

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Системне програмне забезпечення

МЕТОДИЧНІ ВКАЗІВКИ

до виконання курсового проєкту
для студентів базового напрямку “Комп’ютерна інженерія”

*Затверджено
на засіданні кафедри
електронних обчислювальних машин
Протокол № 1 від 16.08.2022 р.*

Системне програмне забезпечення: Методичні вказівки до виконання курсового проєкту для студентів спеціальності “Комп’ютерна інженерія”

Укладач Гаваньо Б.І., д.ф., асистент

Відповідальний за випуск

Рецензент Мельник А.О., доктор техн. наук, професор

ЗМІСТ

ВСТУП	4
1. ТЕМАТИКА КУРСОВИХ ПРОЄКТІВ	5
2. ЗАВДАННЯ НА КУРСОВИЙ ПРОЄКТ	11
3. ПОРЯДОК ВИКОНАННЯ КУРСОВОГО ПРОЄКТУ	12
4. СТРУКТУРА ТА ОБСЯГ КУРСОВОГО ПРОЄКТУ	14
5. ПРАВИЛА ОФОРМЛЕННЯ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ	16
6. КРИТЕРІЇ ОЦІНЮВАННЯ КУРСОВОГО ПРОЄКТУ	19
СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ	22
ДОДАТОК А	23
ДОДАТОК Б	24

ВСТУП

Курсовий проєкт виконується студентами денної та заочної форм навчання відповідно до навчального плану напряму підготовки «Комп'ютерна інженерія» на основі знань і практичних навичок, отриманих ними при вивченні курсу «Системне програмне забезпечення».

Виконання студентами курсового проєкту з дисципліни «Системне програмне забезпечення» дозволяє закріпити теоретичні знання і практичні навички з питань розробки програмного забезпечення, що взаємодіє з операційною системою та апаратним забезпеченням комп'ютера.

Курсовий проєкт – сукупність документів, які розробляють з метою систематизації та закріплення теоретичних знань і практичних навичок, отриманих студентами при вивченні професійно-орієнтованих дисциплін, що входять до навчального плану підготовки бакалаврів зі спеціальності 123 – «Комп'ютерна інженерія». Основним завданням, що ставиться перед студентами при виконанні курсового проєкту з дисципліни «Системне програмне забезпечення», є набуття навичок самостійної розробки програмних виробів та оформлення супровідних програмних документів.

1. ТЕМАТИКА КУРСОВИХ ПРОЄКТІВ

Формування тематики курсових проєктів передбачає два етапи. На першому етапі, що триває протягом першого та другого навчального тижня семестру, студентам надається можливість самостійно вибрати тему курсового проєкту. Сучасна методика підготовки фахівців передбачає наявність у виконуваних ними роботах елементів наукових досліджень, що не завжди можливо досягти у рамках курсового проєкту, який є продуктом інженерної діяльності, однак самостійно вибрана студентом тема курсового проєкту може бути пов'язана з виконуваними ним науковими дослідженнями в будь-якій галузі діяльності зі спеціальності 123 – «Комп'ютерна інженерія». Самостійно вибрана студентом тема курсового проєкту може вирішувати завдання, що охоплюють матеріали інших, у тому числі декількох, навчальних дисциплін напряму освітньої підготовки бакалавра зі спеціальності 123 – «Комп'ютерна інженерія», які пов'язані з розробкою елементів системних програм. Самостійно вибрана студентом тема курсового проєкту повинна в обов'язковому порядку бути узгоджена з викладачем. Студентам, які самостійно не визначилися з вибором теми курсового проєкту на попередньому етапі, надається можливість вибрати її з переліку, запропонованого викладачем. Якщо студент самостійно не зміг визначитись з темою, вона призначається викладачем.

На другому етапі, що триває протягом третього та четвертого навчальних тижнів семестру, студентам потрібно узгодити та затвердити з викладачем технічне завдання згідно обраної теми курсового проєкту.

Перелік основних напрямів для курсового проєкту наведено в таблиці 1.1.

Таблиця 1.1 – Орієнтовний перелік напрямів курсових проєктів

№ з/п	Тема
1.	Збереження та відновлення даних.
2.	Операції над файлами.
3.	Моніторинг використання ресурсів ПК.

4.	Керування підключеними пристроями в операційній системі.
5.	Віддалений доступ.
6.	Віртуалізація операційних систем.
7.	Оптимізація використання пам'яті.
8.	Розробка драйвера пристрою
9.	Кешування для дискової підсистеми
10.	Статистика використання ресурсів
11.	Забезпечення безпеки ОС
12.	Керування процесами
13.	Керування пам'яттю
14.	Керування потоками

Орієнтовний перелік тем курсових проєктів. Цей список не є кінцевим, і може наповнюватись за бажанням викладача (керівника проєкту).

1. Розробка програми для оптимізації роботи операційної системи. Дослідження процесів, що впливають на роботу операційної системи та створення програми, що автоматично виконує операції з підтримки роботи операційної системи.

2. Розробка драйвера для пристроїв вводу-виводу. Розробка драйвера, що дозволяє працювати з новими пристроями вводу-виводу на платформі Windows.

3. Розробка програмного забезпечення для оптимізації роботи мережі. Розробка програми, що автоматично оптимізує роботу мережі, виконуючи діагностику та виправлення проблем з підключенням.

4. Розробка програми для автоматизації процесів віддаленого керування комп'ютером. Розробка програми, що автоматично виконує дії на комп'ютері, на який надано доступ віддалено.

5. Розробка програмного забезпечення для моніторингу системних ресурсів. Розробка програми, що моніторить та контролює ресурси системи для підтримки стабільної роботи операційної системи.

6. Розробка програмного забезпечення для виконання тестів на операційній системі. Розробка програми, що дозволяє проводити тестування

операційної системи з метою виявлення проблем та діагностики помилок.

7. Розробка програмного забезпечення для роботи з файловою системою. Розробка програми, що дозволяє працювати з файловою системою операційної системи.

8. Розробка програмного забезпечення для шифрування файлів та папок. Розробка програми, що дозволяє захистити важливі файли та папки від несанкціонованого доступу з використанням різних алгоритмів шифрування.

9. Розробка програмного забезпечення для відновлення втрачених даних. Розробка програми, що дозволяє відновлювати втрачені дані зі сховища даних операційної системи.

10. Розробка програми для забезпечення безпеки віддаленого доступу до комп'ютера. Розробка програми, що забезпечує безпечний віддалений доступ до комп'ютера за допомогою VPN-підключення.

11. Розробка програмного забезпечення для підтримки забезпечення якості коду. Розробка програми, що дозволяє визначати якість та безпеку коду, написаного для операційної системи.

12. Розробка програмного забезпечення для керування ресурсами комп'ютера. Розробка програми, що дозволяє контролювати та керувати процесами, що виконуються на комп'ютері.

13. Розробка програмного забезпечення для моніторингу подій на комп'ютері. Розробка програми, що моніторить події, що відбуваються на комп'ютері та зберігає історію їх виконання.

14. Розробка програмного забезпечення для оптимізації швидкості роботи комп'ютера. Розробка програми, що дозволяє покращити швидкість роботи комп'ютера за допомогою оптимізації системних налаштувань.

15. Розробка програмного забезпечення для резервного копіювання даних. Розробка програми, що дозволяє створювати резервні копії важливих даних на комп'ютері та зберігати їх на зовнішньому носії.

16. Розробка програмного забезпечення для моніторингу системи в реальному часі. Розробка програми, що дозволяє моніторити систему в

реальному часі, відслідковувати процеси, що виконуються, та виявляти можливі проблеми.

17. Розробка програмного забезпечення для управління службами Windows. Розробка програми, що дозволяє керувати різними службами Windows, зупиняти, запускати та перезапущати їх.

18. Розробка програмного забезпечення для збору та аналізу логів. Розробка програми, що збирає та аналізує лог-файли різних програм та процесів, що виконуються на комп'ютері.

19. Розробка програмного забезпечення для автоматизованого тестування програмного забезпечення. Розробка програми, що дозволяє автоматизувати процес тестування програмного забезпечення на операційній системі.

20. Розробка програмного забезпечення для моніторингу мережі. Розробка програми, що дозволяє моніторити мережу та відслідковувати стан мережевих пристроїв та з'єднань.

21. Розробка програмного забезпечення для керування базами даних. Розробка програми, що дозволяє керувати базами даних та виконувати різні запити до них.

22. Розробка програмного забезпечення для створення віртуальних машин. Розробка програми, що дозволяє створювати та керувати віртуальними машинами на комп'ютері.

23. Розробка програмного забезпечення для керування процесами віртуальної машини. Розробка програми, що дозволяє керувати процесами, що виконуються віртуальною машиною, та виконувати різні дії з їх управлінням.

24. Розробка програмного забезпечення для моніторингу віртуальних машин. Розробка програми, що дозволяє моніторити віртуальні машини та відслідковувати їх стан.

25. Розробка програмного забезпечення для відновлення даних. Розробка програми, що дозволяє відновлювати втрачені дані на комп'ютері.

26. Розробка програмного забезпечення для автоматизованого

резервного копіювання даних. Розробка програми, що дозволяє автоматизувати процес резервного копіювання даних на комп'ютері.

27. Розробка програмного забезпечення для керування відеокартами. Розробка програми, що дозволяє керувати параметрами відеокарт та виконувати різні дії з їх управлінням.

28. Розробка програмного забезпечення для керування звуковими пристроями. Розробка програми, що дозволяє керувати параметрами звукових пристроїв та виконувати різні дії з їх управлінням.

29. Розробка програмного забезпечення для керування дисковими пристроями. Розробка програми, що дозволяє керувати параметрами дискових пристроїв та виконувати різні дії з їх управлінням.

30. Розробка програмного забезпечення для виявлення та усунення помилок в операційній системі. Розробка програми, що дозволяє виявляти та усувати різні помилки, що виникають в операційній системі Windows.

31. Розробка програмного забезпечення для керування файловою системою. Розробка програми, що дозволяє керувати файловою системою на комп'ютері з операційною системою Windows.

32. Розробка програмного забезпечення для автоматизованого тестування операційної системи. Розробка програми, що дозволяє автоматизувати процес тестування операційної системи Windows.

33. Розробка програмного забезпечення для захисту від вірусів та шкідливих програм. Розробка програми, що дозволяє захистити комп'ютер від вірусів та шкідливих програм.

34. Розробка програмного забезпечення для автоматичної настройки операційної системи. Розробка програми, що дозволяє автоматично налаштувати операційну систему Windows з урахуванням потреб користувача.

35. Розробка програмного забезпечення для керування даними в базі даних. Розробка програми, що дозволяє керувати даними в базі даних на комп'ютері з операційною системою Windows.

36. Розробка програмного забезпечення для керування відеофайлами.

Розробка програми, що дозволяє керувати відеофайлами та виконувати різні дії з їх обробкою на комп'ютері з операційною системою Windows.

37. Розробка програмного забезпечення для керування аудіофайлами. Розробка програми, що дозволяє керувати аудіофайлами та виконувати різні дії з їх обробкою на комп'ютері з операційною системою Windows.

38. Розробка програмного забезпечення для керування фотофайлами. Розробка програми, що дозволяє керувати фотофайлами та виконувати різні дії з їх обробкою на комп'ютері з операційною системою Windows.

39. Розробка утиліти для резервного копіювання файлів та даних. Розробка програми, що дозволяє автоматично створювати резервні копії важливих файлів та даних на комп'ютері з операційною системою Windows.

40. Розробка утиліти для очистки системи від непотрібних файлів. Розробка програми, що дозволяє очистити систему від тимчасових та непотрібних файлів на комп'ютері з операційною системою Windows.

41. Розробка утиліти для моніторингу системи. Розробка програми, що дозволяє моніторити систему на комп'ютері з операційною системою Windows та отримувати інформацію про поточний стан системи.

42. Розробка утиліти для очищення реєстру системи. Розробка програми, що дозволяє очистити реєстр системи від застарілих та непотрібних записів на комп'ютері з операційною системою Windows.

2. ЗАВДАННЯ НА КУРСОВИЙ ПРОЄКТ

Відповідно до теми курсового проєкту студент формує вихідні вимоги до проєкту у вигляді технічного завдання, що містить назву теми, вихідні дані, вибір методів, функцій, алгоритмів тощо.

При необхідності в завданні можуть вноситися зміни і доповнення, що повинні бути погоджені зі студентом – автором проєкту і керівником. Студентам, що не з'явилися за затвердженням завдання у встановлений термін, завдання видається тільки з дозволу викладача. Курсовий проєкт повинен бути виконаний і представлений до захисту у термін визначений викладачем.

Керівник курсового проєкту затверджує технічне завдання, проводить систематичні консультації. Керівник рекомендує студентам основну літературу і довідкові матеріали за темою, призначає термін виконання окремих розділів курсового проєкту і періодичних звітів про хід роботи.

У виборі тих або інших рішень ініціатива надається студентові. При цьому керівник може рекомендувати відповідну літературу, статті тощо, ставлячи основну мету – поглиблене самостійне вивчення студентом даного питання.

Студент цілком відповідає за прийняті науково-технічні рішення, правильність виконаних алгоритмів, розрахунків, використаних методів і засобів, якість виконання й оформлення курсового проєкту, а також за своєчасне її завершення.

Ціль виконання курсового проєкту – надбання навичок самостійної розробки системних і інструментальних програм у сучасних операційних середовищах. На захист виносять методи й алгоритми, використані студентом у процесі виконання курсового проєкту до викладеного в пояснювальній записці, а також практичну реалізацію алгоритмів у виді програмного продукту, розробленого студентом самостійно.

3. ПОРЯДОК ВИКОНАННЯ КУРСОВОГО ПРОЄКТУ

Курсовий проєкт починається з затвердження від керівника технічного завдання. У процесі проектування він консультиється з керівником у міру потреби й у зв'язку з виникаючими питаннями.

Рекомендується виконання курсового проєкту рівномірно розподілити протягом всього семестру та розбити на наступні етапи:

Підготовчий етап. Цей етап формує перший розділ пояснювальної записки до курсового проєкту. Студент повинний зрозуміти поставлену перед ним задачу, ознайомитися з рекомендованою літературою. При цьому варто критично підходити до вивчення джерел: рекомендується відбирати найбільш свіжі, останні дані і використовувати найбільш авторитетні джерела. Варто ясно представити мету завдання, що вирішується й уважно проаналізувати вимоги, пред'явлені до її рішення. На даному етапі потрібно зробити постановку завдання, опис вхідних та вихідних даних, сформулювати вимоги до функціонування програмного забезпечення, провести огляд існуючих рішень, аналіз технологій, інструментів, програмних бібліотек, математичних моделей, які будуть покладені в основу роботи програми.

Проектування. Цей етап формує другий розділ пояснювальної записки до курсового проєкту. На цьому етапі студент повинен розглянути та описати різні шляхи рішення поставленої задачі, а також самому розробити алгоритмічне і математичне забезпечення для наступної розробки програми. Має бути описана архітектура програмного забезпечення, в ній виокремлена:

- низькорівнева складова у вигляді бібліотек динамічної/статичної компоновки, які будуть розроблені мовою C/C++;

- рівень представлення – графічний інтерфейс користувача або інтерфейс командного рядка, котрий може бути реалізований будь-якою мовою програмування, якою володіє студент.

На даному етапі студент має представити детальний опис (схему) алгоритмів роботи програми, структур даних.

Реалізація. Цей етап формує третій розділ пояснювальної записки до курсового проєкту. На початку цього етапу студент повинен обрати найбільш раціональне рішення, у ході якого необхідно виконати остаточну розробку і налагодження програмного забезпечення. Варто звернути увагу на повноту, правильність і акуратність ведення документації в ході виконання курсового проєкту, на повноту перевірки правильності роботи програми. Тестування проєкту, розроблення інструкції користувача, відповідної документації – невід’ємні складові проектування програмного забезпечення.

Оформлення. Студент зобов’язаний оформити пояснювальну записку і графічний матеріал згідно з вимогами до оформлення технічної документації, відповідно до діючих стандартів. Ці стандарти описані у розділі 5.

Заключний етап. На цьому етапі проводиться захист курсових робіт. Студент зобов’язаний представити керівникові остаточно оформлену пояснювальну записку до курсового проєкту не пізніше, ніж за 5 днів до захисту. Також, потрібно представити вихідний код програми. Керівник перевіряє проєкт і дає вказівки про виправлення або доповнення, що студентові варто розглянути і внести в проєкт, після чого підписує пояснювальну записку і креслення. На цьому курсовий проєкт вважається закінченим і може бути представленим до захисту.

4. СТРУКТУРА ТА ОБСЯГ КУРСОВОГО ПРОЄКТУ

У пояснювальній записці викладений основний зміст проєкту. У ній подається: обґрунтування актуальності теми, розкривається творчий задум проєкту, представляються розрахунки і пояснення, таблиці і графіки.

Загальними вимогами до розрахунково-пояснювальної записки є: чіткість викладу, логічна послідовність і повна відповідність завданню на курсовий проєкт, переконливість аргументації, конкретність викладу результатів проєкту.

Загальний обсяг пояснювальної записки повинен складати від 35 сторінок тексту з урахуванням рисунків (без додатків).

Обов'язковою складовою курсового проєкту є наявність графічної частини (діаграма класів, алгоритм роботи програми, блок-схеми тощо).

Послідовність розміщення матеріалу в записці курсового проєкту повинна бути такою:

- титульний лист (додаток А);
- завдання на курсовий проєкт;
- анотація;
- зміст;
- вступ (1-2 сторінки);
- розділ 1 (аналітичний: короткий опис, огляд методів, існуючих розробок у даному питанні - до 5 сторінок);
- розділ 2 (проектування: розробка архітектури програми, алгоритмів її роботи, вибір засобів, опис математичної моделі – до 8 сторінок);
- розділ 3 (реалізація: програмування, реалізація проєктованого рішення – від 12 сторінок);
- розділ 4 (тестування/валідація програмного забезпечення, розробка інструкції користувача – кількість сторінок необмежена);
- висновки;
- список використаних джерел;

- додатки (довідкова інформація, результати роботи програми тощо).

У систему ВНС завантажується:

- пояснювальна записка з додатками в форматі PDF;
- вихідний код програми (проект) – в форматі архіву (rar, zip);
- результати роботи програми, якщо програма їх генерує, як окремі об’єкти.

5. ПРАВИЛА ОФОРМЛЕННЯ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

Пояснювальна записка має бути виконана **шрифтом 14 пт**, міжрядковий **інтервал 1.5 пт**. **Поля** – рекомендовані розміри лівого поля – 25 мм, правого - 15 мм, верхнього і нижнього - по 20 мм. На аркушах, де починаються розділи, зміст, анотації, вступ, висновки, список літератури рекомендується збільшувати розмір верхнього поля до 40 мм.

Текст основної частини пояснювальної записки поділяють на розділи, підрозділи, пункти та підпункти. Заголовки структурних частин пояснювальної записки "АНОТАЦІЯ", "ЗМІСТ", "ВСТУП", "РОЗДІЛИ ОСНОВНОЇ ЧАСТИНИ", "ВИСНОВКИ", "СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ", "ДОДАТКИ" **друкують великими літерами симетрично до тексту. Крапку в кінці заголовка не ставлять.** Заголовки підрозділів друкують маленькими літерами (крім першої великої) з абзацного відступу. Розділи повинні бути пронумеровані арабськими цифрами послідовно у всій записці. **Анотація, зміст, вступ, висновки, список використаних джерел не нумеруються.** Після номера розділу ставиться крапка.

Підрозділи нумеруються арабськими цифрами послідовно у всьому розділі. Номер підрозділу повинен містити номер розділу і порядковий номер підрозділу, розділених крапкою. Наприклад: "7.3." – третій підрозділ (параграф) сьомого розділу. Пункти нумеруються арабськими цифрами послідовно у всьому підрозділі. Номер пункту повинен включати у собі номер розділу, підрозділу і пункту, розділених крапками. У кінці номера пункту також ставлять крапку. Наприклад; "7.3.4." – четвертий пункт третього підрозділу сьомого розділу.

Нумерація сторінок записки повинна бути наскрізною: перша сторінка – ТИТУЛЬНИЙ АРКУШ, друга – ЗАВДАННЯ, **третя – АНОТАЦІЯ** і т.д.. Номер сторінки проставляють арабськими цифрами у **правому верхньому куті** (крапку після цифри не ставлять).

Рисунки, формули і таблиці нумеруються двома цифрами – номер_розділу. номер_в_розділі.

Формули в пояснювальній записці (якщо їх більше однієї) **нумерують у межах розділу**. Номер формули складається з номера розділу і порядкового номера формули в розділі, між якими ставлять крапку. Номери формул пишуть біля **правого берега аркуша** на рівні відповідної формули в круглих дужках, наприклад: (3.1) – перша формула третього розділу.

Всі ілюстрації (фотографії, схеми, креслення) у записці повинні називатися однаково – **рисунками**. Рисунки позначаються скорочено: "Рис." Рисунки **нумеруються послідовно у розділі** арабськими цифрами. Номер рисунка повинен містити номер розділу і порядковий номер рисунка, які розділяються крапкою, наприклад: "Рис.1.2." – другий рисунок першого розділу.

Кожна **таблиця** позначається словом "Таблиця" з порядковим номером, що розміщується за словом "Таблиця" з **правого боку**. Таблиця може мати заголовок, який розміщується у наступному рядку після слова "Таблиця". Номер таблиці включає у себе номер розділу і порядковий номер таблиці, що розділені крапкою. Наприклад: "Таблиця 3.2." – друга таблиця третього розділу.

Додатки слід позначати послідовно великими літерами української абетки, за винятком літер Г, Є, З, І, Ї, Й, О, Ч, Ь, наприклад, ДОДАТОК А, ДОДАТОК Б і т. д. Один додаток позначається як ДОДАТОК А.

Бібліографічні описи джерел у переліку наводять згідно з ДСТУ ГОСТ 7.1.

Приклади:

Підручники, монографії, довідники, навчальні посібники:

Мельник А.О. Архітектура комп'ютера: Підручник.- Луцьк: Вид-во обласної друкарні. - 2008.-468с.

Статті:

Мельник А. О. Кіберфізичні системи: проблеми створення та напрями розвитку / Мельник А. О. // Вісник НУ "Львівська політехніка" "Комп'ютерні системи та мережі". – 2014. – № 806. – С. 154–161.

Нормативно-технічні документи:

ДСТУ 3582-97. Інформація та документація. Скорочення слів в українській мові в бібліографічному описі. Загальні вимоги та правила.

Інформаційний лист (datasheet):

Інформаційний лист STM32H743AI [Електронний ресурс] – Режим доступу до ресурсу: <https://www.st.com/resource/en/datasheet/stm32h743ai.pdf>.

Електронні ресурси:

Системне програмне забезпечення [Електронний ресурс] – Режим доступу до ресурсу: <https://vns.lpnu.ua/course/view.php?id=7781>.

6. КРИТЕРІЇ ОЦІНЮВАННЯ КУРСОВОГО ПРОЄКТУ

Проект студента оцінюється за шкалою ECTS (100-бальною шкалою). Найбільш вагомим фактором при оцінюванні курсового проекту є самостійність його виконання студентом, наявність аналізу літератури з опрацюванням матеріалу. Просте копіювання інформації з інформаційних джерел не допускається. Усі розділи курсового проекту повинні містити судження та висновки студента, зроблені ним самостійно. У ході виконання курсового проекту кожні 2 тижні студент звітує викладачу про хід виконання роботи, демонструє виконані пункти розділу. На кожному занятті (консультації) видаються рекомендації щодо оформлення та виконання наступних розділів. У разі, якщо студент без поважних причин пропустив звітні заняття(консультації), його проект може бути оцінена не вище 60 балів (оцінка «задовільно»). Курсові проекти оформлені з порушеннями вимог даних методичних вказівок до захисту не допускаються.

Параметри оцінювання курсового проекту наведено у таблиці 7.1.

При оцінці комісія враховує: якість виконання проекту, зміст повідомлення студента і його відповідей на питання, якість графічних робіт, ступінь самостійності й ініціативи, що виявлена студентом при розробці теми, грамотність викладу матеріалу, володіння науково-технічною термінологією.

Таблиця 7.1 – Розподіл балів для оцінки курсового проекту

№	Оцінюваний параметр	Кількість балів
1.	Оцінювання пояснювальної записки до курсового проекту. Оформлення проекту (відповідність оформлення курсового проекту до вимог нормативних документів, вимог університету та даних методичних вказівок) Теоретична частина (повнота опрацювання теоретичних питань проекту та їх змістове наповнення)	30

	Алгоритмічна та математична складова (відповідність блок-схем стандарту, змістове наповнення та логіка алгоритмів роботи програми, правильність математичних розрахунків)	
2	Програма (стиль написання коду, програма працює у відповідності до завдання, наявність коментарів, чи оптимізований програмний код, чи передбачено захист від помилок користувача)	30
3.	Захист проєкту (повнота та правильність відповідей на запитання членів комісії, стиль викладання матеріалу презентації)	40
Всього		100

Фактична кількість балів за кожним із параметрів курсового проєкту визначається шляхом знаходження відсоткової частки від максимальної кількості балів:

– 100%, якщо студент вміє творчо мислити, узагальнювати вивчений матеріал, використовує для цього теоретичні положення і конкретні параметри, чітко, вичерпно, глибоко розуміє і знає матеріал і вміє їм користуватися в межах вимог програми курсу, вміє вести співбесіду з членами комісії, при відповідях на навідні і додаткові питання самостійно виправляє окремі допущені неточності, якісно й у повному обсязі виконав розрахункові і графічні роботи, що входять у курсовий проєкт.

– 80%, коли за перерахованими показниками є окремі не суттєві недоліки, наприклад, чіткість і повнота звітів не цілкомдостатні, маються дрібні недоліки в розрахунках і конструкторській документації, деяка недбалість оформлення тощо;

- 50%, при наявності істотних недоліків за перерахованими показниками, що студент самостійно зміг виправити;
- 30%, при наявності істотних недоліків за перерахованими показниками, що студент самостійно виправити не міг, навіть якщо окремі відповіді були гарними;
- 0%, якщо студент виявив нерозуміння і незнання основного змісту курсового проєкту, допустив грубі помилки в пояснювальній записці, розрахунках, у креслярській документації, не зміг правильно відповісти на питання комісії.

За рейтинговою оцінкою визначається оцінка за шкалою ECTS та традиційною національною шкалою відповідно до положення університету.

СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Garg, & Verma, G. (2017). Operating Systems: An Introduction. Mercury Learning & Information.
2. Panek. (2020). Windows operating system fundamentals (1st edition). Sybex. – 419 p.
3. David A. Solomon (2017). “Windows Internals”.
4. Габрусєв В.Ю. (2007). Основи операційних систем: ядро, процес, потік. – 96с.
5. Бондаренко М.Ф. Операційні системи: навч. посібник / М.Ф. Бондаренко, О.Г. Качко. – Харків: Компанія СМІТ, 2008. – 432 с.
6. Johnson M. Hart. Windows System Programming, 4th edition / Hart Johnson. – Addison-Wesley, 2010. – 656 p.
7. Thomas W. Doeppner. Operating Systems In Depth: Design and Programming / W. Thomas. – John Wiley & Sons, 2010. – 462 p.

ДОДАТОК А

Приклад заповнення титульного аркуша

Міністерство освіти і науки України

Національний університет

«Львівська політехніка»

Кафедра електронних обчислювальних машин

КУРСОВИЙ ПРОЄКТ

з дисципліни «Системне програмне забезпечення»
на тему: НАЗВА ТЕМИ ЗГІДНО ІЗ ЗАВДАННЯМ

Студента 3-го курсу групи
КІ-
123 «Комп'ютерна інженерія»
Франко І. Я.
Керівник
проф. каф. ЕОМ, канд. техн. наук
Шевченко Т.Г.

Національна шкала: _____

Кількість балів: _____

Оцінка ECTS: _____

Члени комісії: _____
(підпис)

Шевченко Т.Г
(прізвище та ініціали)

(підпис)

(прізвище та ініціали)

(підпис)

(прізвище та ініціали)

ДОДАТОК Б

Приклад заповнення штампів креслень графічної частини

Міністерство освіти і науки України					КУРСОВИЙ ПРОЕКТ				
					Утиліта моніторингу використання віртуальної пам'яті в ОС Windows				
					Модуль відображення даних		Літера	Маса	Масштаб
Зм	Арк.	№ докум.	Підпис	Дата			у		
Виконав	Теслюк				Схема алгоритму				
Керівник	Гаваньо								
Консульт.							Аркуш	Аркушів	1
Консульт.									
Зав. каф.	Мельник						НУ "ЛП", ІКТА, каф. ЕОМ, гр. КІ-34		
Реценз.									
185									

Листи графічної частини як правило виконуються на форматі А2 і додаються в пояснюючу записку в розділ додатків.

В проекті можуть використовуватись такі типи технічних креслень:

- схема алгоритму;
- структурна схема (програмної системи, програми чи модуля);
- діаграма класів (програмної системи, програми чи модуля);
- UML діаграма (можуть бути різними за призначенням);
- структура баз даних;
- схема викликів процедур та функцій (програмної системи, програми чи модуля);
- схема взаємодії (компонент програмної системи, програми чи модуля);
- схема породження, синхронізації та завершення (актуально для процесів та потоків);
- інші типи графічних матеріалів.