

МЕТОДИЧНІ ВКАЗІВКИ

до виконання курсового проекту з дисципліни

"ЗАХИСТ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ"

для студентів напряму підготовки 6.170101

“Безпека інформаційних і комунікаційних систем ”

ЗМІСТ

| | | |
|-----|--|----|
| 1 | ОСНОВНІ ВИМОГИ ДО КУРСОВОГО ПРОЕКТУ | 2 |
| 1.1 | Мета і задачі курсового проекту..... | 2 |
| 1.2 | Тематика курсового проекту | 2 |
| 2 | ВИМОГИ ДО РОЗРОБКИ ПРОГРАМНОГО ПРОДУКТУ | 4 |
| 3 | ОФОРМЛЕННЯ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ | 5 |
| 3.1 | Загальні правила оформлення..... | 5 |
| 3.2 | Загальна структура пояснювальної записки | 8 |
| 3.3 | Вступна частина пояснювальної записки | 8 |
| 3.4 | Вміст і оформлення основної частини | 10 |
| 3.5 | Оформлення додатків..... | 14 |
| 3.6 | Графічна частина..... | 16 |
| 4 | ГРАФІК ВИКОНАННЯ КУРСОВОГО ПРОЕКТУ І ПОРЯДОК ЙОГО ЗАХИСТУ | 20 |
| | ПЕРЕЛІК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ..... | 21 |
| | Додаток А. Варіанти завдань на курсову роботу | 22 |
| | Додаток Б. Приклад оформлення титульного аркуша | 24 |
| | Додаток В. Приклад оформлення першого аркуша змісту | 25 |
| | Додаток Г. Приклад оформлення тексту пояснювальної записки..... | 26 |
| | Додаток Д. Приклад оформлення індивідуального завдання | 27 |
| | Додаток Е. Приклад оформлення анотації..... | 28 |
| | Додаток Ж. Зразок оформлення технічного завдання..... | 29 |
| | Додаток И. Символи даних, процесів і ліній | 32 |
| | Додаток К. Приклад схеми даних програми..... | 35 |
| | Додаток Л. Приклад схеми роботи програми | 36 |
| | Додаток М. Приклад схеми роботи системи | 37 |
| | Додаток Н. Приклад схеми взаємодії програм | 38 |
| | Додаток П. Приклад схеми ресурсів програми | 39 |
| | Додаток Р. Приклад оформлення відомості курсового проект..... | 40 |

1 ОСНОВНІ ВИМОГИ ДО КУРСОВОГО ПРОЕКТУ

1.1 Мета і задачі курсового проекту

Курсовий проект (КП) – навчальний проект з дисципліни, який містить елементи ескізного і технічного проектів та робочої документації.

В курсовому проекті 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 курсового проекту студенти повинні використати всі знання, отримані ними під час вивчення дисциплін “Програмування”, “Системне програмування”, “Операційні системи”, “Захист операційних систем”, “Основи інформаційної безпеки”, “Криптографія і криптоаналіз”.

Під час виконання курсового проекту 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.

1.2 Тематика курсового проекту

Зміст курсового проекту відповідає навчальній програмі та робочому плану дисципліни “Захист програмного забезпечення” і повинен відображати суть обраної студентом теми. Зміст КП 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 курсового проекту 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 суттєво впливатимуть на загальну оцінку з дисципліни за модуль.

Тематика курсового проекту повинна бути пов'язана з майбутньою спеціальністю студентів. Для програмної реалізації даного курсового проекту пропонується ряд основних методів захисту програм від копіювання, від статичного та динамічного дослідження.

Крім того, як об'єкт для курсового проекту 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.

Індивідуальне завдання для курсових робіт визначається викладачем із загального списку завдань на курсову роботу (Додаток А). 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.

2 ВИМОГИ ДО РОЗРОБКИ ПРОГРАМНОГО ПРОДУКТУ

Результатом виконання курсового проекту повинен бути, як правило, повноцінний програмний додаток для конкретної операційної системи (сімейства Windows або Unix), призначений для здійснення захисту деякого об'єкта. Крім того, курсовий проект 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 продукт, виконаний за допомогою певних програмних засобів (мова HTML, пакетний файл тощо) при демонстрації додаткового матеріалу для поглибленого вивчення певних тем.

Розроблена програма повинна відповідати таким вимогам.

1. *Розробка 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 порівнянний з іншими методами захисту, наведено його переваги та недоліки (цьому повинен бути присвячений перший розділ пояснювальної записки).
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. *Графічна частина* розробки повинна подаватися у тексті пояснювальної записки до курсового проекту у вигляді відповідних рисунків, схем функціонування системи захисту, схем взаємодії програм, блок-схем конкретних алгоритмів, схем даних тощо. При цьому всі схеми, рисунки, блок-схеми повинні бути оформлені відповідно до прийнятих правил оформлення технічної документації.
4. *При програмній реалізації задачі* допускається застосування будь-яких мов програмування (Pascal, C++, C#, Assembler, Delphi тощо) та будь-якого візуального середовища програмування, причому вибір тієї або іншої мови програмування і середовища програмування повинні бути також обґрунтовані.
5. *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.
6. 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. Для формування висновків щодо ефективності захисту можна використовувати засоби статичного та динамічного аналізу.
7. *Розробка інструктивних документів* для роботи з програмою є обов'язковою складовою пояснювальної записки до курсового проекту.

3 ОФОРМЛЕННЯ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

3.1 Загальні правила оформлення

Пояснювальна записка відноситься до текстових документів, які містять інформацію, подану в основному технічною мовою та графічну інформацію у вигляді ілюстрацій. Пояснювальна записка виконується згідно з вимогами міждержавного стандарту ГОСТ 2.105-95. Текст пояснювальної записки повинен бути набраний на комп'ютері та роздрукований на стандартних аркушах паперу формату А4 (210×297 мм) з однієї сторони.

Рамки. Кожен аркуш пояснювальної записки до курсового проекту (крім анотації та індивідуального завдання) повинен мати стандартну рамку робочого поля і основний напис. Титульний аркуш повинен мати рамку, наведену у додатку Б, перша сторінка розділу ЗМІСТ повинна бути оформлена на аркуші з рамкою, наведеною у додатку В, наступна та всі інші сторінки повинні бути оформлені на аркушах з рамкою, наведеною у додатку Г. У кожному рамок (окрім рамки титульного аркуша) обов'язково повинні бути вписані номер сторінки та шифровий код проекту. Для формування рамок при виконанні пояснювальної записки у текстовому редакторі Microsoft Office Word рекомендується користуватись можливостями оформлення колонтитулів.

Шрифт і відступи. Текст пояснювальної записки повинен бути набраний у будь-якому текстовому редакторі (наприклад, Microsoft Office Word) шрифтом Times New Roman розміром 14. Текст записки слід друкувати через півтора інтервали. Текст розміщують таким чином, щоб відстань від рамки до робочого поля становила: зліва і справа – 3-5 мм; зверху і знизу – не менше 10 мм; абзац – 5 знаків. Відступи рамок до країв аркуша: зліва – 20-25 мм, решта – 5 мм. Наявність рамок у додатках, які містять графічний матеріал, є обов'язковою.

Нумерація сторінок. Сторінки повинні бути пронумеровані, починаючи з третьої (перша сторінка розділу ЗМІСТ). Сторінки 1-2 (які містять титульний аркуш, індивідуальне завдання, анотація) не нумеруються. Сторінки пояснювальної записки слід нумерувати арабськими цифрами, додержуючись наскрізної нумерації впродовж усього тексту звіту. Номер сторінки проставляють у відповідному місці рамки. Нумерація додатків продовжує основну нумерацію.

Оформлення розділів і підрозділів. Структурними елементами основної частини ПЗ є розділи, підрозділи, пункти, підпункти, переліки.

Розділ – головний ступінь поділу тексту, позначений номером і має заголовок. *Підрозділ* – частина розділу, позначена номером і має заголовок. *Пункт* – частина розділу чи підрозділу, позначена номером і може мати заголовок. *Підпункт* – частина пункту, позначена номером і може мати заголовок. Заголовки структурних елементів необхідно нумерувати тільки арабськими числами.

Кожен розділ рекомендується починати з нової сторінки.

Заголовки розділів, що містять анотацію, зміст, список літератури, виконують посередині рядка великими літерами. Заголовки усіх розділів, підрозділів, пунктів та підпунктів записують з абзацу малими літерами, починаючи з великої.

Після заголовків першого рівня пропускають один рядок. Перед заголовками

другого рівня і після них пропускають один рядок.

Заголовки розділів і підрозділів, пунктів і підпунктів не повинні містити знаків переносу на новий рядок. Назви розділів і підрозділів, пунктів і підпунктів не повинні мати крапки в кінці.

Розділи нумерують порядковими номерами в межах всього документа (1, 2, і т.д.). Підрозділи нумерують в межах кожного розділу, пункти – в межах підрозділу і т.д. (3.1, 3.2, 3.2.1, 3.2.2, 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.

Допускається розміщувати текст між заголовками розділу і підрозділу, між заголовками підрозділу і пункту. Посилання в тексті на розділи виконується за формою: "...наведено в розділі 3".

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 – 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 чи випадковим чином |

На всі таблиці мають бути посилання за формою "... в табл. 1 або в дужках по тексту (табл. 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”, або в дужках по тексту (рис. 3.6). Посилання на раніше наведений рисунок дають зі скороченим словом ”дивись” (див. рис. 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 12 – 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 ГОСТ 1494-77. 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 Загальна структура пояснювальної записки

Кожен етап курсового проектування 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.

Робота, яка подається у вигляді копії, до захисту не приймається.

Пояснювальна записка до курсового проекту повинна мати таку структуру.

1. *Вступну частину*, яка містить:

- титульний аркуш;
- індивідуальне завдання;
- анотацію;
- зміст.

2. *Основна частина*, яка складається із:

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

3. *Додатки*, які розміщуються після основної частини пояснювальної записки курсової роботи.

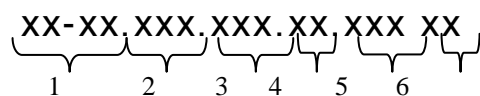
4. *Графічна частина*, в якій розмішено графічний матеріал, що виноситься на захист.

3.3 Вступна частина пояснювальної записки

Титульний аркуш

Титульний аркуш є першою сторінкою курсового проекту, яка не нумерується. Згідно з діючим стандартом ГОСТ 2.105-95 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, ТЗ – технічне завдання, А1 – схема роботи системи вцілому, А2 – схема роботи програми (блок-схема алгоритму), А3 – схема ресурсів, А4 – схема взаємодії програм (модулів); А5 – схема даних, А0 – схема об'єднана, ВКП – відомість курсового проекту.

Індивідуальне завдання

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

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

Керівник проекту пропонує перелік основних етапів реалізації задачі курсового проекту, 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 курсового проекту 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.

Розробка повного технічного завдання до курсового проекту повинна подаватись у першому з додатків, зміст якого буде вказаний в розділі "Додатки".

Анотація

Анотація призначена для ознайомлення з текстовим документом курсового проекту. 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 (третьої), нумерація якої не зазначається. Заголовок (слово АНОТАЦІЯ) розміщується по центру сторінки, після нього пропускається 1 рядок (додаток Е).

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) основної частини пояснювальної записки; 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.

Перша сторінка змісту оформляється на аркуші з великою рамкою (додаток В), решта сторінок – на аркуші з рамкою іншої форми (додаток Г). Номер сторінки на першій сторінці змісту проставляється у відповідній графі рамки. Сам зміст за нумерацією пояснювальної записки є, як правило, третьою сторінкою.

Нумерація у змісті починається зі вступу (відповідно до нумерації у пояснювальній записці). Нумерація сторінок по всій пояснювальній записці, включаючи додатки, повинна бути наскрізною.

Назви заголовків змісту (усіх розділів, підрозділів, пунктів, підпунктів) повинні однозначно відповідати назвам заголовків пояснювальної записки за текстом і вирівнюватись по лівому краю аркуша.

Для запобігання плутанині з назвами розділів, підрозділів і нумерацією сторінок, а також для полегшення редагування і внесення змін в основний текст пояснювальної записки рекомендується використовувати можливості автоматичного формування змісту у документах Microsoft Office Word. Приклад оформлення змісту можна бачити у додатку В.

3.4 Вміст і оформлення основної частини

Вступ

Вступ пишуть з нової пронумерованої сторінки із заголовком „ВСТУП” посередині великими літерами на аркуші з рамкою (додаток Г).

Текст вступу повинен бути коротким і висвітлювати питання актуальності, значення, сучасний рівень і призначення курсового проекту. У вступі і далі за текстом не дозволяється використовувати скорочені слова, терміни, крім загальноприйнятих. Якщо ж в тексті є необхідність використовувати певні загальноприйняті скорочення (абревіатури), 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.

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!).

Реалізація задачі курсового проекту

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

У випадку, якщо курсовий проект 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 та 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 заголовочні 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 (18-20 сторінок).

Тестування програми і розробка інструкцій щодо роботи з нею

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 курсового проекту 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 (відповідно до ГОСТ 19.508-79):

- *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 відповідати ГОСТ 19.503-79 і 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 (відповідно до ГОСТ 19.504-79):

- *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 корегувань);
- *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 підпункти (відповідно до ГОСТ 19.505-79):

- *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.).

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 курсового проекту. 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 ПЗ.

Перелік літератури

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:

Посилання на книги:

1. Мамаев М., Технология защиты информации в интернете : [Специальный справочник] / Максим Мамаев, Сергей Петренко. - СПб. : Питер, 2002. – 848 с. – ISBN 5-318-00244-7.
2. Лужецький В. А. Інформаційна безпека : навчальний посібник / В. А. Лужецький, О. П. Войтович, А. В. Дудатьєв – Вінниця : УНІВЕРСУМ-Вінниця, 2009. – 240 с. – ISBN 978-966-641-265-5

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.

Посилання на ГОСТ і ДСТУ:

5. Единая система конструкторской документации.. Общие требования к текстовым документам : ГОСТ 2.105-95. – [Чинний від 1996—01—07]. –

Мінськ: Межгосударственный совет по стандартизации, метрологии и сертификации, 1996. – 29 с. – (Міждержавний стандарт).

6. 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).

Посилання на патенти:

7. 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.

Посилання на web-сторінки:

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

Посилання на матеріали конференції:

10. Каплун В. А. Використання ущільнення числових послідовностей для захисту програмного забезпечення / В. А. Каплун, Т. М. Алексеева // Методи та засоби кодування, захисту ущільнення інформації : міжнародна науково-практична конференція 20-22 квітня 2011 р. : тези доповідей. - Вінниця: ВНТУ, 2011. – С.121-122.

3.5 Оформлення додатків

Першим аркушем додатків для курсових проектів має бути технічне завдання, в якому вказуються: найменування та область застосування розробки; основа для розробки; мета і призначення; джерела розробки; технічні вимоги (показники призначення, показники надійності, вимоги безпеки тощо); стадії та етапи розробки; порядок контролю та приймання.

Технічне завдання повинно бути першим додатком до пояснювальної записки. У верхній частині посередині поля розміщується напис "Додаток А". Зразок оформлення технічного завдання показаний у додатку Ж.

Крім технічного завдання, 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, Ч, Ь, 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.

Приклад оформлення додатків можна переглянути у додатках до даних методичних вказівок.

3.6 Графічна частина

Курсовий проект повинен супроводжуватись графічним матеріалом, який буде доповнювати і деталізувати опис розробки задачі проекту. Кількість аркушів графічного матеріалу повинна бути не менше 3. Це можуть бути схеми функціонування програми, блок-схеми алгоритмів, схеми ресурсів системи захисту, схеми даних, схеми взаємодії програм тощо.

Аркуші, які містять графічний матеріал, повинні бути виконані на аркушах з рамками (додатки К, Л, М, Н, П).

Окремим і останнім додатком аркушем повинна бути "Відомість курсового проекту" – документ, який містить перелік документів, що входять в проект. Приклад оформлення відомості курсового проекту наведено у додатку Р.

Окремими підпунктами основного розділу або завершенням певних підрозділів бажано розробити і описати різні схеми:

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

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 (додаток И, табл. И. 1);
- символи процесів (додаток И, табл. И. 2). Приклад наведено на рис.1;
- спеціальні символи (додаток И, табл. И. 3). Приклад наведено на рис.2;
- символи ліній (додаток И, табл. И. 4). Приклад наведено на рис.3.

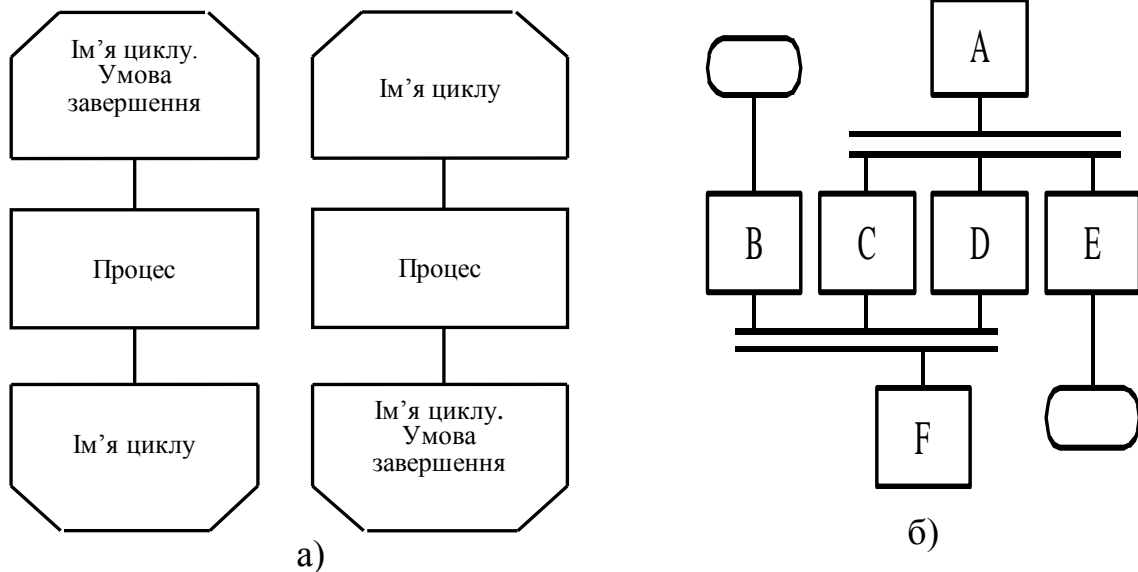


Рисунок 1 – Приклад застосування символів процесу: а) – використання символів меж циклу; б) – використання символів паралельних дій

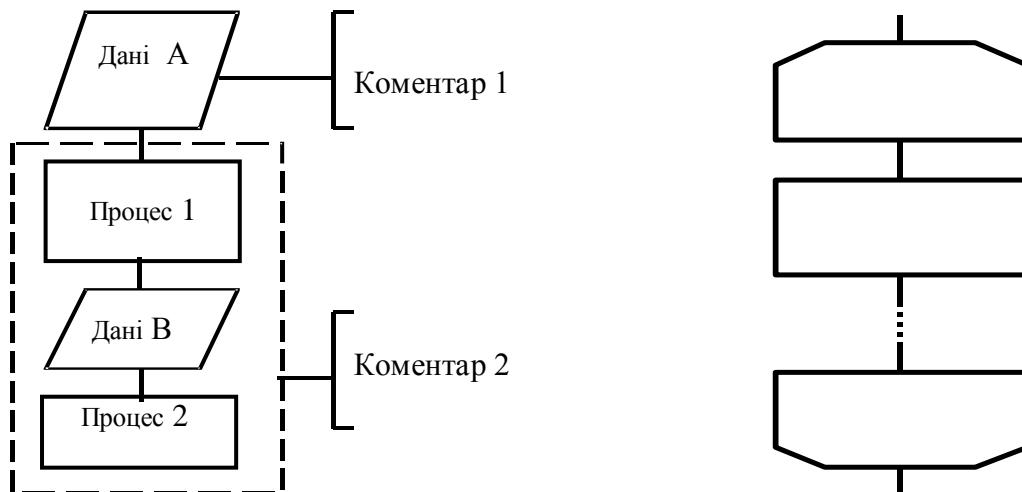


Рисунок 2 – Приклади використання спеціальних символів

Символи поділяються на:

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

Схеми даних відображають шлях даних при розв'язанні задач і визначають етапи обробки, а також різні використовувані носії даних. Схема даних складається із:

- 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.

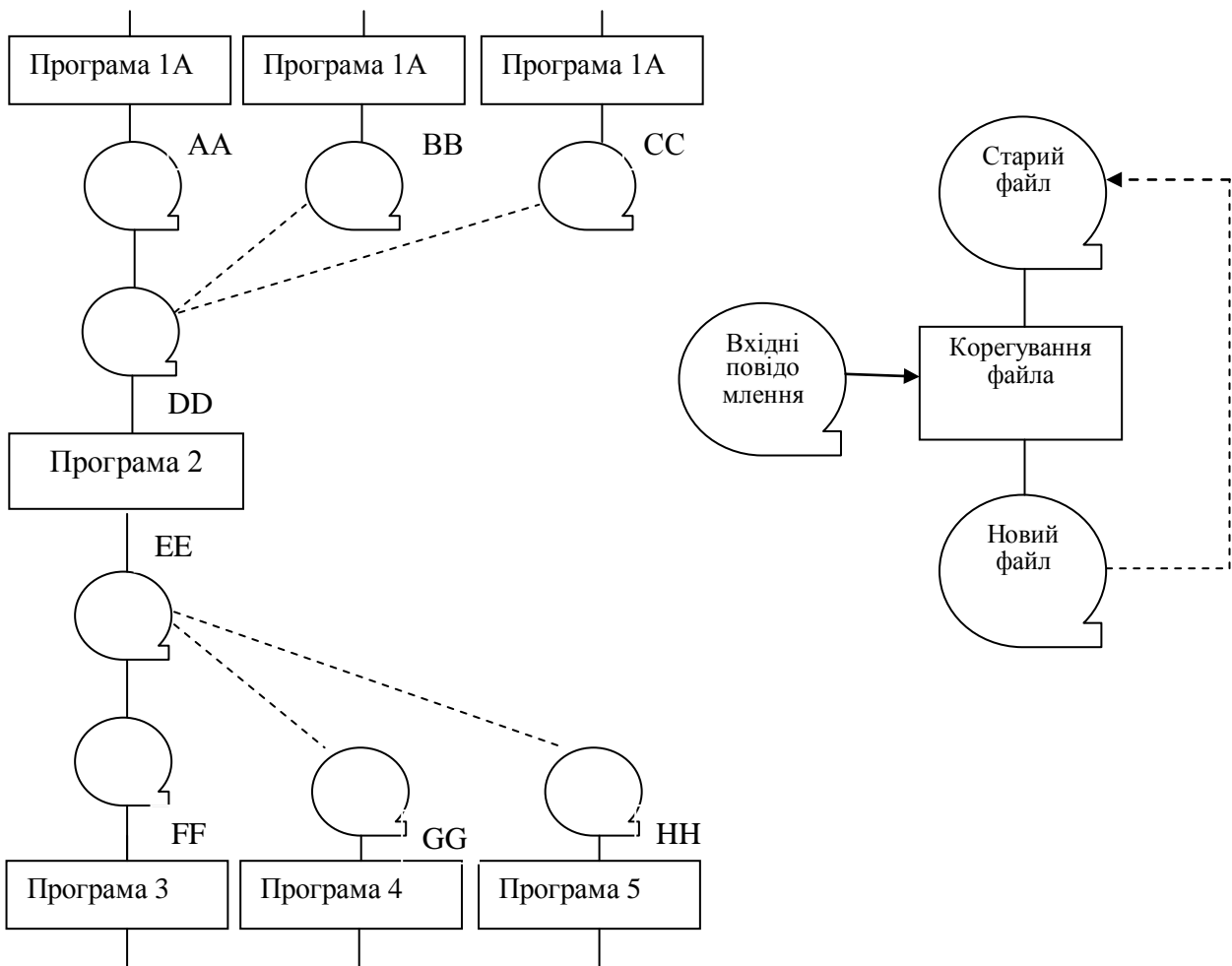


Рисунок 3 – Приклади використання символів ліній

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

Приклад оформлення схеми даних програми наведено у додатку К.

Схеми програм відображають послідовність операцій в програмі. Схема програми складається із:

- 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.

Приклад оформлення схеми програми наведено у додатку Л.

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 вид носія даних);
- символів процесу, що вказують операції, які слід виконати над даними, а також з тих, які визначають логічний шлях, якого слід дотримуватися;

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

Приклад оформлення схеми роботи системи наведено у додатку М.

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 зображатися більш, 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, використовуваних для полегшення написання і читання схеми.

Приклад оформлення схеми взаємодії програм наведено у додатку Н.

Схема ресурсів системи відображає конфігурацію блоків даних і оброблюваних блоків, яка потрібна для розв'язання задачі або набору задач. Схема ресурсів системи складається із:

- символів даних, що відображають вхідні, вихідні і запам'ятовувальні пристрої обчислювальної машини;
- символів процесу, що відображають процесори (центральні 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.

Приклад оформлення схеми ресурсів системи наведено у додатку П.

Розроблені схеми повинні супроводжувати і пояснювати процес розробки програми для реалізації задачі курсового проекту. Крім того, основні схеми повинні бути наведені у тексті пояснювальної записки і складати частину графічного матеріалу до курсового проекту.

4 ГРАФІК ВИКОНАННЯ КУРСОВОГО ПРОЕКТУ І ПОРЯДОК ЙОГО ЗАХИСТУ

Рекомендується такий графік виконання курсового проекту, 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 курсовий проект, розробка і оформлення індивідуального завдання | 1 тижд. |
| Аналіз літературних джерел та сучасного стану проблеми, обґрунтування вибору методів захисту | 2 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 курсового проекту, розробка рекомендацій щодо експлуатації розробленої програми | 12-13 тижд. |
| Здача курсового проекту на попередню перевірку: демонстрація роботи програми та чернетки пояснювальної записки | 13 тижд. |
| Корегування і доповнення (при необхідності) програми згідно із зауваженнями керівника курсового проекту, врахування і виправлення пояснювальної записки | 14-15 тижд. |
| Захист курсового проекту | 15-16 тижні |

Готовність до захисту курсового проекту 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 курсового проекту студент повинен захистити його перед комісією у складі двох викладачів, які призначені кафедрою.

ПЕРЕЛІК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Казарин О. В. Безопасность программного обеспечения компьютерных систем. Монография. – М.: МГУЛ, 2003. – 212 с.
2. Казарин О. В. Теория и практика защиты программ. – М.: МГУЛ, 2004. – 450 с.
3. Соколов А., Степанюк О. Защита от компьютерного терроризма. Справочное пособие. – БХВ-Петербург: Арлит, 2002. – 496 с.
4. Щеглов А. Ю. Защита компьютерной информации от несанкционированного доступа. – Санкт-Петербург: Наука и техника, 2004. – 384 с.
5. Середа С. А. Оценка эффективности систем защиты программного обеспечения. Материалы Международной конференции IPSIT'99, 1999.
6. Середа С. А. Программно-аппаратные системы защиты программного обеспечения. Материалы Международной конференции аспирантов при Экономической академии республики Молдова, 1999.
7. Спесивцев А. В., Вегнер В. А. и др. Защита информации в персональных ЭВМ – М.: Радио и связь, МП "Веста", 1993. – 193 с.
8. Правиков Д. И. Ключевые дискеты: разработка элементов защиты от несанкционированного копирования - М.: Радио и связь, 1995.
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. Дудатьев А. В., Каплун В. А., Семеренко В.П. Захист програмного забезпечення. Навчальний посібник. Частина 1. – Вінниця: ВНТУ, 2005. – 140 с.
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.

Додаток Б

Приклад оформлення титульного аркуша

Міністерство освіти і науки, молоді та спорту України
Вінницький національний технічний університет
Інститут інформаційних технологій та комп'ютерної інженерії
Кафедра захисту інформації

РОЗРОБКА СИСТЕМИ ЗАХИСТУ ПРОГРАМ ВІД ДИЗАСЕМБЛЮВАННЯ

Пояснювальна записка
до курсового проекту з дисципліни
”Захист програмного забезпечення”
за напрямом підготовки
6.170101 – “ Безпека інформаційних і комунікаційних систем ”
08-20.ЗПЗ.010.13.107 ПЗ

| | |
|-----------|--|
| Підпис та | |
| Інв. | |
| На зам. | |
| Підпис та | |
| Інв. | |

Керівник курсового проекту
ст. викл. каф. ЗІ

_____ Каплун В. А.

Розробив студент гр. 1БС-07
_____ Холодов І. В.

Курсову роботу захищено
з оцінкою _____

_____ 2010 р.
(Підпис керівника) (Дата)

(Підпис асистента)

Вінниця 2012

Додаток В
Приклад оформлення першого аркуша змісту

ЗМІСТ

| | |
|---|----|
| ВСТУП | 4 |
| 1 ВАРІАНТНИЙ АНАЛІЗ МЕТОДІВ ЗАХИСТУ ПРОГРАМ ВІД СТАТИЧНОГО ДОСЛІДЖЕННЯ | 6 |
| 1.1 Варіанти | 7 |
| 1.1.1 | 9 |
| ... | |
| 2 ЗАГОЛОВОК ДРУГОГО РОЗДІЛУ | 10 |
| 2.1 Заголовки підрозділів | 11 |
| ... | |
| 3 ЗАГОЛОВОК ТРЕТЬОГО РОЗДІЛУ | 24 |
| 3.1 Заголовки підрозділів | 28 |
| 3.1.1 .. | |
| | |
| ВИСНОВКИ | 32 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ | 33 |
| ДОДАТКИ | 34 |
| Додаток А. Технічне завдання | 34 |
| Додаток Б. Лістинг модуля захисту | 36 |

| | | | | | | | | |
|-----------|------|----------------|--------|------|---|---------------------|------|---------|
| | | | | | 08-20.ЗПЗ.010.13.107 ПЗ | | | |
| Змн. | Арк. | № докум. | Підпис | Дата | Розробка системи захисту програм від дизасемблювання шляхом утруднення логіки Пояснювальна записка | Літ. | Арк. | Акрушів |
| Розроб. | | Петров А.І. | | | | | 3 | 35 |
| Перевір. | | Каплун В.А. | | | | | | |
| Реценз. | | | | | | | | |
| Н. Контр. | | Каплун В.А. | | | | | | |
| Затверд. | | Лужецький В.А. | | | | ВНТУ, гр. 2 БС - 07 | | |

Додаток Г
Приклад оформлення тексту пояснювальної записки

ВСТУП

Стрімкий розвиток інформаційних технологій і використання їх у різних областях людської діяльності привів до того, що крім задач передачі, збереження й обробки інформації виникла важлива задача захисту інформації.

Питання захисту інформації набули першочергову необхідність, адже підприємницька діяльність в даний час тісно пов'язана з використанням інформаційних систем. Саме тому виникли такі проблеми, як незаконне використання алгоритмів, що є інтелектуальною власністю автора, несанкціоноване використання, модифікація, поширення і збут програмних продуктів. Комп'ютерні програми можуть розглядатися як інформація, яка потребує захисту.

...

Актуальність теми курсового проекту зумовлюється тим, що на сьогоднішній день системи захисту програмного забезпечення надзвичайно поширені і знаходяться в постійному розвитку завдяки розширенню ринку програмних продуктів і телекомунікаційних технологій, а також завжди існує взаємодія між розвитком захисту та способами його зламу.

Метою курсового проекту є створення додатку, який прив'язується до архітектури комп'ютера, і коли цієї прив'язки не виконано, щоб отримати доступ до даних необхідно ввести пароль. Програма, що розробляється в даному курсовому проекті, реалізовуватиме способи розмежування доступу, виключення доступу до інформації при копіюванні програми на інший комп'ютер. Програма може використовуватися як приклад для захисту інформації.

Задля досягнення мети курсового проекту необхідно виконати такі задачі: дослідити способи захисту та вибрати

| | | | | | | |
|------|------|----------|--------|------|-------------------------|-----------|
| | | | | | 08-20.ЗПЗ.010.13.107 ПЗ | Арк. 4 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

Додаток Д

Приклад оформлення індивідуального завдання

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

ЗАТВЕРДЖУЮ

Зав. кафедри ЗІ, д. т. н., проф.

_____ В. А. Лужецький

” ___ ” _____ 2012 р.

Протокол № ___ від _____

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на курсовий проект з дисципліни ”Захист програмного забезпечення”
студенту групи ІЗІ-04 факультету КСМ 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.

Короткий зміст частин проекту

1. **Текстова** (пояснювальна записка): Вступ. Аналіз літературних джерел і обґрунтування методу захисту. Розробка системи захисту (розробка схем та програмна реалізація). Тестування програмного засобу. Висновки. Список використаних джерел. Додатки.
2. **Графічна:** Схема функціонування програми. Схема алгоритму здійснення прив'язки до ПЕОМ. Схема ресурсів системи захисту. Візуальне подання результатів програми.

Дата видачі _____ 2012 р.

Керівник _____ Каплун В. А.

Завдання отримав _____ Холодов І. В.

Додаток Е

Приклад оформлення анотації

АНОТАЦІЯ

УДК 004.056

Гадалін М. С. “Розробка захисту від несанкціонованого копіювання шляхом прив’язки до архітектури ПЕОМ”. Курсовий проект – Вінниця: ВНТУ, 2011 – 37 с.

Українською мовою. Рисуноків – 10, таблиць – 2, бібліографій – 10.

В даному курсовому проекті розроблено власний алгоритм вбудованого захисту від несанкціонованого копіювання. На основі цього алгоритму розроблено програму, що реалізує захист певної інформації всередині себе. Розробка виконувалась у програмному середовищі Visual C#. Надана можливість модифікації коду програми з метою покращення та подальшого використання. При тестуванні програмного засобу доведено ефективність використання такого алгоритму і наведені можливості для його застосування на практиці.

Додаток Ж
Зразок оформлення технічного завдання

Додаток А

Вінницький національний технічний університет
Інститут інформаційних технологій та комп'ютерної інженерії
Факультет комп'ютерних систем та мереж

ЗАТВЕРДЖУЮ
Керівник, ст. викл. кафедри ЗІ
_____ В. А. Каплун
_____ 2012 р.

ТЕХНІЧНЕ ЗАВДАННЯ
на курсовий проект
з дисципліни "Захист програмного забезпечення"
на тему:
**«Розробка захисту програмного забезпечення
від несанкціонованого дослідження
шляхом ускладнення логіки»**
08-20.ЗПЗ.012.13.108 ТЗ

Вінниця 2012

1 Назва та область використання

Засіб для захисту від несанкціонованого дослідження програм шляхом здійснення утруднення логіки початкових кодів програм.

2 Основа для розробки

Робоча навчальна програма дисципліни "Захист програмного забезпечення" та робочий план.

3 Мета та призначення розробки

Підвищення ефективності захисту початкових кодів програм від статичного дослідження шляхом утруднення логіки.

4 Джерела розробки

4.1 Абашев А. А., Жуков И. Ю., Иванов М. А. Ассемблер в задачах защиты информации – М.: КУДИЦ-ОБРАЗ, 2004. – 544 с.

4.2

5 Вимоги до програми

5.1 Параметри розроблюваної системи захисту:

- механізм захисту – вбудований;
- принцип захисту – захист від статичного дослідження (дизасемблювання та декомпіляції);
- метод захисту – ускладнення логіки, заплутування графа потоку керування (внесення мертвого і надлишкового кодів);
- об'єкт захисту – початковий код програми мовою програмування С.

5.2 Програма повинна працювати без помилок і давати змогу:

- бачити вихідний та модифікований код програми,
- змінювати і комбінувати методи утруднення;
- користуватись динамічною системою підказок.

5.3 Програма має бути простою у використанні, мобільною для швидкого введення змін та оснащеною системою допомоги.

5.4 Вимоги до апаратного і програмного забезпечення, на якому повинна працювати програма:

- процесор – Pentium, Athlon і сумісні з ними;
- оперативна пам'ять – не менше 128 Мбайт;
- обсяг зовнішньої пам'яті для розташування додаткових файлів – не менше 10 Кбайт;
- середовище функціонування – ОС сімейства Windows.

5.5 Програма повинна мати зручний та естетичний графічний інтерфейс користувача.

5.6 Вимоги до техніки безпеки при роботі з програмою повинні відповідати існуючим вимогам та стандартам з техніки безпеки при користуванні комп'ютерною технікою.

6 Вимоги до програмної документації

6.1 Графічна і текстова документація повинна відповідати діючим стандартам

України.

6.2 Програма повинна супроводжуватись:

- “Інструкцією з технічного обслуговування” для окреслення складу апаратного забезпечення, на якому може експлуатуватися програма;
- “Інструкцією системному адміністратору” для користування нею під час інсталяції і перевірки правильності роботи;
- “Інструкцією програмісту” для користування нею при підготовці тих змін, які будуть внесені у програмний код для його захисту. Для цього необхідно дати правила формування вхідних даних;
- “Інструкцією оператору” для користування нею під час експлуатації здійснення захисту програми.

7 Стадії та етапи розробки

Робота з теми виконується в чотири етапи.

| Етап | Зміст | Початок | Закінчення | Результат |
|------|--|---------|------------|---|
| 1 | Аналіз літературних джерел, вибір і обґрунтування методів захисту | | | Розділ звіту, технічне завдання |
| 2 | Розробка схеми роботи програми в цілому і алгоритмів її складових | | | Розділ звіту, додатки у вигляді схем |
| 3 | Підбір програмних засобів для реалізації завдання. Створення початкового варіанта програми, проведення тестувань щодо коректності роботи | | | Макет програми. Розділ звіту |
| 4 | Створення кінцевого варіанта програми, тестування на ефективність захисту, розробка інструкцій для роботи з програмою | | | Діюча програма і остаточний звіт. Графічний матеріал |

8 Порядок контролю та прийому.

До прийому і захисту курсового проекту подається:

- остаточний звіт (пояснювальна записка);
- робоча програма;
- інструкції для роботи з програмою;
- графічний матеріал

Початок розробки

16.09.2012.

Крайній термін виконання курсового проекту

22.12.2012.

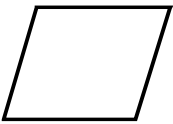
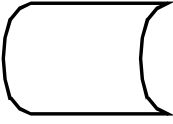


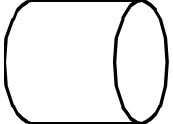



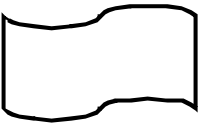
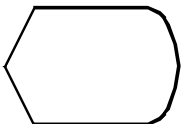
Розробив студент групи 2 БС_10 _____

Холодов І. П.




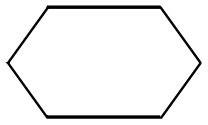
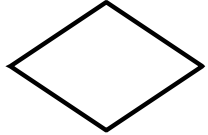
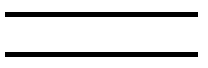
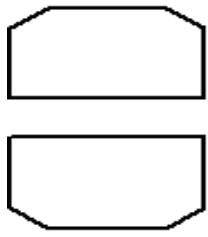
Додаток И

Символи даних, процесів і ліній

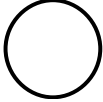

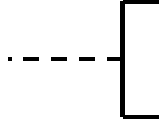
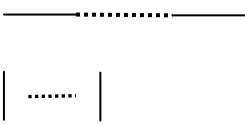
Таблиця И. 1 – Символи даних

| <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 не визначений |
| <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, магнітний барабан, гнучкий магнітний диск) |
| Документ |  | Символ відображає дані, подані на носії в легкій для читання формі (документ для оптичного або магнітного зчитування, мікрофільм, рулон стрічки з підсумковими даними, бланки введення даних) |
| Ручне введення |  | Символ відображає дані, що вводяться вручну під час оброблення з пристроїв будь-якого типу (клавіатура, перемикачі, кнопки, світлове перо, смужки зі штриховим кодом) |
| Карта |  | Символ відображає дані, подані на носії у вигляді карти (перфокарти, магнітні карти, карти зі зчитуваними мітками, карти з відривним ярликом, карти зі сканованими мітками). |
| Паперова стрічка |  | Символ відображає дані, подані на носії у вигляді паперової стрічки |
| Дисплей |  | Символ відображає дані, подані у візуальній людиночитабельній формі на носії у вигляді пристрою відображення (екран для візуального спостереження, індикатори введення інформації) |


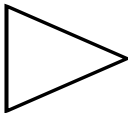


Таблиця И. 2 – Символи процесу

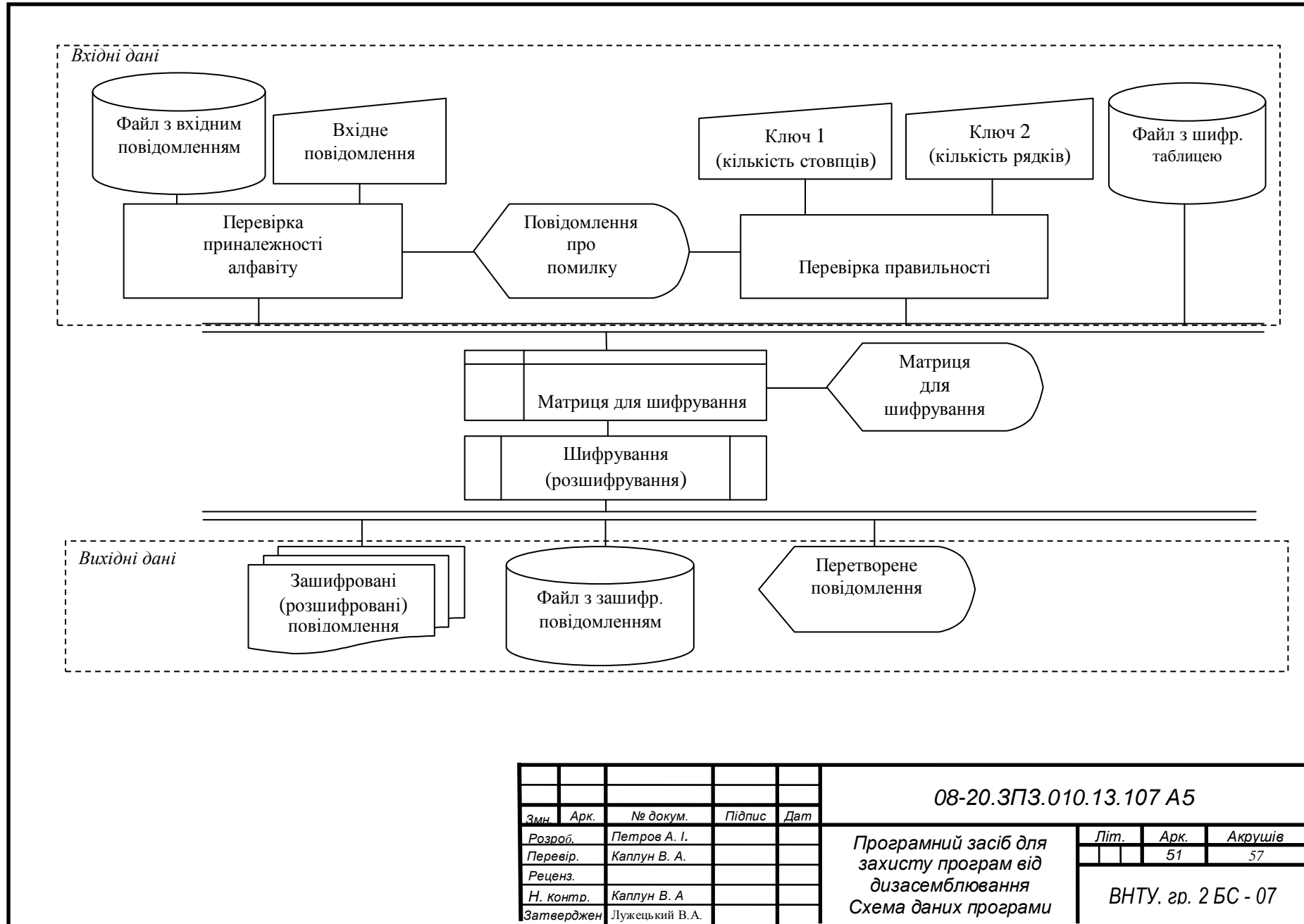
| <i>Основні символи процесу</i> | | |
|-----------------------------------|---|--|
| Процес |  | Символ відображає функцію обробки даних будь-якого вигляду (виконання певної операції або їх групи, що приводить до зміни значення, форми інформації). |
| <i>Специфічні символи процесу</i> | | |
| Підпорядкований 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, прирости, завершення поміщаються усередині символу на початку або в кінці залежно від розташування операції, що перевіряє умову |

Таблиця И. 3 – Спеціальні символи

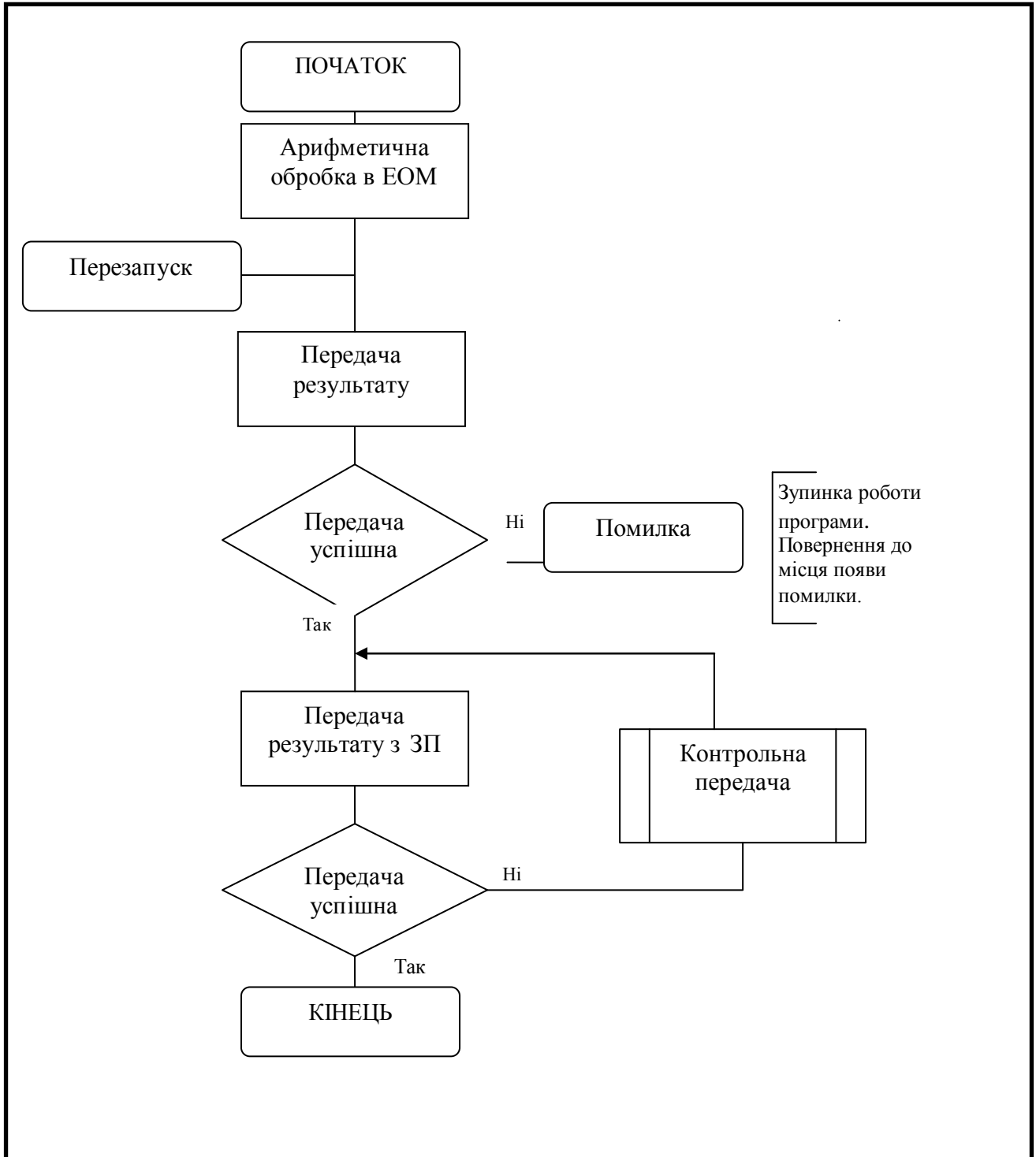
| | |
|---|---|
| <p>З'єднувач</p>  | <p>Символ відображає вихід в частину схеми і вхід з іншої частини цієї схеми і використовується для обривання лінії і продовження її у іншому місці. Відповідні символи-з'єднувачі повинні містити одне і те ж унікальне позначення</p> |
| <p>Термінатор</p>  | <p>Символ відображає вихід в зовнішнє середовище і вхід із зовнішнього середовища (початок або кінець схеми програми, зовнішнє використання і джерело або пункт призначення даних)</p> |
| <p>Коментар</p>  | <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> |
| <p>Пропуск</p>  | <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</p> |

Таблиця И. 4 – Символи ліній

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

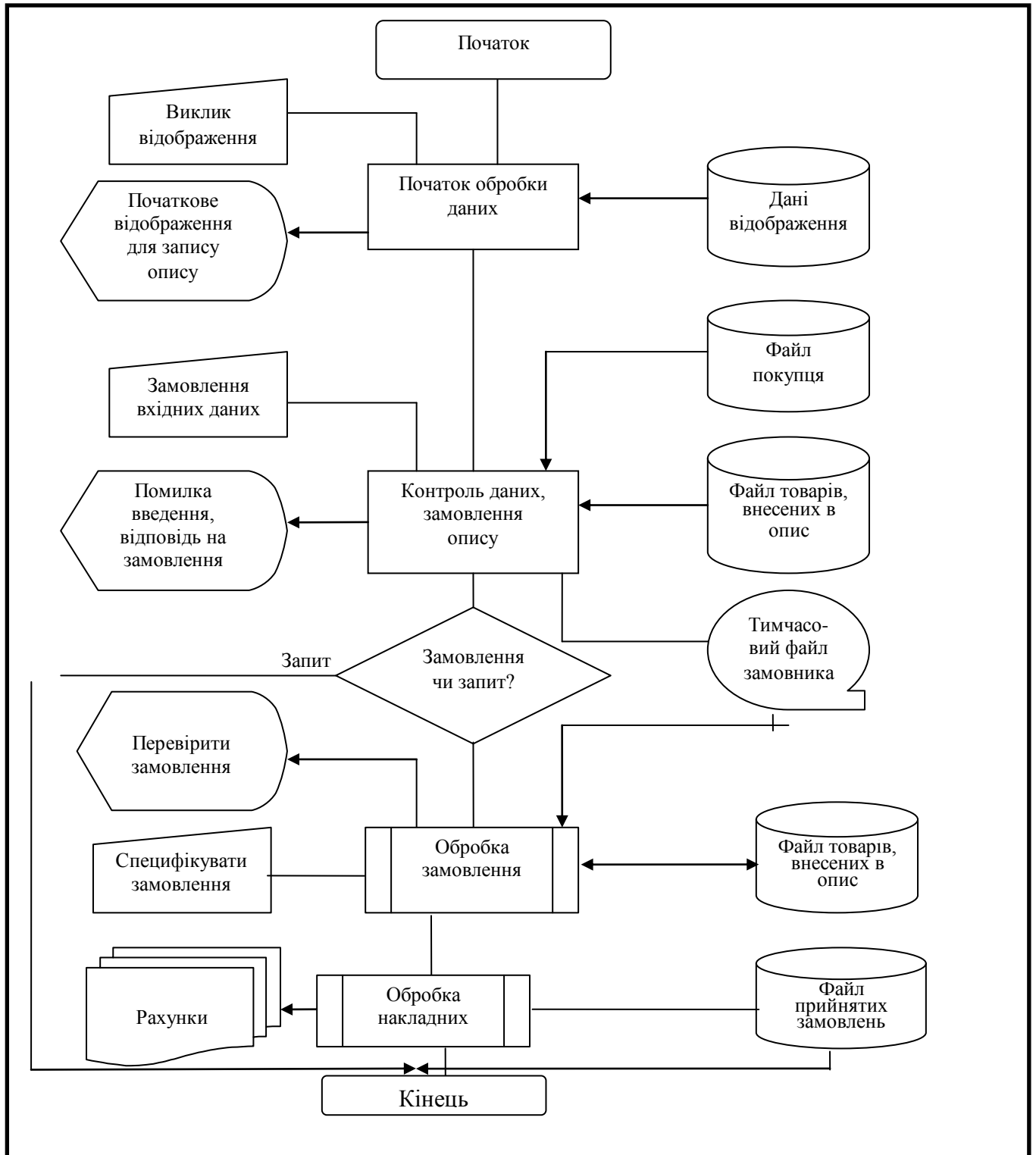


Додаток Л
Приклад схеми роботи програми



| | | | | | | | | |
|-----------|------|----------------|--------|------|---|---------------------|------|---------|
| | | | | | 08-20.ЗПЗ.010.13.107 А2 | | | |
| Змн. | Арк. | № докум. | Підпис | Дата | | | | |
| Розроб. | | Петров А. І. | | | Програмний засіб для захисту програм від дизасемблювання Схема роботи програми | Літ. | Арк. | Акрушів |
| Перевір. | | Каплун В. А. | | | | | 52 | 57 |
| Реценз. | | | | | | ВНТУ, гр. 2 БС - 07 | | |
| Н. Контр. | | Каплун В. А. | | | | | | |
| Затверд. | | Лужецький В.А. | | | | | | |

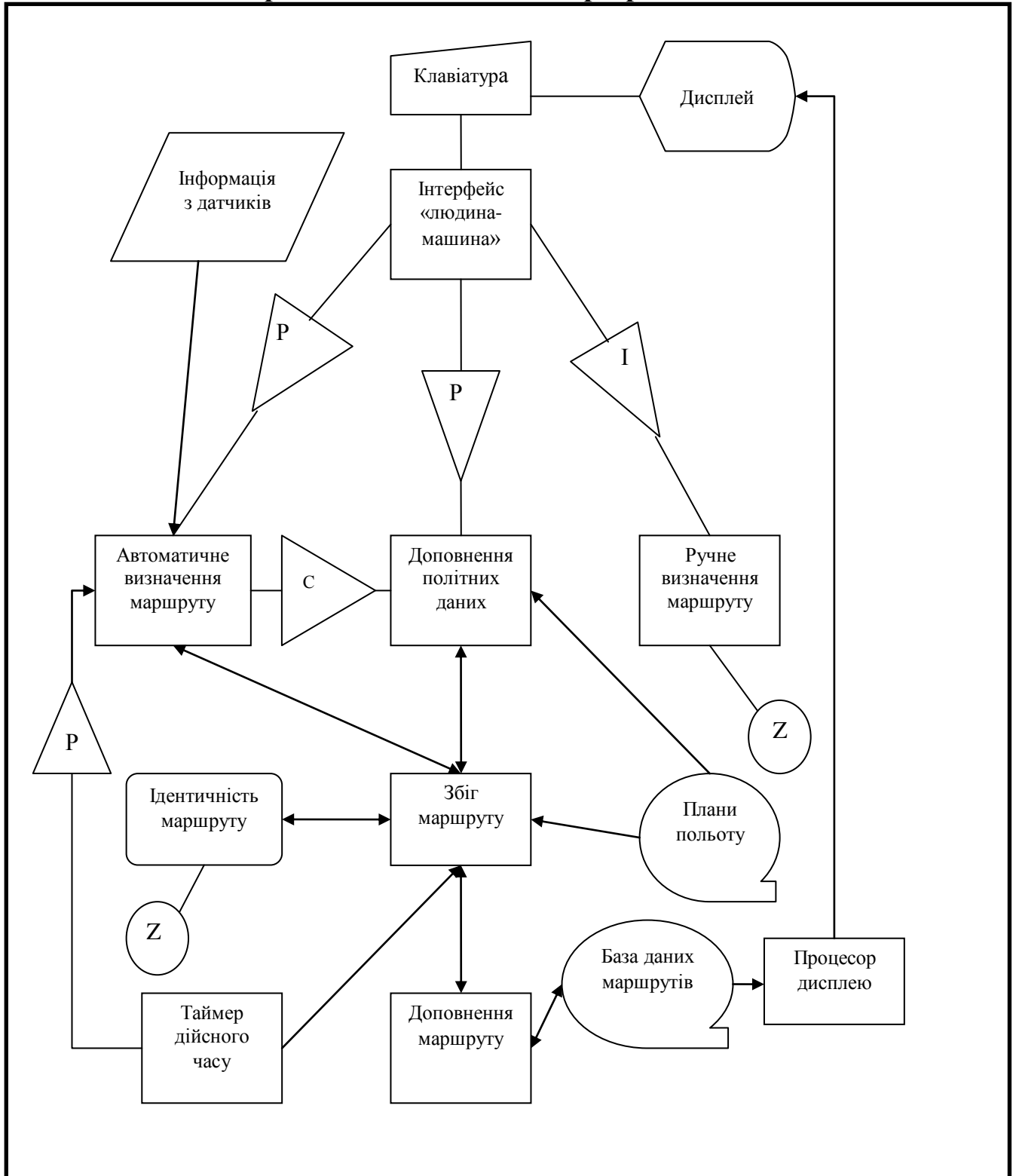
Додаток М Приклад схеми роботи системи



| | | | | |
|--------------------------|------|--------------|--|----------------------------------|
| 08-20.ЗПЗ.010.13.107 А 1 | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |
| Розроб. | | Петров А .І. | | |
| Перевір. | | Каплун В. А. | | |
| Реценз. | | | | |
| Н. Контр. | | Каплун В. А | | |
| Затверд. | | | | |
| | | | Програмний засіб для захисту програм від дизасемблювання Схема роботи системи захисту | Лім. 52 Арк. 57 Акрушів 57 |
| ВНТУ, гр. 2 БС - 07 | | | | |

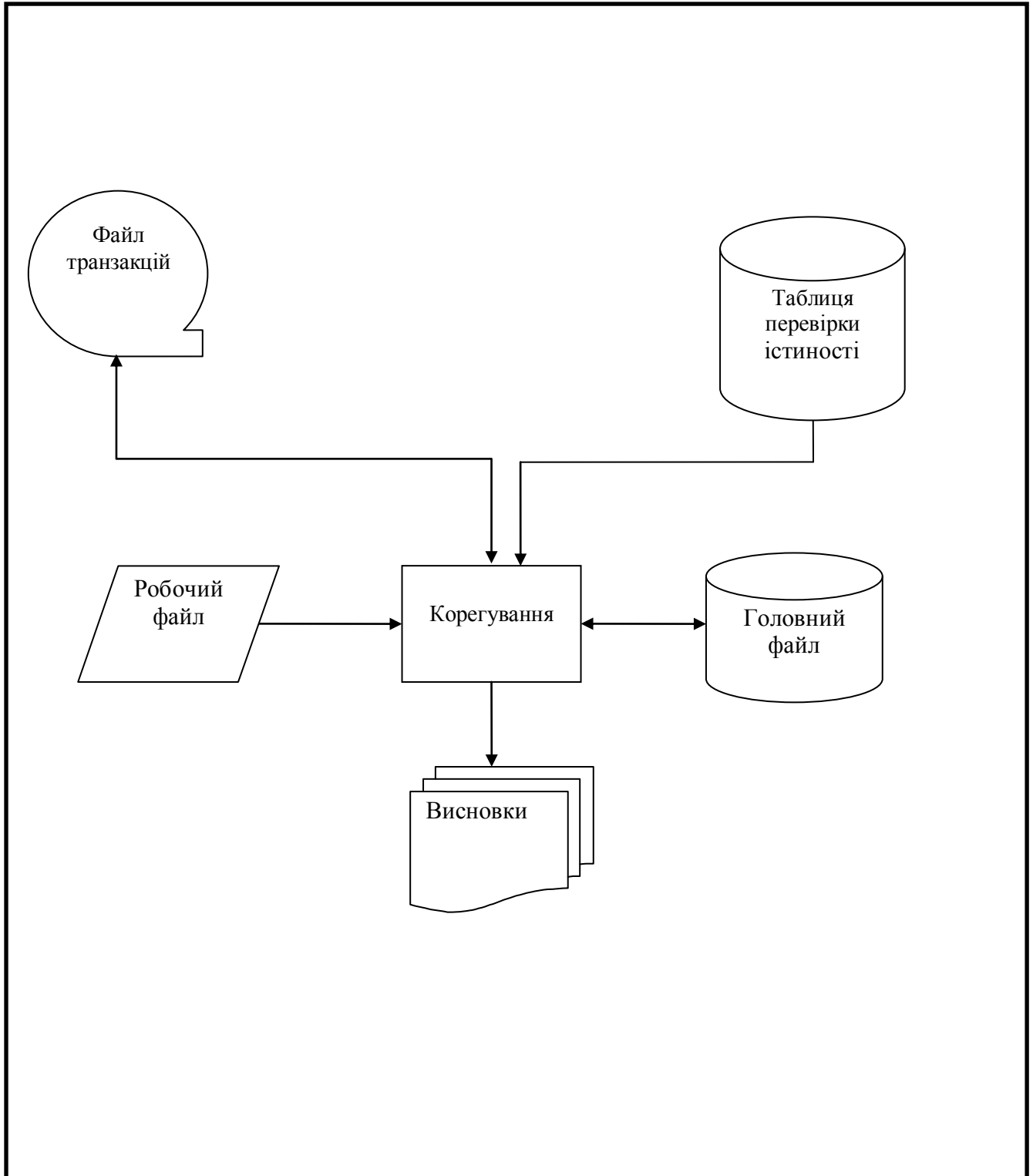
Додаток Н

Приклад схеми взаємодії програм



| | | | | | | | | |
|------|------|-----------|--------------|------|--|------|------|--------|
| | | | | | 08-20.ЗПЗ.010.13.107 А4 | | | |
| Змн. | Арк. | № докум. | Підпис | Дата | | | | |
| | | Розроб. | Петров А.І. | | Інтелектуальна система "людина-машина" | Літ. | Арк. | Акрюші |
| | | Перевір. | Каплун В. А. | | Схема взаємодії модулів програм | | 54 | 57 |
| | | Реєнз. | | | ВНТУ, зр. 2БС - 07 | | | |
| | | Н. Контр. | Каплун В. А. | | | | | |
| | | Затверд. | | | | | | |

Додаток II Приклад схеми ресурсів програми



| | | | | | | | | |
|------------------|-------------|---------------------|---------------|-------------|---|--------------------------------|-------------|----------------|
| | | | | | | 08-20.ЗПЗ.010.13.107 АЗ | | |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | Програмний засіб для захисту програм від дизасемблювання Схема ресурсів програми | <i>Літ.</i> | <i>Арк.</i> | <i>Акрюшів</i> |
| <i>Розроб.</i> | | <i>Петров А. І.</i> | | | | 55 | 57 | |
| <i>Перевір.</i> | | <i>Каплун В. А.</i> | | | | | | |
| <i>Реценз.</i> | | | | | | | | |
| <i>Н. Контр.</i> | | <i>Каплун В. А.</i> | | | | ВНТУ, зр. 2 БС - 07 | | |
| <i>Затверд.</i> | | | | | | | | |

