

Міністерство освіти і науки України
Кіровоградський національний технічний університет

Механіко-технологічний факультет
Кафедра програмування та захисту інформації

Системне програмування

Методичні вказівки до курсового проектування для студентів,

які навчаються за напрямом підготовки

6.050102 «Комп'ютерна інженерія»

Кіровоград
2015

ЗМІСТ

1. МЕТА І ЗАВДАННЯ КУРСОВОГО ПРОЕКТУ
2. ОСНОВНІ ЕТАПИ КУРСОВОГО ПРОЕКТУ
3. РЕКОМЕНДОВАНІ ТЕМИ ДО КУРСОВОГО ПРОЕКТУ
4. СТРУКТУРА І ОБСЯГ КУРСОВОГО ПРОЕКТУ5 ЗАВДАННЯ ТА КАЛЕНДАРНИЙ ПЛАН КУРСОВОГО ПРОЕКТУ
5. ЗАВДАННЯ ТА КАЛЕНДАРНИЙ ПЛАН КУРСОВОГО ПРОЕКТУ
6. АНАЛІЗ ЛІТЕРАТУРИ ТА СКЛАДАННЯ БІБЛІОГРАФІЇ
7. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КУРСОВОГО ПРОЕКТУ
8. ЗМІСТ РОЗДІЛІВ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ
9. ВИМОГИ ДО ОФОРМЛЕННЯ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ КУРСОВОГО ПРОЕКТУ
10. ЗАХИСТ КУРСОВОГО ПРОЕКТУ ТА КРИТЕРІЇ ОЦІНЮВАННЯ

1 МЕТА І ЗАВДАННЯ КУРСОВОГО ПРОЕКТУ

Метою виконання курсового проекту є закріплення та поглиблення знань, отриманих в процесі вивчення курсу *«Системне програмування»*, набуття практичних навичок та вмінь подальшого їх використання для проектування і розробки програмного забезпечення.

Основними завданнями курсового проекту є:

- узагальнення теоретичних знань, отриманих під час вивчення дисципліни *«Системне програмування»*, за допомогою поглибленого вивчення додаткової фахової літератури;
- набуття навичок практичного застосування теоретичних знань, проведення дослідження й аналізу існуючих програмних систем та побудова інформаційної системи предметної області.
- набуття практичних вмінь постановки інженерних задач, проектування складних систем та їх реалізація; розробка супровідної технічної документації до розробленої роботи.

Форма реалізації – робота, виконана відповідно до Державних стандартів та вимог, визначених специфікою кафедри програмування та захисту інформації Кіровоградського національного технічного університету.

Затвердження теми та контроль роботи студента над курсовим проектом здійснює керівник курсового проекту.

2 ОСНОВНІ ЕТАПИ КУРСОВОГО ПРОЕКТУ

1. Отримати тему курсового проекту або затвердити власну тему у керівника.
2. Скласти графік виконання курсового проекту, затвердити його у керівника та внести у календарний план завдання на курсовий проект.
3. Розпочати збір і обробку інформації за темою курсового проекту. Виконати чітку постановку задачі на виконання курсового проекту.
4. Розробити структурну, функціональну та інформаційно-логічну схеми, блок-схему, алгоритм програмного забезпечення.
5. Побудова інтерфейсу користувача (головна форма та меню користувача).
6. Розробка програмного продукту.
7. Оформлення пояснювальної записки курсового проекту.
8. Подати курсовий проект на попередній захист.
9. Виправити виявлені на перед захисті помилки та недоліки роботи, врахувати зауваження керівника і підготувати роботу до захисту.
10. Не пізніше встановленого календарним планом терміну подати курсовий проект до захисту.

3 РЕКОМЕНДОВАНІ ТЕМИ ДО КУРСОВОГО ПРОЕКТУ

1. Розробити програму для пошуку файлів за різними ознаками (аналог пошукової системи, вбудованої в Windows). Необхідно реалізувати всі можливості пошуку, передбачені в стандартній пошуковій системі.

2. Розробити програму DiskInfo, видає повну інформацію про диски, наявних на даному комп'ютері: наявність FDD, CD; обсяг HDD, логічні диски: розмір, обсяг зайнятого та вільного простору на кожному з них, серійний номер, розмір кластера, кількість секторів, розмір сектора. Передбачити можливість управління деякими з цих ресурсів (наприклад, висувати і засувати платформу для CD і т.п.).

3. Розробити програму SystemInfo, видає повну інформацію про даний комп'ютері: частоту процесора, обсяг оперативної пам'яті, наявність портів введення-виведення, дисководів, миші, клавіатури; системний каталог і каталог Windows. Передбачити можливість управління деякими з цих ресурсів (наприклад, відключати і підключати мишу, перепризначити програмно кнопки миші, керування дисками і т.п.)

4. Розробити програму VideoManager, що дозволяє отримати інформацію і управляти налаштуваннями відеосистеми даного комп'ютера (роздруківка здатність монітора, і т.п.).

5. Розробити програму FileInfo, видає повну інформацію про заданому файлі: тип, папка, ім'я, розмір, тимчасові характеристики, атрибути; для текстових файлів статистичні дані (кількість слів, кількість рядків, кількість абзаців, кількість символів з пробілами, к-ть слів без пробілів). Передбачити можливість зміни атрибутів файлів, а також копіювання, перейменування і переміщення файлу.

6. Розробити програму «Призначена для користувача оболонка», що виконує основні дії з файлами та каталогами. Для вибору дій використовувати

службові клавіші (F1, F2, і т.д.), командний рядок або меню, панель інструментів. (Аналог Norton Commander).

7. Розробити програму Task Manager, тобто додаток, що виводить інформацію про запущені в даний момент процеси.

8. Розробити програму, яка видає відомості про стан основної пам'яті: визначати стан адресного простору: передана чи за цією адресою фізична пам'ять і доступний він для операцій читання або запису; повідомляти про стан системної пам'яті і віртуального адресного простору в тому чи іншому процесі.

9. Розробити програму, яка демонструвала б можливості створення і управління параметрами вікна деякого додатка, в тому числі змінювати форму і розміри вікна, колір і тип ліній рамки і т.д. (Повинна бути вивчена функція, яка малює вікно).

10. Пошук файлів, які дублюють один одного, але перебувають в різних каталогах. Слід знайти файли з однаковими іменами, потім порівняти їх вміст і запропонувати користувачеві можливість видалення файлів дублікатів.

11. Створення диспетчера задач Windows з можливістю збору статистичних даних.

12. Емуляція роботи операційної системи з диспетчеризації процесів.

13. Емуляція роботи операційної системи по виділенню пам'яті RAM і віртуальної пам'яті.

14. Створення утиліти для роботи з мережним адаптером.

15. Створення програми, що використовує процесори відеоадаптера для вирішення неграфічних завдань.

16. Створення СОМ-модуля, що інкапсулює клас для апроксимації одновимірного числового ряду і додаток для його тестування.

17. Створення СОМ-модуля, що інкапсулює клас для двовимірної апроксимації матриці і додаток для його тестування.

18. Створення COM-модуля, що інкапсулює клас для роботи з мережею через протокол TCP і додаток для його тестування.
19. Створення проміжного драйвера пристрою для роботи з мережевою картою.
20. Створення проміжного драйвера пристрою, що реалізує основні функції для роботи з USB.
21. Створення програми, що виводить повну інформацію про жорсткому диску.
22. Створення проміжного драйвера пристрою, що реалізує основні функції для роботи з COM-портом.
23. Створення проміжного драйвера для роботи з мережею з використанням протоколу SSL.
24. Створення клієнт-серверного додаток для обміну повідомленнями по вільному протоколу.
25. Створення програми діагностики помилок.
26. Створення редактора віконних інтерфейсів.
27. Створення конвертера форматів.
28. Створення інтерпретатора програм на мові високого рівня.
29. Створення інтерпретатора файлів в заданому форматі.
30. Створення компілятора мови високого рівня в об'єктний код.
31. Створення диспетчера задач.

4 СТРУКТУРА І ОБСЯГ КУРСОВОГО ПРОЕКТУ

Документ курсового проекту повинен мати наступну структуру:

1 – Титульний аркуш.

2 – Завдання на курсовий проект та календарний план

3 – Пояснювальна записка:

- **Вступ** (2-5% від загального об'єму пояснювальної записки).

- **Розділ 1** Призначення і область застосування програмного забезпечення(10-15%).

- **Розділ 2** Огляд існуючого програмного забезпечення. Постановка задачі. (10-15%).

- **Розділ 3** Опис і обґрунтування проектних рішень щодо розробки програмного забезпечення(15-20%).

- **Розділ 4** Експериментальне підтвердження вірності програмних та проектних рішень. Реалізація роботи (45-50%),

- **Розділ 5** Інструкція користувача (5-10%).

- **Розділ 6** Основні висновки(5-10%).

4 Перелік скорочень, символів і спеціальних термінів.

5 Список літератури. 6 Додатки.

В залежності від особливостей теми курсового проекту, за погодженням з керівником можливе уточнення змісту розділів (текстові, графічні документи: функціональну схему системи та блок-схему алгоритму програм, тощо), введення нових або об'єднання деяких розділів пояснювальної записки курсового проекту. Пояснювальна записка курсового проекту повинна мати не менше 30 сторінок машинописного тексту, який виконано виключно чорним кольором і чітким шрифтом (рисунки та інші графічні матеріали – чорно-білі). На сторінці має бути не більше 32 рядків і не більше 70 символів в рядку.

Рекомендований шрифт – Times New Roman, розмір шрифту – 14 пт, міжрядковий інтервал – 1.5 пт, абзацний відступ – 15 мм.

5 ЗАВДАННЯ ТА КАЛЕНДАРНИЙ ПЛАН КУРСОВОГО ПРОЕКТУ

Завдання на курсовий проект видається студентові перед початком роботи над курсовим проектом і затверджується керівником курсового проекту.

Календарний план складається студентом та затверджується керівником курсового проекту.

Рекомендовано наступний план курсового проекту:

1. Вивчення предметної області інформаційної системи та побудова інформаційно-логічної моделі.
2. Побудова структури системи.
3. Побудова схеми даних відповідно до інформаційно-логічної моделі.
4. Побудова інтерфейсу користувача, функціональної схеми системи.
5. Розробка програмного забезпечення.
6. Побудова презентації інформаційної системи.
7. Попередній захист курсового проекту.
8. Захист курсового проекту.

Термін виконання етапів курсового проектування визначається студентом, погоджується з керівником, вноситься до календарного плану листа-завдання та підписується студентом і керівником.

Під час виконання курсового проекту кожен етап, відповідно до календарного плану, узгоджується з керівником, який робить відповідний запис у графі «Примітки» календарного плану.

6 АНАЛІЗ ЛІТЕРАТУРИ ТА СКЛАДАННЯ БІБЛІОГРАФІЇ

Під час виконання курсового проекту необхідно підібрати актуальну літературу та інші інформаційні джерела, які стосуються теми курсового проекту. Після їх вивчення та аналізу інформаційних джерел, що будуть використані для реалізації роботи, складається бібліографія, яка подається у розділі “Список літератури” пояснювальної записки курсового проекту.

Самостійний пошук фахової літератури бажано здійснювати в бібліотеці університету або Обласній універсальній науковій бібліотеці імені Д.І. Чижевського (м. Кіровоград, вул. Карла Маркса, 24). Особливу увагу слід звернути на нові публікації та періодичні фахові видання - газети, вісники, журнали -, в яких можна знайти найновіші розробки в галузі комп’ютерної техніки, результати досліджень спеціалістів тощо. Також важливо опрацювати джерела сучасної зарубіжної літератури.

Окрім друкованих видань допускається використання електронних інформаційних джерел глобальної інформаційної мережі Internet. Проте перевагу слід надавати офіційним web-сайтам. Найвідоміший серед них - електронний фонд Національної бібліотеки України імені В.І. Вернадського (НБУВ). Він доступний за адресою:

www.nbuv.gov.ua

Після завантаження головної сторінки сайту необхідно обрати посилання “Електронний фонд” → “Розширений пошук”.

В електронній базі даних НБУВ пошук електронних документів здійснюється шляхом введення пошукових термінів (прізвища автора книги чи публікації, слів з назви документа, рефератів) або за 32-ма тематичними розділами. У режимі розширеного пошуку (рисунок 6.1) користувачу надаються додаткові параметри пошуку: можливість визначити вид публікації (книга, автореферат дисертації, стаття з періодичного видання); уточнити результати пошуку за роком видання (з 1997 до поточного року). У формі розширеного пошуку поля “Автор” та “Слова з назви” розділено, оскільки

запит за ключовим словом у звичайному режимі пошуку (“пошук скрізь”) матиме низьку релевантність, а результат пошуку в більшості випадків користувача не задовільнить.

Після встановлення наявності в електронній базі даних потрібного документа (виводиться у вигляді списку) можна отримати повний текст публікації, що зберігається в окремому файлі.

Обов’язковим для студента є дотримання загальних правил обробки літератури та інформаційних джерел. Спочатку необхідно ознайомитися з основною літературою (підручниками, журналами, збірниками наукових праць тощо), потім вивчаються електронні джерела інформації, розміщені в мережі або розповсюджуються на електронних носіях.

Відібрана література підлягає уважній, ретельній обробці. Попереднє ознайомлення включає огляд змісту, читання передмови, анотації. Розділи, які мають особливе значення для курсового проекту, вивчаються більш глибоко, робляться нотатки, які будуть використані під час оформлення пояснювальної записки. Слід звертати особливу увагу на ідеї та принципи розв’язку інженерних задач, які є близькими до обраної теми та можуть лягти в основу виконання курсового проекту.

Зроблені під час вивчення літератури нотатки групуються в окремі блоки (за розділами). В подальшому складання бібліографії полегшить роботу над оформленням пояснювальної записки.

Бібліографічний опис у списку літератури повинен містити прізвище та ініціали автора, назву, видавництво, рік видання, кількість сторінок. Нариклад, Мельник А.О. Архітектура комп’ютера. Підручник. - Луцьк: “Волинське обласне видавництво”, 2007. - 470 с.

У тексті пояснювальної записки курсового проекту рекомендовано вказувати в кінці речення (абзацу) посилання на використану літературу за допомогою квадратних дужок “[]”, в яких вказується номер джерела зі списку

літератури. Наприклад, [3] - посилання на джерело зі списку літератури під номером 3; [2-4] - посилання на джерела під номерами 2, 3 та 4; [1-3, 6] - посилання на джерела під номерами 1, 2, 3, 6.

7. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КУРСОВОГО ПРОЕКТУ

Розробка програмного забезпечення роботи здійснюється після аналізу літератури, розробки структурних, функціональних схем та блок-схеми алгоритма програми, інформаційно-логічної моделі даних.

Першим етапом є вибір мови розробки ПЗ, що задовольняє вимогам даної інформаційної системи.

Другий етап - розробка інтерфейса користувача. Інтерфейс повинен мати такі властивості: природність, узгодженість, дружність, принцип “зворотнього зв’язку”, простота, гнучкість, естетична привабливість, чіткість.

Ефективність інтерфейса полягає у швидкому розвитку в користувачів простої концептуальної моделі взаємодії. Це досягається через узгодженість. Концепція узгодженості полягає в тому, що під час роботи з ПК у користувача формується система очікування однакових реакцій на однакові дії, що постійно підкріплює призначену для користувача модель інтерфейсу.

Інтерфейс може бути узгоджений в трьох аспектах: фізичному, синтаксичному і семантичному.

Фізична узгодженість відноситься до технічних засобів: схема клавіатури, розташування клавіш, використання миші. Наприклад, для клавіші РЗ фізична узгодженість має місце, якщо вона завжди знаходиться в одному і тому ж місці, незалежно від обчислювальної системи. Аналогічно кнопка вибору миші буде фізично узгоджена, якщо вона завжди розташовується під вказівним пальцем.

Синтаксична узгодженість відноситься до послідовності й порядку появи елементів на екрані (мова зображень), послідовності запитів (мова дій). Наприклад, синтаксична узгодженість матиме місце, якщо заголовок панелі завжди розміщується в центрі і вгорі панелі.

Семантична узгодженість відноситься до значення елементів, складових інтерфейсу. Наприклад, що означає “Вихід”? Де користувачі роблять запит на “Вихід” і що потім відбувається?

Природність інтерфейса - найважливіша його характеристика, оскільки він не змушує користувача істотно змінювати звичні для нього способи розв’язку задачі. Це, зокрема, означає, що повідомлення і результати, які видаються програмним продуктом, не повинні вимагати додаткових пояснень. Доцільно також зберегти систему позначень і термінологію, які використовувались в даній предметній області.

Використання знайомих користувачеві понять і образів (метафор) забезпечує інтуїтивно зрозумілий інтерфейс при виконанні його завдань. Разом з тим вони не повинні обмежувати їх машинну реалізацію повною аналогією з однойменними об’єктами реального часу.

Користувачі зазвичай вивчають особливості роботи з новим програмним продуктом методом спроб і помилок. Ефективний інтерфейс повинен враховувати такий підхід. На кожному етапі роботи він повинен виконувати тільки відповідний набір дій і попереджати користувачів про ті ситуації, де вони можуть зашкодити системі або даним; ще краще, якщо у користувача є можливість відмінити або виправити виконані дії.

Навіть за наявності добре спроектованого інтерфейсу користувачі можуть робити ті або інші помилки. Ці помилки можуть бути як “фізичного” типу (випадковий вибір неправильної команди або даних) так і “логічного” (ухвалення неправильного рішення на вибір команди або даних). Ефективний інтерфейс повинен дозволяти запобігати ситуаціям, які, ймовірно, закінчатся помилками.

Він також повинен уміти адаптуватися до потенційних помилок користувача і полегшувати йому процес усунення наслідків таких помилок.

Завжди забезпечуйте зворотний зв’язок для дій користувача. Кожна дія користувача повинна отримувати візуальне, а іноді і звукове підтвердження

того, що програмне забезпечення сприйняло введену команду; при цьому вид реакції, за можливістю, повинен враховувати природу виконаної дії.

Зворотний зв'язок ефективний в тому випадку, якщо він реалізується своєчасно, тобто якомога ближче до точки останньої взаємодії користувача з системою. Коли комп'ютер обробляє завдання, що надійшло, корисно надати користувачеві інформацію щодо стану процесу, а також можливість перервати цей процес у разі потреби.

Інтерфейс повинен бути простим. При цьому мають на увазі не спрощеність, а забезпечення легкості в його вивченні і у використанні. Крім того, він повинен надавати доступ до всього набору функціональних можливостей, передбачених даною програмою. Реалізація доступу до широких функціональних можливостей і забезпечення простоти роботи суперечать один одному. Розробка ефективного інтерфейса покликана збалансувати ці цілі.

Один з можливих шляхів підтримки простоти - зображення на екрані інформації, мінімально необхідної для виконання користувачем чергового кроку завдання.

Інший шлях до створення простого, але ефективного інтерфейсу - розміщення і представлення елементів на екрані з урахуванням їх смислового значення і логічного взаємозв'язку. Це дозволяє використовувати в процесі роботи асоціативне мислення користувача.

Під гнучкістю інтерфейсу розуміють здатність враховувати рівень підготовки і продуктивність праці користувача. Властивість гнучкості припускає можливість зміни структури діалогу і/або вхідних даних.

Проектування візуальних компонентів є найважливішою складовою розробки програмного інтерфейсу. Коректне візуальне представлення об'єктів забезпечує передачу дуже важливої додаткової інформації про поведінку і взаємодію різних об'єктів.

В той же час слід пам'ятати, що кожен візуальний елемент, який з'являється на екрані, потенційно вимагає уваги користувача, яка, як відомо, не безмежна. Слід забезпечити формування на екрані такого середовища, яке не тільки сприяло б розумінню користувачем представленої інформації, але і дозволяло б зосередитися на найбільш важливих її аспектах.

До візуальних атрибутів інформації, що відображається, відносяться:

- взаємне розташування і розмір об'єктів, що відображаються;
- палітра;
- засоби привертання уваги користувача.

Проектування розміщення даних на екрані передбачає виконання наступних дій:

- визначення складу інформації (елементів, компонентів, об'єктів тощо), яка повинна з'являтися на екрані;
- вибір формату представлення цієї інформації;
- визначення взаємного розташування даних (чи об'єктів) на екрані;
- вибір засобів привертання уваги користувача;
- розробка макету розміщення даних на екрані;
- оцінка ефективності розміщення інформації.

Загальні принципи розташування інформації на екрані повинні забезпечувати для користувача:

- можливість переглядання екрану в логічній послідовності;
- простоту вибору потрібної інформації;
- можливість ідентифікації зв'язаних груп інформації;
- помітність виключних ситуацій (повідомлень про помилки або попередження);
- можливість визначити, яка дія з боку користувача потрібна (і чи потрібна взагалі) для продовження виконання завдання.

Питання про те, яка інформація підлягає відображенню, вирішується залежно від специфіки теми курсового проекту.

В даному випадку істотну роль грає правильне розбиття завдання на операції (етапи), які не вимагають одночасної присутності великого об'єму даних на екрані.

Ця умова витікає з такої психофізіологічної особливості людини, як обмеженість його короткочасної пам'яті, здатної зберігати одночасно не більше п'яти-дев'яти об'єктів.

Якщо вся інформація початкового документа не вміщується на одному екрані, деякі елементи даних можуть повторюватися на інших екранах для збереження цілісності і послідовності обробки. Як правило, повторювана інформація не повинна міняти свого розташування на всіх кроках виконання завдання.

Властивість природності інтерфейсу припускає, що інформація відображається на екрані у вигляді, придатному для безпосереднього використання. Не слід примушувати користувача додатково обробляти цю інформацію (наприклад, уточнювати за допомогою довідників значення кодів, форматів, проводити які-небудь перетворення, перерахунки тощо).

Формат для виведення дати, часу і інших подібних стандартизованих даних повинен бути загальноприйнятим, а не індивідуальним для даної системи. Загальноприйнята система поєднання великих і малих літер в тексті покращує його сприйняття.

Виділення інформації - це використання таких атрибутів, які дозволяють привернути увагу користувача до певної області екрану. В якості подібних атрибутів можуть виступати: колір символів, колір фону, рівень яскравості, мерехтіння і застосування різних шрифтів для символів, що виводяться. Часто для виділення інформації використовують підкреслення, висновок в інверсному вигляді, різні рамки і "тіні".

Ефект застосування цих атрибутів різний, а їх поєднання – часто непередбачуване і залежать від індивідуальних особливостей користувача. Основна рекомендація: слід прагнути використовувати мінімально необхідну кількість атрибутів.

Програмна реалізація є **третім етапом** розробки програмного забезпечення. На цьому етапі за допомогою обраної мови програмування та на основі розробленого алгоритма здійснюється написання програми.

Важливою складовою лістинга програми є *коментарі*. Вони не тільки полегшать “читання” та відлагодження програми, але й є ознакою гарного стилю програмування.

Обов’язковим компонентом програмного забезпечення є меню “Допомога” (“Довідка”, “Help”), в якому розміщується інструкція користувача (з детальним описом принципів роботи з програмою). Крім того, необхідно додати меню “Автор”, що містить довідникову інформацію про розробника програмного забезпечення (прізвище, ім’я, по батькові, назва ВНЗ, факультет, група, рік розробки тощо).

Заключним, **четвертим, етапом** розробки програмного забезпечення є відлагодження програмного забезпечення, після чого його необхідно погодити та затвердити у керівника курсового проекту. Рекомендовано зробити скріни основних вікон програми, описати основні їх компоненти та включити цей матеріал до інструкції користувача ПЗ.

8. ЗМІСТ РОЗДІЛІВ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

1 Титульний аркуш, завдання

Використовувати стандартизований титульний лист.

2 Пояснювальна записка. Вступ

Обґрунтовується актуальність теми курсового проекту, здійснюється вступ до розробки програмного забезпечення, наводяться докази важливості та необхідності програмного забезпечення (пристрою, системи), яке є результатом реалізації роботи. Приводиться короткий огляд і аналіз літератури (1-2 абзаци) з посиланням на використані джерела та стислий висновок щодо розробки програмного забезпечення.

2.1 Огляд існуючого програмного забезпечення

Здійснюється огляд існуючого програмного забезпечення, систем, приладів, основних напрямків розвитку існуючих систем. Наводиться аналіз їх характеристик, властивостей, недоліків і переваг, на основі чого робиться висновок про доцільність (необхідність) розробки програмного забезпечення відповідно до теми курсового проекту. Здійснюється короткий опис обраного інструментарію. Обґрунтовуються основні принципи розробки програмного забезпечення, його переваги над іншими для вирішення обраної задачі. Здійснюється чітка постановка задачі.

2.2 Опис поставленої задачі, призначення програмного продукту.

Детально описується хід теоретичної побудови моделі програмного забезпечення, її структурної, функціональної схеми, принципів роботи. Приводиться область її застосування. Наводиться обґрунтування

запропонованих рішень. Робляться стислі висновки щодо проектування системи в цілому.

2.3 Огляд існуючих систем, обґрунтування вибору принципу розробки і методики побудови системи. Постановка задачі. Приводяться стислі характеристики існуючих систем. Приводиться аналіз їхніх властивостей. На підставі проведеного аналізу робиться висновок про доцільність (необхідність) проектування системи згідно з теми курсового проекту. Обґрунтовуються основні принципи проектування системи, методика проектування. Здійснюється постановка задачі на проектування системи.

2.4 Опис і обґрунтування проектних рішень щодо проектування системи . Описується хід теоретичної побудови моделі проекту, приводиться їх обґрунтування. Робляться стислі висновки.

2.5 Основні висновки.

Дається стисла характеристика розробленої системи за всіма основними параметрами. Відзначаються основні нововведення, методика і засоби реалізації проекту. Приводяться викладення про ефективність, новаторство і основні переваги розробленої системи.

3 Перелік скорочень, символів і спеціальних термінів.

Наводиться перелік скорочень, символів і спеціальних термінів, які використані у пояснювальній записці курсового проекту, їх опис, повне розшифрування, коротке пояснення.

4 Список літератури.

Наводиться нумерований список літератури та інформаційних джерел, які використано для реалізації курсового проекту. Рекомендується

використовувати виключно друковані видання: підручники, навчальні посібники, фахові журнали, збірники наукових праць.

5 Інструкція користувача. Описуються основні об'єкти програмного забезпечення та методика їх використання. Розробляється детальна інструкція використання розробленого програмного забезпечення користувачем (також додається до головного меню програми у вигляді меню «допомоги» або «Довідка»).

6 Додатки

Містять додаткові матеріали, що не входять до тексту пояснювальної записки курсового проекту.

9. ВИМОГИ ДО ОФОРМЛЕННЯ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ КУРСОВОГО ПРОЕКТУ

Виконання і оформлення пояснювальної записки курсового проекту здійснюється відповідно до Державних стандартів:

1. ГОСТ 2.105-95 ЕСКД. Загальні вимоги до текстових документів.
2. ГОСТ 19.404-79 ЕСПД. Пояснювальна записка. Вимоги до змісту і оформлення.
3. ГОСТ 2.106-68 ЕСКД. Текстові документи.
4. ГОСТ 23501.4-79 САПР. Загальні вимоги до програмного забезпечення.
5. ГОСТ 24.203-80 АСУ. Вимоги до змісту загальносистемних документів.
6. ГОСТ 24.301-80 АСУ. Загальні вимоги до виконання текстових документів.
7. ГОСТ 19.401-78 ЕСПД. Текст програми.
8. ГОСТ 19.402-78. Опис програми.
9. ГОСТ 19.106-78 ЕСПД, Вимоги до програмних документів, виконаних друкованим засобом.
10. ГОСТ 2.701-76 ЕСКД. Схеми. Вигляди і типи. Загальні вимоги до виконання.
11. ГОСТ 2.708-81 ЕСКД. Правила виконання електричних схем цифрової обчислювальної техніки.
12. ГОСТ 19.701-90 ЕСПД. Схеми алгоритмів і програм. Позначки умовні графічні і ін.

Вимоги до документів, форми і правила виконання регламентуються ГОСТ 2.105-95.

Відповідно до листа Міністра освіти України №1/9-73 від 01.03.99 курсові та дипломні проекти, обов'язкові домашні та розрахунково - графічні роботи повинні виконуватись державною мовою.

Зміст пояснювальної записки повинен бути викладений від третьої особи однини. Наприклад, "...в роботі досліджено...", "...програмне забезпечення розроблено..." тощо. Текст та ілюстрації виконуються виключно чорними чорнилами (рисунки, графіки, схеми - чорно-білі).

Відстань від рамки форми до тексту на початку та в кінці рядків - не менше 3 мм. Відстань від верхнього або нижнього рядка тексту або до верхньої або нижньої рамки повинно бути не менш 10 мм. Абзац в тексті починають відступом 15-17 мм.

Пояснювальна записка (код 81) - програмний документ, входить до складу документів на стадії розробки ескізного і технічного проектів програми. Зміст і оформлення документа регламентується ГОСТ 19.404-79. Документ містить схему алгоритма, загальний опис алгоритма і (або) функціонування програми, а також обґрунтування прийнятих техніко- економічних рішень. Укладання інформаційної частини є необов'язковим. Пояснювальна записка повинна містити розділи: вступ; призначення і область застосування; технічні характеристики; очікувані техніко-економічні показники; джерела, що використовувалися при розробці. У ВНЗ не допускається аркуші паперу для тексту в пояснювальній записці оформляти без рамки і основного напису. Сторінки тексту, що мають розмір більше формату А4, не нумеруються, але враховуються як дві сторінки в послідовності нумерації сторінок ПЗ. В програмному документі викладаються відомості про результати виконаної роботи з розробки курсового проекту з додатком (за необхідності) графічних документів. В відповідності з означеними стандартами пояснювальна записка виконується за формами 5 і 5а на форматі А4 (210x297мм) і з титульним аркушем за формою ГОСТ 2.105-79.

10. ЗАХИСТ КУРСОВОГО ПРОЕКТУ ТА КРИТЕРІЇ ОЦІНЮВАННЯ

Захист курсового проекту здійснюється після його повного виконання і перед захисту у термін, встановлений календарним планом.

Оцінювання курсового проекту здійснюється за 100-бальною системою. Критерії оцінювання повноти виконання та захисту курсового проекту наведено в таблиці 10.1, оцінювання етапів (розділів) курсового проекту - таблиця 10.2.

Таблиця 10.1 - Критерії оцінювання повноти виконання і захисту

Кількість	Вимоги до виконання та захисту курсового проекту
100-90	Етапи курсового проекту виконано вчасно, подано на передзахист і захист у встановлені календарним планом терміни. Вчасно розроблено, погоджено і затверджено алгоритм та програмне забезпечення. Розробка має творчий підхід та оригінальні проектні рішення з елементами наукової новизни. Пояснювальну записку оформлено відповідно до встановлених вимог, теоретико-практичні аспекти розділів розкрито повністю, подано якісний графічний матеріал, структурні, функціональні схеми, використано вітчизняні та зарубіжні фахові видання. Захист передбачає чітко поставлену мету роботи, результати проведених досліджень і аналізу існуючого програмного забезпечення, опис і обґрунтування прийнятих проектних рішень щодо розробки програми, висновки; студент повинен мати ґрунтовні відповіді на всі додаткові питання.
82-89	Етапи проектування виконано вчасно, подано на передзахист і захист у встановлені календарним планом терміни. Вчасно розроблено, погоджено і затверджено алгоритм та програмне забезпечення. Пояснювальна записка оформлена відповідно до вимог, теоретико-практичні аспекти розділів розкрито повністю. Захист передбачає чітко поставлену мету роботи, результати дослідження і аналізу існуючого програмного забезпечення, опис проектних рішень щодо розробки програми, висновки. Під час захисту студент в повному обсязі виклав теоретико-практичний матеріал, на поставлені додаткові питання мав ґрунтовні відповіді.
75-81	Етапи виконано вчасно, проект подано на передзахист і захист у встановлені календарним планом терміни. Алгоритм та програмне забезпечення є недосконалим та потребує подальшого вдосконалення, але відповідає темі проекту. Пояснювальна записка оформлена відповідно до вимог, теоретико-практичні аспекти розділів розкрито повністю. Під час захисту проекту студент не в повному обсязі викладав матеріал, на поставлені питання мав нечіткі відповіді.

67-74	Проект подано до захисту з порушенням встановленого календарним планом термін. Алгоритм та програмне забезпечення є недосконалим, потребує подальшого вдосконалення, але відповідає темі. Пояснювальна записка оформлена відповідно до вимог, суть розділів розкрито не повністю. Під час захисту роботи студент відповів на більшість додаткових питань.
60-66	Етапи проектування виконувались з систематичним запізненням, проект подано до захисту пізніше встановленого календарним планом терміну. Алгоритм та програмне забезпечення має суттєві недоліки, потребує подальшого вдосконалення, але відповідає темі. Пояснювальна записка оформлена з порушенням вимог, суть розділів розкрито не повністю. Під час захисту роботи студент не відповів на більшість додаткових питань.
34-59	Алгоритм, програмне забезпечення розроблено з грубими помилками та потребують суттєвого доопрацювання. Пояснювальна записка проекту виконана з порушеннями вимог, матеріал викладено в неповному обсязі. (До захисту студент не допускається, проект - доопрацьовується.)

Таблиця 10.1 - Критерії оцінювання

Назва критерії	Максимальна кількість балів
Пояснювальна записка	
Вступ	5
Розділ 1 Призначення і область застосування програмного забезпечення	5
Розділ 2 Огляд існуючого програмного забезпечення. Постановка задачі	5
Розділ 3 Опис і обґрунтування проектних рішень щодо розробки програмного забезпечення	15
Розділ 4 Експериментальне підтвердження вірності програмних та проектних рішень. Реалізація роботи	10
Розділ 5 Інструкція користувача	5
Розділ 6 Основні висновки	5
Додатки	50
ВСЬОГО:	100