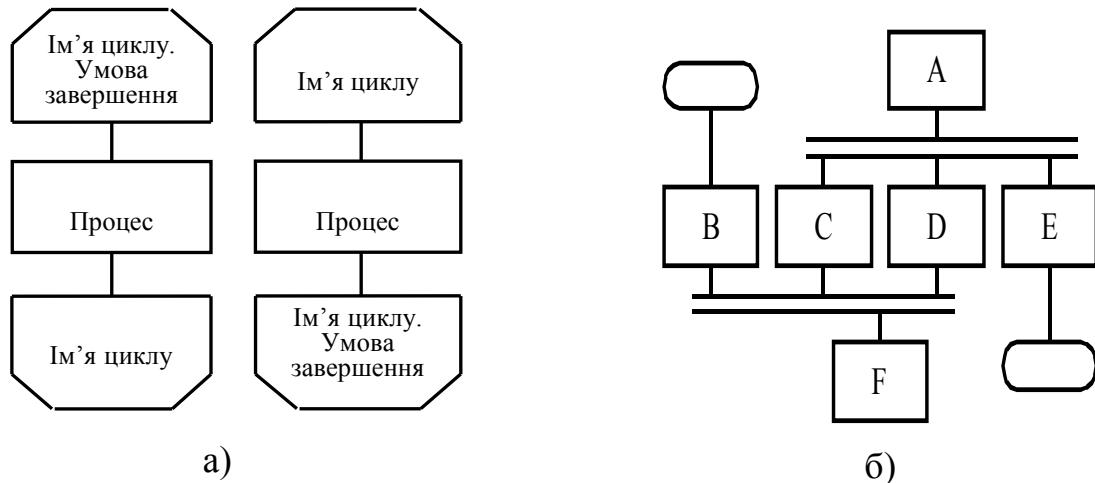


Рисунок Г.1 – Приклад застосування символів процесу (а – використання символів меж циклу; б – використання символів паралельних дій)

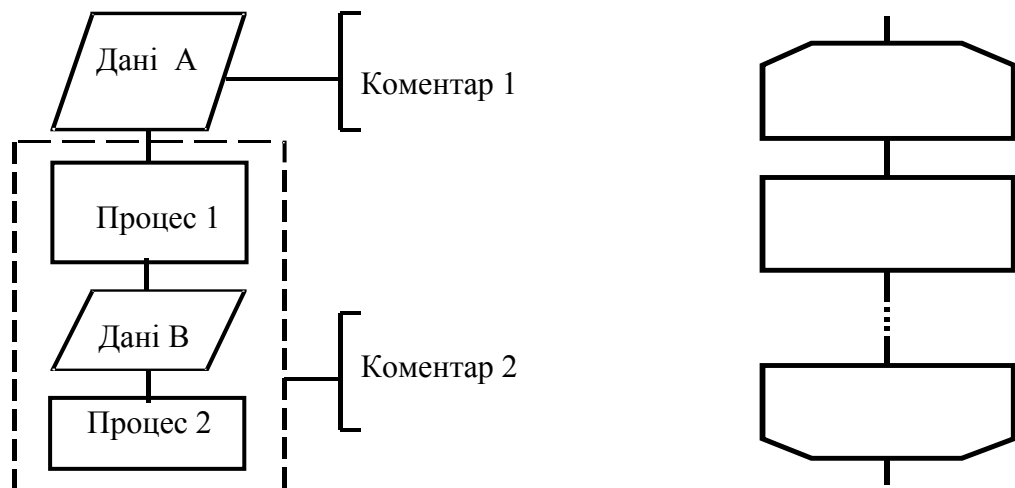


Рисунок Г.2 – Приклади використання спеціальних символів

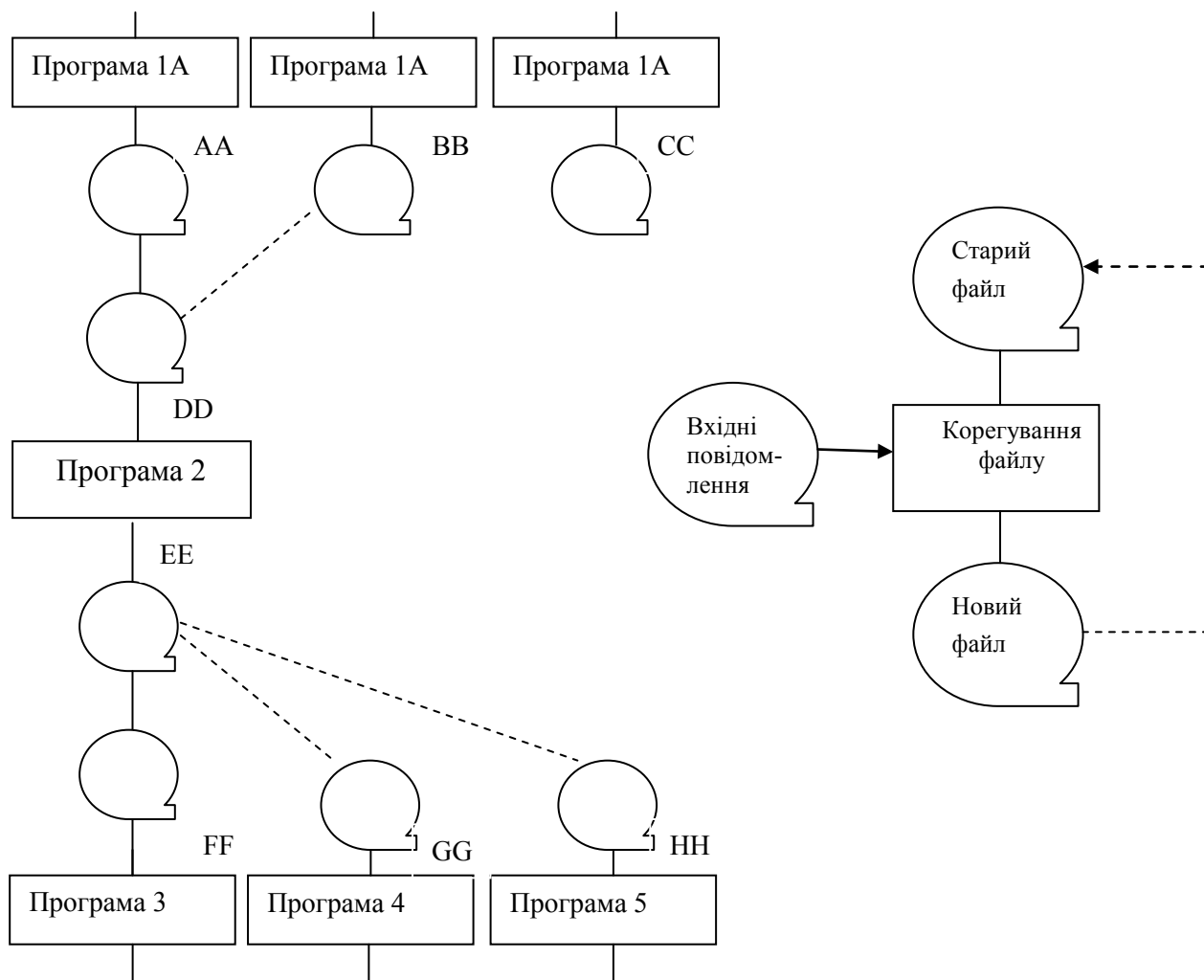


Рисунок Г.3 – Приклади використання символів ліній

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

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

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


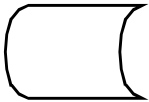
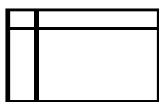

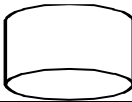


Схеми роботи системи відображають управління операціями і потік даних в системі. Схема роботи системи складається з: символів даних, що вказують на наявність даних (символи даних можуть також вказувати на вид носія даних); символів процесу, що вказують операції, які слід виконати над даними, а також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

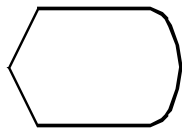
даних між процесами і (або) носіями даних, а також потік управління між процесами; спеціальних символів, використовуваних для полегшення написання і читання блок-схеми.

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





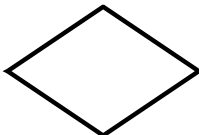
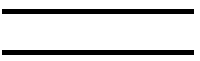
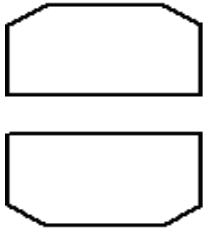
Схема ресурсів системи відображає конфігурацію блоків даних, блоків обробки цих даних, яка потрібна для розв'язання задачі або набору задач. Схема ресурсів системи складається з: символів даних, що відображають вхідні, вихідні і запам'ятовують пристрої обчислювальної машини; символів процесу, що відображають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

Таблиця Г.1 – Символи даних

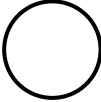

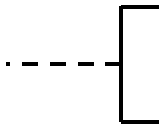

<i>Основні символи даних</i>		
Дані		Символ відображає дані, носій даних невизначений
Дані, що запам'ятовуються		Символ відображає дані, що зберігаються, у вигляді, придатному для обробки, носій даних невизначений
<i>Специфічні символи даних</i>		
Оперативний запам'ятовувальний пристрій		Символ відображає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

Дисплей		Символ відображає дані, подані у візуальній людиночитабельній формі на носії у вигляді пристрою відображення (екран для візуального спостереження, індикатори введення інформації)
---------	---	--


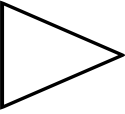


Таблиця Г. 2 – Символи процесу

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

Таблиця Г. 3 – Спеціальні символи

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

Таблиця Г. 4 – Символи ліній

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

Додаток Д

Приклади UML-діаграм

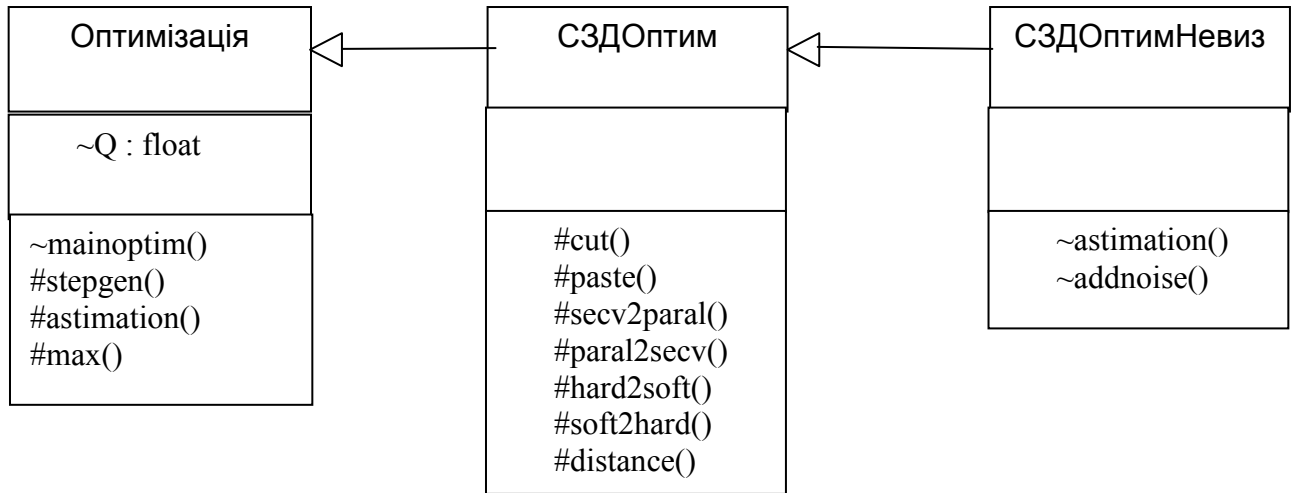


Рисунок Д.1 – Приклад діаграми класів

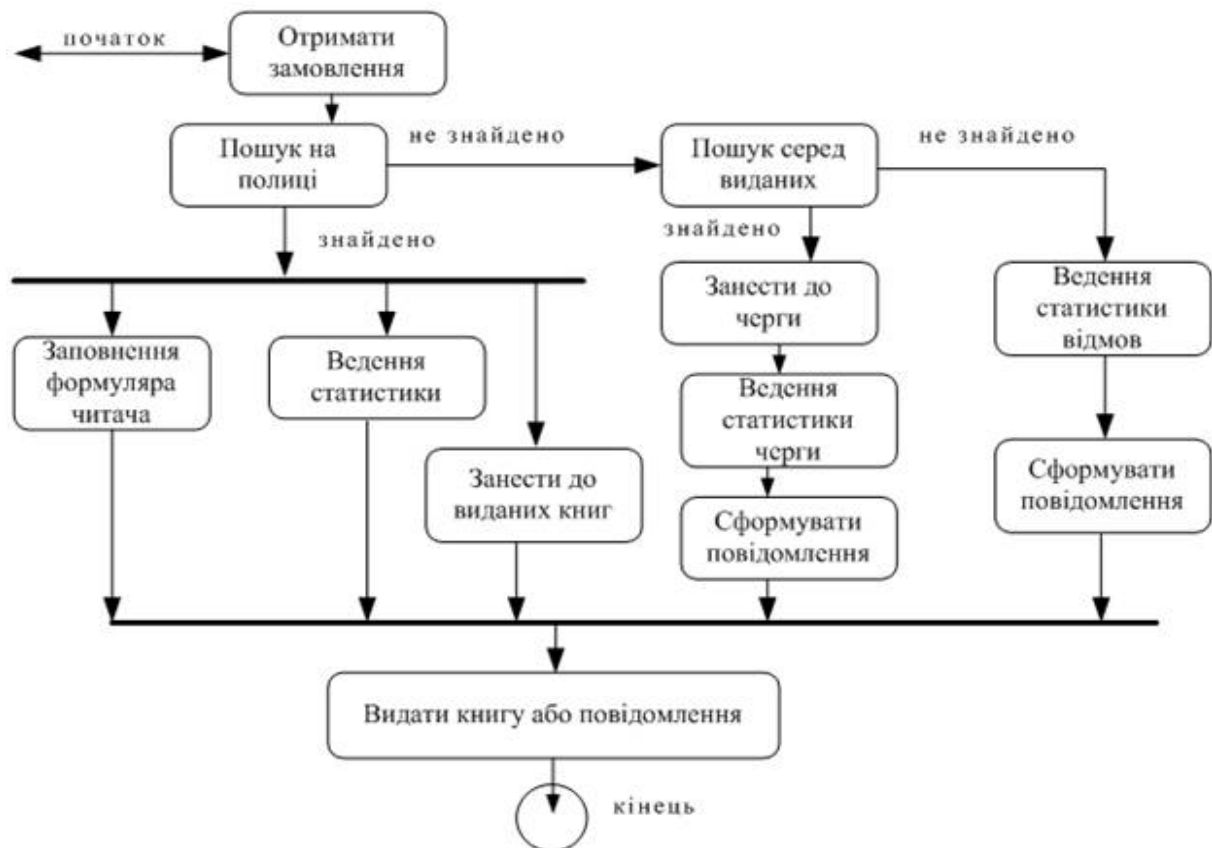


Рисунок Д.2 – Приклад діаграми діяльності

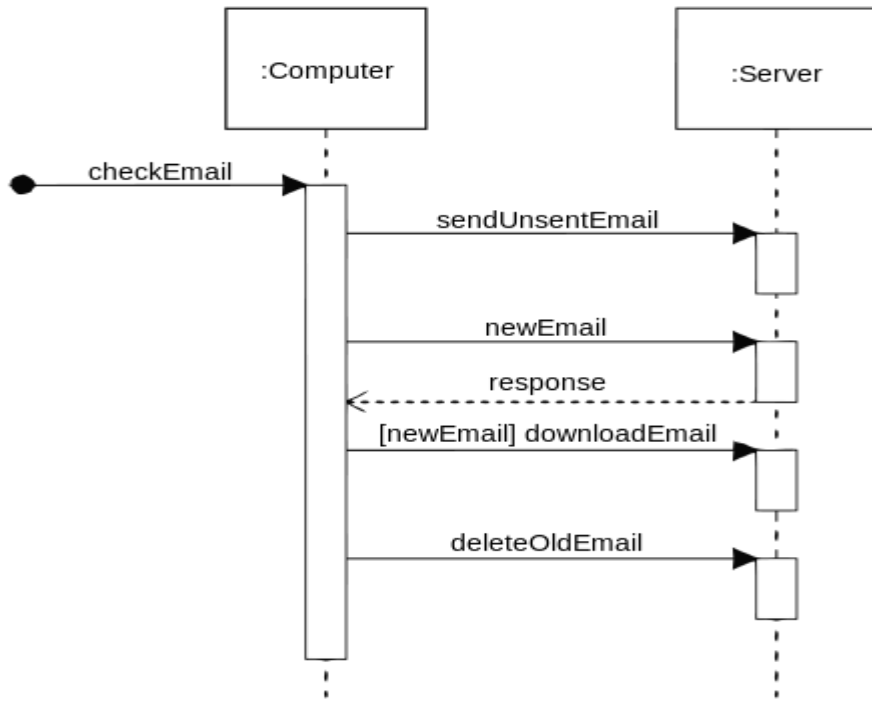


Рисунок Д.3 – Приклад діаграми послідовності

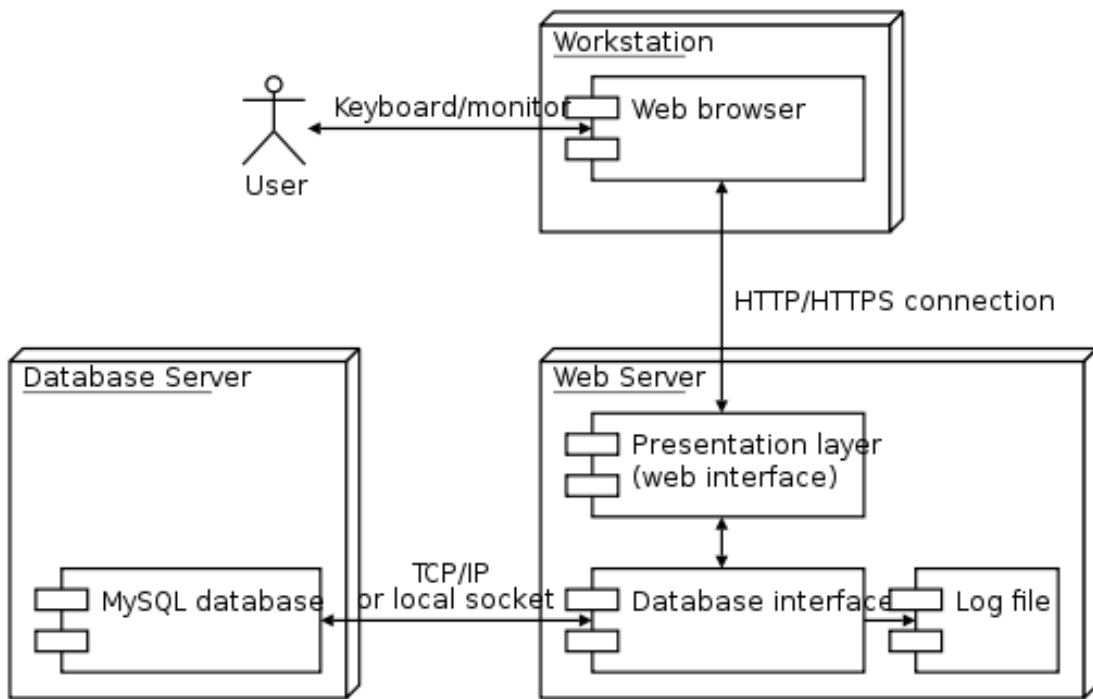


Рисунок Д.4 – Приклад діаграми розгортання