

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ**

**МЕТОДИЧНІ ВКАЗІВКИ
ДЛЯ ВИКОНАННЯ КУРСОВОЇ РОБОТИ З ОБЧИСЛЮВАЛЬНОЇ
ТЕХНІКИ ТА ПРОГРАМУВАННЯ**

Донецьк ДонНТУ 2010

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ**

**МЕТОДИЧНІ ВКАЗІВКИ
ДЛЯ ВИКОНАННЯ КУРСОВОЇ РОБОТИ З “ОБЧИСЛЮВАЛЬНОЇ
ТЕХНІКИ ТА ПРОГРАМУВАННЯ”**

(для студентів напрямку підготовки 6.050702 “електромеханіка”)

Розглянуто
на засіданні кафедри ВМ і П
протокол №10 від 05.05.10 р.

Затверджено на засіданні навчально-
видавничої ради ДонНТУ
протокол № від 10г.

Донецьк ДонНТУ 2010

УДК 681.3.06(071)

МЕТОДИЧНИЙ ПОСІБНИК ДЛЯ ВИКОНАННЯ КУРСОВОЇ РОБОТИ З
ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ ТА ПРОГРАМУВАННЯ / Укладачі: І.В.
Диннік, О.А.Тихонова - Донецьк: ДонНТУ, 2010 - 54 с.

Методичний посібник містить завдання , теоретичний матеріал і методичні вказівки для виконання та оформлення курсової роботи з обчислювальної техніки та програмування, призначений для студентів напрямку підготовки 6.050702 “електромеханіка ”.

Автори: І. В. Диннік, ст. викладач

О.А.Тихонова, асистент

Рецензент: І.Я.Зеленьова, доцент

Відповідальний за випуск
зав. кафедрою ОМіП

В.М.Павлиш

ЗМІСТ

1. Структура пояснювальної записки та вимоги до її структурних елементів	5
2. Правила оформлення пояснювальної записки	7
3. Теоретичний матеріал ,необхідний для виконання курсової роботи	9
4. Загальне завдання до курсової роботи	21
5. Варіанти індивідуальних завдань до курсової роботи	21
6. Методичні вказівки до виконання курсової роботи	31
ДОДАТКИ	
Д1. Приклад оформлення титульного аркушу	45
Д2. Приклад оформлення сторінки завдання	46
Д3. Приклад оформлення календарного плану	47
Д4. Приклад оформлення змісту	48
Д5. Приклад оформлення реферату	49
Д6. Приклад оформлення переліку посилань	50
Д7. Приклад оформлення контрольного прорахунку	51
Д8. Зміст файлу результату	53

1. Структура пояснювальної записки та вимоги до її структурних елементів

Курсова робота - це самостійне навчальне дослідження, виконане студентом по одній з тем, запропонованих керівником й оформлене у вигляді наукового звіту. Метою курсової роботи є закріплення, поглиблення й узагальнення знань, отриманих студентом при вивченні курсу "Обчислювальна техніка й програмування"; розвиток інженерних навичок застосування теоретичних положень при рішенні інженерних задач за фахом. Курсова робота є найважливішим елементом самостійної роботи студентів. Основною метою курсової роботи є створення й розвиток навичок дослідницької роботи, уміння працювати з науковою літературою, робити на основі її вивчення висновки й узагальнення.

Перед студентом ставиться конкретна інженерна або дослідницька задача. Студент повинен самостійно розробити всі основні розділи роботи. Курсова робота відрізняється від наукових доповідей й аудиторних виступів студентів тим, що її повинен виконувати кожен студент в письмовому виді, у погодженій з науковим керівником формі й у строго позначений термін. Тим часом, проблематика курсової роботи може бути використана в усному виступі на семінарському або практичному занятті.

Курсова робота не може бути простою компіляцією й складатися із фрагментів різних статей і книг. Вона повинна бути науковим, завершеним матеріалом, мати факти й дані, що розкривають взаємозв'язок між явищами, процесами, аргументами, діями й містити щось нове: узагальнення великої кількості літератури, матеріалів досліджень, у яких з'являється авторське бачення проблеми і її рішення. Цьому загальнотеоретичному положенню підпорядковується структура курсової роботи, її ціль, задачі, методика дослідження й висновки.

На виконання й захист курсової роботи приділяється шістнадцять тижнів. Виконання роботи починається з аналізу завдання й вивчення даних методичних вказівок. У процесі виконання роботи студент може користуватися консультаціями й повинен відповідно до графіка являтися до керівника. Курсова робота виконується за графіком, що доводиться до відома студентів одночасно з видачею курсового завдання.

Не пізніше, ніж за тиждень до зазначеного в графіку строку захисту, студент зобов'язаний представити оформлену роботу керівникові для висновку про відповідність завданню й вимогам до оформлення. Якщо буде потреба доробки, виправлений текст пояснювальної записки до курсової роботи повинен бути зданий керівникові разом з електронною копією не пізніше чим за 2 дні до захисту. Захист курсових робіт проводиться у встановлений термін до початку екзаменаційної сесії й містить у собі усну доповідь студента тривалістю не більше 10 хв і відповіді на питання комісії.

Текст курсової роботи – це виклад відомостей про предмет (об'єкт) дослідження, які є необхідними й достатніми для розкриття сутності означеної роботи (опис теорії, методів роботи) та її результати. Текст курсової роботи викладають, поділяючи матеріал на розділи. Розділи можуть поділятися на пункти

або на підрозділи і пункти. Пункти, якщо це необхідно, поділяють на підпункти. Кожен пункт і підпункт повинен містити закінчену інформацію і висновки. Пояснювальна записка повинна бути розділена на:

- вступну частину;
- основну частину;
- додаток (при необхідності).

Вступна частина містить наступні структурні елементи: титульний аркуш ; сторінку завдання, календарний план; реферат; зміст;

Основна частина повинна містити : вступ; суть проекту (роботи); висновки; перелік посилань. Додатки розміщують після основної частини пояснювальної записки.

Титульний аркуш є першою сторінкою пояснювальної записки і служить основним джерелом бібліографічної інформації, необхідної для обробки і пошуку документів. Титульний аркуш повинен мати відомості, які подають у такій послідовності:

- назва міністерства і навчального закладу;
- повна назва документа;
- прізвище, повні ім'я і по батькові автора;
- підписи відповідальних осіб, включаючи керівника роботи;
- рік виконання курсової роботи;

Приклад оформлення титульного листа наведено у додатку Д1.

Реферат призначений для ознайомлення з роботою. Він повинний бути коротким, інформативним і містити інформацію, що дозволяє представити сутність роботи. Реферат повинний містити: інформацію про обсяг записки, кількість ілюстрацій, таблиць, додатків, кількість джерел по переліку посилань; текст реферату; перелік ключових слів. Текст реферату повинний відображати інформацію, представлену в пояснювальній записці і, як правило, у визначеній послідовності: об'єкт дослідження; ціль роботи; методи дослідження; результати і їхня новизна; значимість роботи і висновки. Реферат необхідно виконувати обсягом не більш 500 слів і розміщати на одній сторінці формату А4. Ключові слова, істотні для розкриття суті записки, формують на основі тексту реферату і розміщують до тексту реферату. Перелік ключових слів включає від 5 до 15 слів (словосполучень), зображених (надрукованих) прописними літерами в називному відмінку в рядок через коми.

Зміст розміщують безпосередньо після реферату, починаючи з нової сторінки. Зміст включає: вступ; послідовно перераховані найменування всіх розділів, підрозділів, пунктів і підпунктів (якщо вони мають); висновки; перелік посилань; найменування додатків.

У **вступі** має бути: ціль роботи; актуальність проблеми; коротко викладена оцінка сучасного стану проблеми; об'єкт дослідження; предмет дослідження; мета дослідження(вивчити і проаналізувати...). Вступ розташовують на окремій сторінці.

Перелік джерел, на які посилаються в записці, повинний бути приведений наприкінці тексту записки, починаючи з нової сторінки. У відповідних місцях записки повинні бути дані посилання. Бібліографічні описи в переліку посилань приводять у порядку, у якому вони вперше згадувалися в тексті. Порядкові номери

описів у переліку є посиланнями в тексті (номерні посилання).

2. Правила оформлення пояснювальної записки

Пояснювальну записку оформлюють на аркушах формату А4 (210x297) мм. Пояснювальну записку виконують за допомогою комп'ютерної техніки на одній стороні аркушу білого папера. Текст записки варто друкувати, дотримуючи наступних розмірів полів: верхнє, ліве і нижнє - не менш 20 мм, праве - не менш 10 мм. Розмір шрифту-14 пт. Міжрядковий інтервал-полуторний.

У записці повинні бути чіткі, не розпливчасті лінії, букви, цифри й інші знаки. Усі букви, цифри і знаки повинні бути виконані чорним чорнилом по всій записці.

Помилки, описки і графічні неточності допускається виправляти чи підчищенням зафарбуванням білою фарбою і нанесенням на тім же чи місці між рядків виправленого зображення машинописним чи способом від руки. Кожен пункт, підпункт і перерахування записуються з абзацного відступу.

Кожен розділ текстового документа варто починати з нової сторінки. Переноси слів у заголовку не допускаються. Розділи і підрозділи повинні мати заголовки. Пункти, як правило, заголовків не мають. Заголовки повинні чітко і коротко відбивати зміст розділів, підрозділів.

Заголовки структурних елементів записки і заголовки розділів варто розташовувати в середині рядка і друкувати прописними буквами без крапки наприкінці, не підкреслюючи. Заголовки підрозділів, пунктів і підпунктів записки варто починати з абзацного відступу і друкувати, крім першої, прописний малими літерами, не підкреслюючи, без крапки наприкінці. Якщо заголовок складається з двох речень, їх розділяють крапкою. Відстань між заголовком і текстом при виконанні записки повинна бути не менш одного рядка.

Текст пояснювальної записки повинний бути коротким, чітким і не допускати різних тлумачень.

Структурні елементи «Реферат», «Вступ», «Висновки», «Перелік посилань» не нумерують, а їхні найменування служать заголовками структурних елементів. Розділи, підрозділи, пункти, підпункти варто нумерувати арабськими цифрами. Розділи записки повинні мати порядкову нумерацію в межах викладу суті, наприклад 1, 2, 3, і т.д. Підрозділи повинні мати порядкову нумерацію в межах кожного розділу. Номер підрозділу складається з номера розділу і порядкового номера підрозділу, розділених крапкою. Після номера підрозділу крапку не ставлять, наприклад 1.1, 1.2 і т.д. Пункти повинні мати порядкову нумерацію в межах кожного розділу чи підрозділу, наприклад, 1.1, 1.2 чи 1.1.1, чи 1.1.2 і т.д.

Нумерація сторінок записки: сторінки пояснювальної записки варто нумерувати арабськими цифрами, дотримуючи наскрізної нумерації по всьому тексту. Номер сторінки проставляють у правому верхньому куті сторінки без крапки наприкінці. «Титульну сторінку», «Завдання» включають у загальну нумерацію сторінок записки. Номери сторінок на титульній сторінці не проставляють. Ілюстрації і таблиці, розташовані на окремих сторінках, включають у загальну нумерацію сторінок записки.

Побудова таблиць: таблиці застосовують для кращої наочності і зручності

представлення показників. Назва таблиці, при її наявності, повинна відбивати її зміст, бути точною, короткою. Назву варто поміщати над таблицею. При переносі частини таблиці на ту ж чи інші сторінки назву поміщають тільки над першою частиною таблиці.

Таблиці, за винятком таблиць додатків, можна нумерувати арабськими цифрами наскрізною нумерацією у межах розділів. В останньому випадку номер таблиці складається з номера розділу і порядкового номера таблиці, відділених крапкою. При цьому перша цифра номер розділу, а друга – порядковий номер таблиці в розділі.

Таблиці кожного додатка позначають окремою нумерацією арабськими цифрами з додаванням перед цифрою позначення додатка. Якщо в документі одна таблиця, вона повинна бути позначена «Таблиця 1» чи «Таблиця В. 1 », якщо вона приведена в додатку В.

На всі таблиці повинні бути приведені посилання в тексті записки. При посиланні варто писати слово «таблиця (табл.)» із указівкою її номера. Наприклад, таблиця 2.1, що означає, що це перша таблиця другого розділу.

Заголовки граф і рядків таблиці варто писати з прописної літери, а підзаголовки граф – з малої літери, якщо вони складають одне речення з заголовком або із прописної літери, якщо вони мають самостійне значення. Наприкінці заголовків і підзаголовків таблиць крапки не ставляться. Заголовки і підзаголовки граф вказують в однині.

Розділяти заголовки і підзаголовки діагональними лініями не допускається. Горизонтальні і вертикальні лінії, що розмежовують рядки таблиці, допускається не проводити, якщо їхня відсутність не утрудняє користування таблицею. Заголовки граф, як правило, записують паралельно рядкам таблиці. При необхідності допускається перпендикулярне розташування заголовків граф.

Заголовок таблиці повинен бути відділений від іншої частини таблиці. Висота рядків таблиці повинна бути не менш 8 мм. Таблицю, у залежності від її розміру, поміщають під текстом, у якому вперше дане посилання на неї, чи на наступній сторінці, а, при необхідності, в додатку до документа.

Якщо рядки чи графи таблиці виходять за формат сторінки, таблицю поділяють на частини, поміщаючи одну частину під іншу, чи поруч, чи переносючи частину таблиці на наступну сторінку. При цьому в кожній частині таблиці повторюють її заголовок і боковик.

При розподілі таблиці на частині допускається її заголовок і боковик замінити відповідно номером граф і рядків. При цьому нумерують арабськими цифрами графи і рядки першої частини таблиці.

Слово «Таблиця» вказують один раз ліворуч над першою частиною таблиці. Над іншими частинами пишуть слова «Продовження таблиці» із указівкою номера. Якщо наприкінці сторінки таблиця переривається і її продовження буде на наступній сторінці, у першій частині таблиці нижню горизонтальну лінію, що обмежує таблицю, не проводять.

Формули і рівняння розташовують безпосередньо після тексту, у якому вони згадуються, в центрі сторінки. Вище і нижче кожної формули чи рівняння повинне бути залишено не менш одного вільного рядка.

Формули і рівняння в записці варто нумерувати порядковою нумерацією в

межах розділу. Номер формули чи рівняння складається з номера формули чи рівняння, розділених крапкою, і номеру розділу -наприклад, формула (5.3) -третя формула п'ятого розділу. Номер формули чи рівняння вказують на рівні формули чи рівняння в дужках у крайньому правому положенні на рядку.

Пояснення значень символів і числових коефіцієнтів, що входять у формулу чи рівняння, варто приводити безпосередньо під формулою в тій послідовності, у якій вони дані в формулі чи рівнянні.

Пояснення кожного символу і числового коефіцієнта варто давати з нового рядка. Першу строку пояснень варто починати з абзацу словом «де» без двокрапки.

Додатка :матеріал, що доповнює текст пояснювальної записки, допускається поміщати в додатках. Додатки повинні мати загальну з іншою частиною наскрізну нумерацію сторінок. У тексті пояснювальної записки на всі додатки повинні бути дані посилання. Додатки розташовують у порядку посилань на них у тексті пояснювальної записки.

Кожен додаток варто починати з нової сторінки з указівкою нагорі в центрі сторінки слова «Додаток» і його позначення. Додаток повинний мати заголовок, що записують симетрично щодо тексту з прописної букви окремим рядком.

3.Теоретичний матеріал, необхідний для виконання курсової роботи

Метою програмування є опис процесів обробки даних (надалі - просто процесів). Під даними розуміється подання фактів й ідей у формалізованому виді, придатному для передачі й переробці в якомусь процесі. Обробка даних - це виконання систематичної послідовності дій з даними. Дані представляються й зберігаються на так званих носіях даних. Сукупність носіїв даних, використовуваних при якій-небудь обробці даних, прийнято називати інформаційним середовищем. Набір даних, що втримуються в якій-небудь момент в інформаційному середовищі можна називати станом цього інформаційного середовища. Процес можна визначити як послідовність змінюючих одне одного станів деякого інформаційного середовища. Описати процес - значить визначити послідовність

станів заданого інформаційного середовища. Формалізований опис процесу називається програмою. Програма складається формалізованою мовою програмування. Перш ніж скласти програму, доводиться проробляти велику підготовчу роботу з уточнення постановки завдання, вибору методу її рішення, з'ясування специфіки застосування програми, проясненню загальної організації розроблювальної програми й багато чого іншого. Використання цієї інформації може істотно спростити завдання розуміння програми, тому досить корисно її якось фіксувати у вигляді окремих документів (часто не формалізованих, розрахованих тільки для сприйняття людиною). Як правило, програми розробляються розраховуючи на те, щоб ними могли користуватися люди, що не беруть участь у їхній розробці (їх називають користувачами). Для освоєння програми користувачем крім її тексту потрібно певна додаткова документація. Програмна документація дозволяє зрозуміти, які функції виконує та або інша програма, як потрібно

підготувати вихідні дані й запустити необхідну програму в процес її виконання, а також інтерпретувати одержувані результати. Крім того, програмна документація допомагає розібратися в самій програмі, що необхідно, наприклад, при її модифікації.

Під технологією програмування розуміється вдосконалювання професійної культури програмування, організації й упорядкування праці програміста, незалежно від конкретної мови програмування й розв'язуваного завдання. Дуже важливе правило: "Програми пишуться для людей, а не для комп'ютера". Яку би складну й витончену програму Ви не написали, у випадку, якщо вона зрозуміла тільки Вам, якщо в ній відсутні коментарі, якщо вона погано оформлена, те ця програма марна.

Програмування з використанням підпрограм – це робота з універсальним методом конструювання складних програм, що отримав назву спадне програмування. Відповідно до цього методу створення програми починається «зверху», тобто з розробки самого головного, генерального алгоритму. На верхньому рівні як правило ще не ясні деталі реалізації тієї або іншої частини програми, тому ці частини пишуться окремо.

Під налагодженням програми розуміється процес, що дозволяє отримати програму, що правильно працює з характеристиками, що вимагаються, у заданій області вхідних даних. Налагодження тісно стикається з тестуванням, передує йому і є по суті мистецтвом виявлення природи помилок й їхньої локалізації. Методології й стратегія налагодження починається не по завершенні написання тексту програми, а безпосередньо в момент написання. Для легкості наступного налагодження рекомендується писати невеликі, добре закоментовані й специфіковані модулі, які виконують вузькі функції.

Під час налагодження треба постаратися визначити природу помилки, чи є це помилка помилкою апаратури, операційної системи (що мало ймовірно), компілятора (читайте інструкції) або власної програми. Якщо помилка в програмі, постарайтеся визначити модуль, у якому вона виникає, виключите з розгляду найменш імовірні джерела помилки, спробуйте звужити область пошуку. Перевірте, чи є помилка повторюваної або стійкою.

Як правило найбільші труднощі з налагодженням зазнають програмісти, що пробують звалити провину на апаратуру, операційну систему, компілятор, і інші

"зовнішні" причини. Будьте скурпульозні, методичні й логічні в пошуку помилок. Застосовуйте систематичний підхід до пошуку помилки. Насамперед, перевіряйте найпростіші припущення, у них легше й швидше знайти помилку або переконатися в її відсутності. Нічого не приймайте на віру. Наприклад, доводи "це дуже простий модуль, у ньому не може бути помилки", "зміни в працюючому модулі були мінімальні", "ще вчора це працювало правильно" і тому подібні не є гарантією відсутності помилки. Дуже часто помилка лежить на поверхні, і ви просто "придивилися" до програми. Свіжий погляд, іноді, дозволяє не тільки виявити помилку, але й підвищити ефективність роботи програми. Ще одним, добре забутих, способом налагодження є включення в програму елементів, які роздруковують значення підозрюваних змінних, видають повідомлення про нормальну роботу того або іншого модуля. Проте, найкращими критеріями налагодження є досвід і терпіння.

Якщо й існує комп'ютерна мова, що визначає суть сучасного програмування,

те, безумовно, це C++. Мова C++ призначена для розробки високопродуктивного програмного забезпечення. Її синтаксис став стандартом для інших професійних мов програмування, а її принципи розробки відображають ідеї розвитку обчислювальної техніки в цілому. C++ послужив фундаментом для розробки мов майбутнього. Наприклад, як Java, так і C# - прямі нащадки мови C++. Сьогодні бути професійним програмістом високого класу означає бути компетентним в C++.

Мова C стала вважатися першою сучасною "мовою програміста", оскільки до її винаходу комп'ютерні мови в основному розроблялися або як навчальні вправи, або як результат діяльності бюрократичних структур. З мовою C усе було інакше. Вона була задумана і розроблена реальними, практикуючими програмістами й відображала їхній підхід до програмування. Її засоби були багаторазово обдумані, відточені й протестовані людьми, які дійсно працювали із цією мовою. У результаті цього процесу з'явилася мова, що сподобалась багатьом програмістам-практикам і швидко поширилась по усім світі. Її небачений успіх обумовило те, що вона була розроблена програмістами для програмістів.

Мова C виникла у результаті революції структурованого програмування 60-х років. До появи структурованого програмування позначилися проблеми в написанні великих програм, оскільки програмісти користувалися логікою, що приводила до створення так названого "спагетті-коду", що складається із безлічі переходів, послідовність яких було важко простежити. У структурованих мовах програмування ця проблема вирішувалася шляхом використання добре знаних інструкцій управління, підпрограм, локальних змінних й інших засобів. Поява структурованих мов дозволила писати вже досить великі програми.

Незважаючи на те що в той час уже існували інші структуровані мови (наприклад Pascal), C став першою мовою, у якій успішно сполучалися міць, добірність і виразність. Її лаконізм, простота синтаксису й філософія були настільки привабливими, що у світі програмування незбагненно швидко утворилася ціла армія її прихильників. З погляду сьогодення цей феномен важко зрозуміти, але, проте, мова C стала свого роду довгоочікуваним "свіжим вітром", що вдихнув у програмування нове життя. У результаті C був загально визнаний як сама популярна структурована мова 1980-х років.

Мова C++ була створена Бьерном Страуструпом (Bjarne Stroustrup) в 1979 році в компанії Bell Laboratories (м. Муррей-Хилл, шт. Нью-Джерсі). Спочатку нова мова одержала ім'я "C с класами" (C with Classes), але в 1983 році вона стала називатися C++. Страуструп побудував C++ на фундаменті мови C, що включає всі її засоби, атрибути й основні достоїнства. Для неї також залишається в силі принцип C, відповідно до якого програміст, а не мова, відповідає за результати роботи своєї програми. Саме цей момент дозволяє зрозуміти, що винахід C++ не було спробою створити нову мову програмування. Це було скоріше вдосконалення вже існуючої (і при цьому досить успішної) мови.

Більшість нововведень, якими Страуструп збагатив мову C, було призначено для підтримки об'єктно-орієнтованого програмування. По суті, C++ став об'єктно-орієнтованою версією мови C. Взявши мову C за основу, Страуструп підготував плавний перехід до ООП(об'єктно-орієнтованого програмування). Тепер, замість того, щоб вивчати зовсім нову мову, C-програмістові досить було освоїти тільки ряд нових засобів, і він міг пожинати плоди використання

об'єктно-орієнтованної технології програмування.

Створюючи C++, Страуструп розумів, наскільки важливо, зберігши споконвічну суть мови C, тобто його ефективність, гнучкість і принципи розробки, внести в нього підтримку об'єктно-орієнтованного програмування. На щастя, ця мета була досягнута. C++ як і раніше надає програмістові свободу дій і владу над комп'ютером (які були властиві мові C), значно розширюючи при цьому його (програміста) можливості за рахунок використання об'єктів.

Незважаючи на те що C++ споконвічно був націлений на підтримку дуже великих програм, цим, звичайно ж, його використання не обмежувалося. І справді, об'єктно-орієнтовані засоби C++ можна ефективно застосовувати практично до будь-якої задачі програмування. Не дивно, що C++ використовується для створення компіляторів, редакторів, комп'ютерних ігор і програм мережевого обслуговування. Оскільки C++ має ефективність мови C, то програмне забезпечення багатьох високоефективних систем побудовано з використанням C++.

Працюючи із програмою, ви повинні виконати наступні дії.

1. Ввести текст програми.
2. Скомпілювати її.
3. Виконати програму.

Перш ніж приступати до виконання цих дій, необхідно визначити два терміни: вихідний код й об'єктний код. Вихідний код - це версія програми, що може читати людина. Вона зберігається в текстовому файлі.

Виконувана версія програми (вона створюється компілятором) називається об'єктним або виконуваним кодом.

Введення тексту програми. При введенні тексту програм повинні бути створені винятково текстові файли, а не файли, у яких разом з текстом (при використанні текстового процесора) зберігається інформація про його форматування. Пам'ятайте, що інформація про форматування перешкодить роботі C++- компілятора. Ім'я файлу, що буде містити вихідний код програми, формально може бути

будь-яким. Але C++- програми звичайно зберігаються у файлах з розширенням .cpp. Тому називайте свої C++- програми будь-якими іменами, але як розширення використовуйте .cpp.

Компілювання програми. Спосіб компіляції програми залежить від використовуваного компілятора й обраних опцій. Більше того, багато компіляторів, наприклад Visual C++ (Microsoft) і C++ Builder (Borland), надають два різних способи компіляції програм: за допомогою компілятора командного рядка й інтегрованого середовища розробки (Integrated Development Environment - IDE). Тому для компілювання C++- програм неможливо дати універсальні інструкції, які підійдуть для всіх компіляторів. Це значить, що ви повинні додержуватися інструкцій, приведених у супровідній документації, прикладеної до вашого компілятора.

Виконання програми. Скомпільована програма готова до виконання. Оскільки результатом роботи C++- компілятора є виконуваний об'єктний код, то для запуску програми як команду досить ввести її ім'я в режимі роботи із запрошення. Наприклад, щоб виконати програму Sample.exe, використовуйте цей командний рядок:

C:\...>Sample.cpp

Якщо ви використовуєте інтегроване середовище розробки, то виконати програму можна шляхом вибору з меню команди Run (Виконати).

Необхідно відзначити, що всі ці програми являють собою консольні додатки, а не додатки, засновані на застосуванні вікон, тобто вони виконуються в сеансі запрошення на ввід команди (Command Prompt). Консольні додатки набагато коротше графічних і краще підходять для навчання програмуванню.

Обробка синтаксичних помилок.

Введіть текст програми, скопіюйте її й виконайте. Кожному програмісту відомо, наскільки легко при вводі тексту програми в комп'ютер вносяться випадкові помилки. На щастя, при спробі скопіювати таку програму компілятор "просигналізує" повідомленням про наявність синтаксичних помилок. Більшість C++-компіляторів спробують "побачити" смисл у вихідному коді програми, незалежно від того, що ви ввели. Тому повідомлення про помилку не завжди відображає правдиву причину проблеми. Тому при одержанні повідомлення про помилку перегляньте дві-три рядки коду, що передують рядку з "виявленою" помилкою. Адже іноді компілятор починає "чути недобре" тільки через кілька рядків після реального місця розташування помилки.

Практично всі C++-компілятори видають як результати своєї роботи не тільки повідомлення про непоправні синтаксичні помилки, але й попередження (warning) різних типів. Якщо компілятор "упевнений" у некоректності програмного коду (наприклад, інструкція не завершується крапкою з комою), він сигналізує про помилку, а якщо в нього є лише "підозра" на некоректність при видимій правильності з погляду синтаксису, то він видає попередження. Тоді програміст сам повинен оцінити, наскільки справедливі підозри компілятора.

Попередження також можна використовувати (за бажанням програміста) для інформування про застосування в коді неефективних конструкцій або застарілих засобів. Компілятори дозволяють вибирати різні опції, які можуть інформувати про речі, що вас цікавлять.

Програми, які ми пишемо, мають два основних аспекти:

набір алгоритмів;

набір даних, якими оперують.

Ці два аспекти залишалися незмінними за всю недовгу історію програмування, зате відносини між ними (парадигма програмування) мінялися. У процедурній парадигмі програмування методом рішення великої задачі є розбивка її на ряд задач поменше. Якщо підзадачи усе ще занадто складні, ми, у свою чергу, розбиваємо їх на ще менші, поки кожна з підзадач не буде вирішена. Таку стратегію називають покроковою деталізацією або принципом "розділяй і пануй". C++ підтримує процедурне програмування.

В C++ передбачені різні типи даних, що дозволяє програмістові вибрати найкращий варіант використання системних ресурсів. Наприклад, обчислення з використанням цілочислової арифметики виконуються набагато швидше, ніж обчислення, вироблені над значеннями із плаваючою крапкою. Таким чином, якщо вам не потрібні значення із дробовою частиною, то вам і не варто даремно витрачати системні ресурси, пов'язані з обробкою таких типів даних, як `float` або `double`. Крім того, для зберігання значень різних типів потрібні різні по розміру

області пам'яті. У C++ є набір убудованих типів даних для представлення цілих і речовинних чисел, символів, а також тип даних “символьний масив”, що служить для зберігання символьних рядків.

Тип `char` служить для зберігання окремих символів і невеликих цілих чисел. Він займає один машинний байт.

Типи `short`, `int` й `long` призначені для представлення цілих чисел. Ці типи розрізняються тільки діапазоном значень, які можуть приймати числа, а конкретні розміри перерахованих типів залежать від реалізації. Як правило, `short` займає половину машинного слова, `int` - одне слово, `long` - одне або два слова.

Типи `float`, `double` й `long double` призначені для чисел із плаваючою крапкою й розрізняються точністю представлення (кількістю значущих розрядів) і діапазоном. Як правило `float` (одинарна точність) займає одне машинне слово, `double` (подвійна точність) - два, а `long double` (розширена точність) - три.

Коли в програмі зустрічається деяке число, наприклад 1, те це число називається літералом, або літеральною константою. Літерал є неадресуємою величиною: хоча реально він зберігається в пам'яті машини, немає ніякого способу довідатися його адресу. Кожен літерал має визначений тип. Літерали цілих типів можна записати в десятковому, восьмеричному й шістнадцятеричному вигляді. От як виглядає число 20, представлене десятковим, восьмеричним і шістнадцятеричним літералами:

```
20// десятковий
024// восьмеричний
0x14 // шістнадцятеричний.
```

Якщо літерал починається з 0, він трактується як восьмеричний, якщо з 0x або 0X, то як шістнадцятеричний. Звичний запис розглядається як десяткове число. За замовчуванням всі цілі літерали мають тип `signed int`.

Слова `true` й `false` є літералами типу `bool`.

Використання специфікатора `const` оголошує об'єкт як константу, значення якої не може бути змінене

```
Наприклад const int bufSize = 512;
```

Спеціальні символи записуються як `escape`-послідовності. Визначено наступні послідовності (вони починаються із символу зворотної косої риси):

```
новий рядок           \n
горизонтальна табуляція \t
вертикальна табуляція  \v
повернення каретки    \r
прогін аркуша         \f
```

Змінна, або об'єкт – це іменована область пам'яті, до якої ми маємо доступ із програми; туди можна поміщати значення й потім витягати їх. Кожна змінна C++ має визначений тип, що характеризує розмір і розташування цієї області пам'яті, діапазон значень, які вона може зберігати, і набір операцій, застосованих до цієї змінної. Оператор визначення змінної виділяє для неї пам'ять. C++ вимагає, щоб об'єкт був відомий до першого звертання до нього.

Ім'я змінної, або ідентифікатор, може складатися з латинських букв, цифр і символу підкреслення. Прописні й малі літери в іменах різняться (мова чутлива до регістра). Мова C++ не обмежує довжину ідентифікатора, однак користуватися

занадто довгими іменами незручно. Деякі слова є ключовими в C++ і не можуть бути використані як ідентифікатори.

Оператор визначення об'єкта складається зі специфікатора типу й імені об'єкта й закінчується крапкою з комою. Наприклад:

```
double wage;
int month;
```

В одному операторі можна визначити кілька об'єктів одного типу. У цьому випадку їх імена перелічуються через кому:

```
double salary, wage;
```

Початкове значення може бути задане безпосередньо в операторі визначення змінної. У C++ припустимі дві форми ініціалізації змінної - явна, з використанням оператора присвоювання:

```
string project = "Fantasia ";
```

і неявна, із завданням початкового значення в дужках:

```
string project( "Fantasia " );
```

Арифметичні операції

Символ операції	Значення	Використання
*	Множення	expr * expr
/	Ділення	expr / expr
%	Залишок від ділення	expr % expr
+	Додавання	expr + expr
-	Віднімання	expr - expr

Операція залишок (%), називана також діленням по модулю, повертає залишок від ділення першого операнда на другий, але застосовується тільки до операндів цілого типу(char, short, int, long). Результат додатний, якщо обидва операнда додатні.

Операції порівняння й логічні операції

Символ операції	Значення	Використання
!	Логічне НІ	!expr
<	Менше	expr1 < expr2
<=	Менше або дорівнює	expr1 <= expr2
>	Більше	expr1 > expr2
>=	Більше або дорівнює	expr1 >= expr2
==	Дорівнює	expr1 == expr2
!=	Не дорівнює	expr1 != expr2
&&	Логічне І	expr1 && expr2
	Логічне АБО	expr1 expr2

Операції порівняння й логічні операції в результаті дають значення типу bool, тобто true або false. Якщо ж таке вираження зустрічається в контексті, що вимагає цілого значення, true перетворюється в 1, а false - в 0.

Операції присвоювання

У мові C++ оператором присвоювання служить одинарний знак рівності (=). Його загальний формат має такий вигляд

```
змінна = вираження;
```


В C++ передбачені спеціальні складені оператори присвоювання, у яких об'єднане присвоювання із ще однією операцією.

Наприклад $x = x + 10$; Використовуючи складений оператор присвоювання, операцію можна переписати в такому виді $x += 10$; Пара операторів $+=$ служить вказівкою компілятору привласнити змінної x суму поточного значення змінної x і числа 10.

Інструкція $x = x - 100$ аналогічна такій: $x -= 100$;

У результаті операції присвоювання об'єкт отримує нове значення, при цьому старе пропадає

Що відбувається, якщо тип об'єкта не збігається з типом значення, що йому хочуть присвоїти? У такому випадку компілятор намагається трансформувати тип об'єкта, що стоїть праворуч, у тип об'єкта, що стоїть ліворуч. Якщо таке перетворення можливо, компілятор неявно змінює тип, причому при втраті точності звичайно видається попередження. Якщо неявне приведення типів неможливо, компілятор сигналізує про помилку:

Операції інкременту ($++$) і декременту ($--$) дають можливість компактного й зручного запису для зміни значення змінної на одиницю.

У рішенні багатьох задач найважливішу роль відіграють функції. Їхнє використання в програмах дає нам наступні переваги: функції спрощують тіло основної програми, завдяки функціям легше розробити, зрозуміти й модифікувати складну програму. Іноді визначена дія складається із групи окремих команд. У цьому випадку набагато зручніше звернутися відразу до всієї групи, а не вписувати всі необхідні команди окремо. Інакше кажучи, всю послідовність інструкцій ми визначаємо як єдину функцію, що описує, як виконується ця послідовність. Аналогічний елемент в інших мовах програмування називають процедурою або підпрограмою. Хоча в термінах C++ такі частини програми називають функцією (function), як і раніше цілком природно застосовувати й вихідний термін - процедура .

Зазвичай для створення функції необхідно зробити наступне:

Розробити програму для рішення поставленої задачі.

Зробити програму як можна коротше.

Забезпечити можливість її багаторазового використання.

Максимально спростити можливість розуміння й модифікації програми.

Крім того, дуже важливо, щоб всі ваші програми й функції були документовані. Забезпечте програму розумним числом коментарів і при необхідності документацією (типу користувальницького посібника).

Визначенням функції (function definition) називається та частина її програмного коду, у якій описані дії, виконувані при виклику функції. У визначення входить заголовок і тіло функції. У загальному випадку визначення функції виглядає в такий спосіб:

заголовок функції

{

тіло функції (фрагмент програмного коду, межі якого позначені фігурними дужками, що містить одну або кілька інструкцій)

}

Заголовок функції містить:

Тип значення, що повертається, (для функцій без значення вказується тип void). Якщо тип значення, що повертається, не зазначений, в C++ це рівносильно завданню типу int. Щоб функція повертала значення, необхідно використовуючи ключове слово return, включити в тіло функції хоча б одну інструкцію, що визначає результат, який повертається.

Ім'я (або ідентифікатор) функції, а також список параметрів й їхніх типів, розташований у круглих дужках (якщо параметрів немає, то простір усередині дужок залишається порожнім).

Тут доречно нагадати наступні правила:

Якщо деякі аргументи мають значення за замовчуванням, вони повинні знаходитися наприкінці списку аргументів.

Не можна викликати функцію раніше, ніж вона визначена!

Засоби зв'язку з функціями забезпечують параметри. Параметри можна використовувати для:

Передачі інформації у функцію.

Виводу інформації з функції (у цьому випадку параметр повинен передаватися по посиланню).

При виклику функції повинна бути забезпечена відповідність типів (type matching) її аргументів і параметрів. Компілятор перевіряє відповідність між типами аргументів і параметрів, а при відсутності такого видає повідомлення про помилку.

C++ дозволяє показати, що функція призначена для роботи з оригіналом об'єкта, а не з його копією. У цьому випадку говорять, що параметр передається не за значенням (value), а по посиланню (reference). Передача по посиланню виконується дуже просто. Відзначте, які параметри ви бажаєте передати по посиланню (передача за значенням відбувається за замовчуванням). В C++ для цього перед ім'ям параметра ставиться символ амперсанда &.

За замовчуванням аргументи передаються у функцію за значенням. Це означає, що при виклику функції створюється копія її аргументу (об'єкта або змінної), що потім використовується функцією. Таким чином, при зміні параметра функції її аргумент не міняється.

Якщо в заголовку функції перед ім'ям змінної поставити символ амперсанда &, то у всіх виконуваних функцією операціях буде використовуватися вихідна змінна, а не її копія.

В інструкцію виклику функції(function call) входить її ім'я й у дужках список її аргументів: ім'я_функції (список_аргументів);

У функції з параметрами в дужках повинні бути перераховані всі її аргументи. Якщо функція отримує кілька аргументів, то відповідність між реальним об'єктом і параметром визначається тільки порядком їхнього проходження.

В C++ й інших мовах програмування використовується поняття області видимості. Область видимості дозволяє вводити різні об'єкти з однаковими іменами за умови, що вони знаходяться у різних областях. Якщо об'єкт визначений усередині функції, то він недоступний поза неї, тобто об'єкт локальний стосовно функції. Якщо ж об'єкт визначений поза всіма функціями, то він доступний для всіх функцій, визначених після нього. Такий об'єкт називається глобальним. Хоча в глобальних об'єктах немає нічого поганого, все-таки краще їх уникати. Основна користь області видимості полягає в тому, що Вам не потрібно пам'ятати про імена

об'єктів всіх своїх функцій. Кожне ім'я діє тільки усередині своєї області видимості.

Щораз, коли ваша програма викликає функцію, комп'ютеру потрібен певний час, щоб спочатку переключитися на функцію, а потім повернутися назад в програму. Головним чином цей час витрачається на те, щоб скопіювати в параметри функції значення, що зберігаються в її аргументах. Відмітною рисою мови C++ є те, що вона допускає особливий тип функцій, код яких розміщується в місці їхнього виклику. Це так називані вбудовані функції (inline functions). Інакше кажучи, там, де у вашій програмі записаний виклик такої функції, компілятор просто робить копію всіх її інструкцій. Таким чином, програмі взагалі не доводиться викликати функцію, оскільки у відповідне місце програми компілятор уписує її повний код. Однак не всі функції можуть бути тими, що вбудовуються. Є кілька обмежень, головне з яких полягає в тому, що усередині таких функцій не можна використовувати цикли. Для позначення вбудованих функцій у їх заголовок включається специфікатор inline.

Робота з файлами в C++ аналогічна роботі з дисплеєм або клавіатурою, але мають місце деякі відмінності:

- Файл ідентифікується по своєму імені на диску.
- Можна використовувати той самий файл як для запису інформації (вивід з комп'ютера), так і для зчитування інформації (ввієн в комп'ютер).

Можна обробляти два види файлів:

- Об Текстові файли.
- Об Бінарні файли.

Текстові файли організовані в рядки, кожен з яких позначений символом кінця рядка (endl). При цьому всі дані зберігаються в зручному вигляді. Поля даних розділені пробілами, тому вміст файлу можна легко зрозуміти. Навпроти, бінарні файли не організовані в рядки. Одна частина інформації безупинно переходить в іншу, а числа зберігаються так само, як й у пам'яті комп'ютера, у бінарному форматі, а не у вигляді звичних нам десяткових цифр. Тому прочитати роздрукований бінарний файл зовсім не просто.

Щоб можна було використовувати файл, його, як і будь-який інший об'єкт, необхідно оголосити:

fstream ідентифікатор;

Ідентифікатор — це ім'я, що використовується в програмі для звертання до файлу, а не ім'я файлу на диску! Відповідність між ідентифікатором й іменем файлу на диску визначається функцією open().

Після оголошення файлу операції над ним можна виконувати за допомогою функцій класуfstream. Найбільш важливими функціями є: open (), close (), eof().

Функція open () встановлює відповідність між файловим об'єктом і даними на диску. Вона визначає також режим доступу до файлу: для введення, для виводу, для додавання й т.д. Синтаксис функції open ():

open(char ім'я_файлу[], int режим^доступу);

Функція отримує два аргументи: рядок символів (ім'я_файлу), що визначає ім'я файлу на диску, і прапор (режим_доступу), що визначає режим доступу до

файлу. Ім'я файлу є рядком, що кінчається нулем. У рядку можна задати повний шлях до файлу, включаючи диск, каталог і т.д. При цьому зворотна коса риса \ у повному шляху до файлу в аргументі функції замінюється подвійною зворотною косою рисою \\.

Режим доступу може бути одним з наступних:

Прапорець `ios::out` відкриває файл для виводу (запис у файл).

Прапорець `ios::in` відкриває файл вводу (зчитування з файлу).

Прапорець `ios::app` відкриває файл для додавання (запис у кінець попередньої інформації).

Прапорець `ios::nocreate` відкриває файл тільки в тому випадку, якщо він існує.

Наприклад: `fstream myf i le;`

Відкриття файлу `ist.txt`, що знаходиться на диску `a` в каталозі `\franca`, для вводу: `myf i le.open("a:\\franca\\ist.txt", ios:: in);`

Відкриття файлу, ім'я якого написано в символьному масиві `"ім'я_файлу"`, для виводу: `myf i le.open(ім'я_файлу, ios::out);`

Прапори режиму доступу можна комбінувати, використовуючи логічний оператор диз'юнкції (`|`). Наприклад:

`ваш_файл.open("roster.txt", ios::nocreate|ios:: in);`

У цьому випадку система спробує відкрити для вводу (прапор `ios::in`) файл `roster.txt`. Якщо цей файл у поточному каталозі не знайдений, новий файл створений не буде й відкриття природно не відбудеться.

Завжди корисно перевірити, наскільки правильно відкрився файл. Наприклад, файл може не відкритися, якщо його немає в указаному місці. У компіляторах компаній Microsoft й Borland така перевірка виконується по-різному.

При перевірці правильності відкриття файлу в компіляторах Microsoft викликається функція `is_open()`. Якщо функція повертає нуль, виходить, файл не відкрився. Використовується функція `is_open()` у такий спосіб:

`ваш_файл.open("го81ег", ios:: in| ios::nocreate);`

`ваш.файл.is_open()==0) cout<<"Oиun6Ka відкриття файлу";`

При перевірці правильності відкриття файлу в компіляторах Borland необхідно порівняти об'єкт класу `fstream` з константою `NULL`. Якщо результатом порівняння є істина, виходить, файл не відкрився. Наприклад:

`ваш_файл.open("roster", ios::in| ios::nocreate);`

`if(Ваш_|>alta==NULL) сої1<<"0швидка відкриття файлу";`

Функція `close()` забезпечує правильне закриття файлу й збереження даних на диску. Іноді дані потрапляють на диск не відразу, і при неправильному закритті файлу можуть бути загублені. Ця функція без параметрів.

Функція `eof()` повертає 1, якщо в процесі зчитування даних ви переходите за кордон файлу, що відзначається символом кінця файлу (символ `end-of-file` або просто `eof`). Зручність функції `eof()` визначається тим, що в більшості випадків об'єм даних у файлі заздалегідь не відомий.

4. Загальне завдання до курсової роботи

Розробити алгоритм і скласти програму рішення поставленої задачі інженерно-технічного напрямку. Розрахунок окремих величин (на вибір розроблювача) оформити у вигляді функції. Всі вихідні дані брати із файлу, що попередньо був створений розроблювачем. Результати помістити в інший файл. Перевірити працездатність програми та побудувати необхідні графіки засобами електронної таблиці. Розробити документацію до всіх структурних частин програми, у якій необхідно вказати призначення, вимоги до даних, що вводяться, формат інформації, що виводиться й дати опис розробленого алгоритму. Зробити інженерний аналіз отриманих результатів. Оформити пояснювальну записку засобами текстового редактора.

5. Варіанти індивідуальних завдань до курсової роботи

5.1. Тема 1. Розрахунок опору фази пускового реостата, необхідного для одержання припустимого пускового струму

В процесі експлуатації електричного обладнання часто постає питання економічної доцільності обслуговування і строку експлуатації систем приводу. При цьому треба намагатися уникати важких для обладнання режимів, тим самим подовжуючи його строк служби. Одним з проблемних режимів для двигуна є його запуск, при якому спостерігається значне перевищення струму машини. При використанні асинхронного двигуна з фазним ротором для полегшення пуску використовується пусковий реостат, що вмикається в коло ротора.

Додавання пускового опору підвищує пусковий момент та знижує перевантаження на машину. Величина опору підбирається для певної системи виходячи з потреб щодо цієї конкретної системи.

Дійсний опір фази пускового реостата обчислюється в такий спосіб

$$R_p = \frac{R'_p}{K_i \cdot K_E}$$

При цьому струм статора при пуску двигуна з реостатом

Дійсний струм ротора при пуску двигуна з реостатом

Коефіцієнт потужності при реостатному пуску двигуна

Номинальний лінійний струм статора

$$I_{1H} = \frac{P_H}{\sqrt{3} \cdot U_H \cdot \eta \cdot \cos\varphi_H}$$

Синхронна кутова швидкість обертання магнітного поля

$$\omega_1 = \frac{2\pi \cdot n_1}{60}$$

Перехід від в.о. до Ом здійснюється в такий спосіб:

$$U_\phi = \frac{U_H}{\sqrt{3}}$$

$$R_1 = R_1^* \cdot \frac{U_\phi}{I_{1H}}$$

$$X_1 = X_1^* \cdot \frac{U_\phi}{I_{1H}}$$

$$R_2' = R_2^* \cdot \frac{U_\phi}{I_{1H}}$$

$$X_2' = X_2^* \cdot \frac{U_\phi}{I_{1H}}$$

Номінальний електромагнітний обертаючий момент двигуна

$$M_H = \frac{m_1 U_\phi^2 R_2'}{\omega_1 s_H \left[\left(R_1 + \frac{R_2'}{s_H} \right)^2 + (X_1 + X_2')^2 \right]}$$

Струм статора при пуску двигуна із замкнутим накоротко ротором без урахування потоку, що намагнічує

$$I_{1HE} = \frac{U_\phi}{\sqrt{(R_1 + R_2')^2 + (X_1 + X_2')^2}}$$

ЕДС обмотки статора

Коефіцієнти трансформації по ЕДС і струму системи статор-ротор

Обертаючий момент при пуску двигуна

Вихідними даними до завдання є наступні величини:

P_H – Номінальна потужність двигуна (квт);

U_H – Номінальна напруга (В);

n_1 – Номінальні синхронні оберти машини (об/хв);

η - Номінальний ККД машини;

$\cos\varphi_H$ – Номінальний коефіцієнт потужності;

E_{2H} – Номінальна вторинна ЕДС (В);

I_{2H} – Номінальний струм ротора (А);

S_H – Номінальне ковзання;

X_m^* – Індуктивний опір намагнічування машини (в. о.);

R_1^* – Активний опір статорної обмотки двигуна (в. о.);

x_1^* – Індуктивний опір статорної обмотки двигуна (в. о.);

$R_2'^*$ – Наведений активний опір ротора (в. о.);

$x_2'^*$ – Наведений індуктивний опір ротора (в. о.).

Значення всіх вихідних величин наведені в таблиці

Варіант	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	1.10
P_H	*	150	70	180	*	90	50	115	*	90
U_H	380	600	380	600	380	600	380	600	380	600
n_1	1500	*	3000	750	500	*	750	3000	1500	4500
η	0,85	0,83	*	0,89	0,92	0,94	*	0,87	0,86	0,85
$\cos\phi_H$	0,7	0,85	0,75	0,9	0,73	0,87	0,89	0,78	0,84	0,86
E_{2H}	160	220	170	270	190	300	180	260	185	280
I_{2H}	150	180	195	165	170	175	185	155	200	160
S_H	0,001	0,005	0,003	0,002	0,004	0,005	0,003	0,002	0,001	0,004
m_1	3	3	3	3	3	3	3	3	3	3
R_1^*	0,021	0,032	0,022	*	0,025	0,031	0,028	*	0,03	*
x_1^*	0,007	0,008	0,009	0,1	0,12	0,11	0,115	0,008	0,007	0,12
$R_2'^*$	0,034	0,043	0,036	0,041	0,038	0,042	0,037	0,035	0,04	0,043
$x_2'^*$	0,07	0,09	0,13	0,11	0,08	0,1	0,085	0,095	0,012	0,097

Варіанти завдань до теми 1

Для всіх варіантів значення опору пускового реостата варіювати в межах $R_p' = 0 \div 25 \cdot R_1$ (R_1 є розрахунковою величиною). Підібрати за графіком $I_{1П} = f(R_p')$ такі значення опору пускового реостата, щоб пусковий струм не перевищував дворазового номінального значення ($2 \cdot I_{1Н}$). Указати отримані при цьому момент двигуна M_n , струм ротора $I_{1П}$ і коефіцієнт потужності $\cos\phi_n$.

1.1. Проаналізувати вплив величини потужності двигуна P_n на величину пускового струму $I_{1П}$, потужність двигуна варіювати в межах $70 \div 120$ кВт із кроком 15 кВт.

1.2. Проаналізувати вплив номінальних синхронних обертів n на величину пускового струму $I_{1П}$, оберти приймати рівними: 500, 750, 1000, 1500, 3000 об/хв.

1.3. Проаналізувати вплив величини ККД двигуна η на величину пускового струму $I_{1П}$, ККД варіювати в межах $0,8 \div 0,9$ із кроком 0,02.

1.4. Проаналізувати вплив величини опору статорної обмотки двигуна R_{1*} на величину пускового струму $I_{1П}$, опір варіювати в межах $0,02 \div 0,03$ в.о. із кроком 0,0025 в.о.

1.5. Проаналізувати вплив величини потужності двигуна P_n на величину пускового струму $I_{1П}$, потужність двигуна приймати рівною: 80, 90, 105, 110, 115 кВт.

1.6. Проаналізувати вплив номінальних синхронних обертів n на величину пускового струму $I_{1П}$, оберти варіювати в межах $750 \div 4500$ об/хв із кроком 750 об/хв.

1.7. Проаналізувати вплив величини ККД двигуна η на величину пускового струму $I_{1П}$, ККД приймати рівним 0,7, 0,75, 0,8, 0,85, 0,87.

1.8. Проаналізувати вплив величини опору статорної обмотки двигуна R_{1*} на величину пускового струму $I_{1П}$, опір приймати рівним 0,15, 0,2, 0,22, 0,3, 0,35 в. о.

1.9. Проаналізувати вплив величини потужності двигуна P_n на величину пускового струму $I_{1П}$, потужність двигуна варіювати в межах $100 \div 200$ кВт із кроком 25 кВт.

1.10. Проаналізувати вплив величини опору статорної обмотки двигуна R_{1*} на величину пускового струму $I_{1П}$, опір варіювати в межах $0,015 \div 0,035$ в.о. із кроком 0,004 в.о.

Тема 2. Аналіз системи привода

Амплітудно-частотна характеристика (АЧХ) електромеханічної системи приводу виконавчих органів має вигляд:

$$\lambda = \sqrt{\frac{A^2(\omega) + B^2(\omega)}{M^2(\omega) + N^2(\omega)}}$$

де $A(\omega) = \frac{1}{I_{30}I_{34}}; B(\omega) = \frac{\beta_{34}\omega}{I_{30}};$

$$M(\omega) = I_3 I_4 \omega^4 - \left(\frac{I_3 + I_4}{I_{34}} + \beta_d \beta_{34} + \frac{I_4}{I_{30}} \right) \omega^2 + \frac{1}{I_{30}I_{34}};$$

$$N(\omega) = -[\beta_{34}(I_3 + I_4) + \beta_d I_4] \omega^3 + \left(\frac{\beta_d}{I_{34}} + \frac{\beta_{34}}{I_{30}} \right) \omega;$$

I_3 - момент інерції ротора електродвигуна, кг/м²;

I_4 - наведений до ротора електродвигуна момент інерції виконавчого органа, кг/м²;

I_{34}, I_{30} - наведені до ротора електродвигуна коефіцієнти крутильної піддатливості, відповідно, моделі електродвигуна й трансмісії, Н⁻¹м⁻¹;

β_d, β_{34} - наведені до ротора електродвигуна коефіцієнти лінійного демпфірування, відповідно, моделі електродвигуна й трансмісії.

$$\beta_{34} = \frac{e^{\delta} - e^{-\delta}}{2\pi I_{34} \omega};$$

δ - логарифмічний декремент коливань;

$$\beta_d = S_k \omega_c I_3$$

S_k - критичне ковзання двигуна

$\omega_c = 314 \text{ с}^{-1}$ - кругова частота мережі

ω - частота коливань, Гц

У зарезонансній зоні значення АЧХ монотонно убувають, причому практичний інтерес представляють значення $\lambda \geq 0.05$. Для всіх варіантів

Вихідними даними до завдання є наступні величини:

I_4 - наведений до ротора електродвигуна момент інерції виконавчого органа, кг/м²;

I_3 - момент інерції ротора електродвигуна, кг/м²;

I_{34} , I_{30} - наведені до ротора електродвигуна коефіцієнти крутильної піддатливості, відповідно, моделі електродвигуна й трансмісії, $\text{H}^{-1}\text{m}^{-1}$;
 δ - логарифмічний декремент коливань;
 S_k - критичне ковзання двигуна.

Значення всіх вихідних величин наведені в таблиці

Варіант	2.1	2.2	2.3	2.4	2.5	2.6	2.7	2.8	2.9	2.10
I4	0.18	*	0.15	0.19	0.175	0.16	0.165	0.17	*	0.185
I3	*	1.8	1.65	1.7	1.75	1.85	1.9	*	1.6	1.95
I34	6	5	*	7	6	5	6	5	4	5
I30	1.6	1.5	1.8	1.7	1.5	*	1.8	1.9	1.8	*
Sk	0.14	0.13	0.165	0.155	*	0.16	0.135	0.145	0.16	0.15
Δ	0.45	0.5	0.7	*	0.65	0.55	*	0.6	0.55	0.75

Варіанти завдань до теми 2

2.1. Проаналізувати вплив величини моменту інерції I_3 на вид АЧХ системи привода виконавчого органа очисного комбайна при наступних параметрах: момент інерції ротора електродвигуна варіювати в межах $1 \leq I_3 \leq 2$ кг/м² із кроком 0.23 кг/м², частоту коливань варіювати в межах $20 \leq \omega \leq 60$ Гц.

2.2. Проаналізувати вплив величини наведеного до ротора електродвигуна моменту інерції виконавчого органа I_4 на вид АЧХ системи привода при наступних параметрах: наведений до ротора електродвигуна момент інерції виконавчого органа варіювати в межах $0.15 \leq I_4 \leq 1.95$ кг/м² із кроком 0.2 кг/м², частоту коливань варіювати в межах $20 \leq \omega \leq 60$ Гц.

2.3. Побудувати АЧХ системи привода виконавчого органа при зміні коефіцієнта крутильної піддатливості електродвигуна I_{34} діапазоні від 5.5 до 10 Н-1м-1 із кроком 1 Н-1м-1, частоту коливань варіювати в межах $20 \leq \omega \leq 60$ Гц.

2.4. Проаналізувати вплив на вид АЧХ досліджуваної системи величини декременту коливань δ при наступних параметрах: величину декременту коливань δ змінювати від 0.4 до 0.75 із кроком 0.15, частоту коливань варіювати в межах $20 \leq \omega \leq 60$ Гц.

2.5. Досліджувати вплив на вид АЧХ системи привода величини критичного ковзання електродвигуна Sk при наступних параметрах: критичне ковзання двигуна варіювати в межах $0.05 \leq Sk \leq 0.15$ с кроком 0.03, частоту коливань варіювати в межах $20 \leq \omega \leq 60$ Гц.

2.6. Досліджувати вплив на вид АЧХ системи привода виконавчого органа значення коефіцієнта крутильної піддатливості трансмісії I_{30} при наступних

параметрах: коефіцієнт крутильної піддатливості трансмісії I_{30} варіювати в межах $1 \leq I_{30} \leq 3$ із кроком 0.55, частоту коливань варіювати в межах $20 \leq \omega \leq 60$ Гц.

2.7. Проаналізувати вплив величини декременту коливань δ на вид АЧХ досліджуваної системи при наступних значеннях δ : 0.5; 0.62; 0.74; 0.85; 0.9, частоту коливань варіювати в межах $20 \leq \omega \leq 60$ Гц.

2.8. Досліджувати вплив величини моменту інерції I_3 на вид АЧХ системи привода виконавчого органа очисного комбайна при наступних параметрах:

момент інерції ротора електродвигуна I_3 приймати рівним : 0,8; 0,95; 1,1; 1,65; 1,9 кг/м², частоту коливань варіювати в межах $20 \leq \omega \leq 60$ Гц.

2.9. Досліджувати вплив величини наведеного до ротора електродвигуна моменту інерції виконавчого органа I_4 на вид АЧХ системи привода при наступних параметрах:

наведений до ротора електродвигуна момент інерції виконавчого органа приймати рівним : 0.5; 0.53; 0.62; 0.67; 0.73 кг/м², частоту коливань варіювати в межах $20 \leq \omega \leq 60$ Гц.

2.10. Проаналізувати вплив коефіцієнта піддатливості I_{30} на вид АЧХ системи привода виконавчого органа при наступних параметрах: коефіцієнт крутильної піддатливості трансмісії I_{30} приймати рівним : 1.5; 1.8; 2.3; 3.5; 4.2, частоту коливань варіювати в межах $20 \leq \omega \leq 60$ Гц.

Тема 3. Розрахунок статичних характеристик двигуна і їхніх характерних точок.

Статичними характеристиками двигуна є графіки функцій $M=f(S)$ і $I_2=f(S)$. Залежність між обертальним моментом двигуна й ковзанням називається механічною характеристикою, а між струмом ротора й ковзанням - швидкісною.

Статичні характеристики визначають поведінку машини при навантаженні. Механічними характеристиками, як правило, зручно описувати перехід машин з одного режиму роботи в інший, і вид їх залежить від параметрів режиму - напруги живлення, опору в роторному ланцюзі, частоти живлячої напруги й інших. При розрахунку характеристик у межах одного режиму досить, змінюючи величину ковзання, визначати момент двигуна й струм ротора. Слід зазначити, що при побудові характеристик виділяють три характерні точки

- точка ідеального холостого ходу, коли машина не навантажена й ковзання немає;
- точка короткого замикання, що характеризується тим, що вал машини стоїть на місці;
- номінальна точка - режим, описаний паспортними даними двигуна.

Штучну реостатну механічну характеристику АД при включенні в ланцюг ротора опору розраховують по рівнянню

$$M = \frac{m_1 \cdot U_{\phi}^2 \cdot (R'_2 + R'_p)}{\omega_1 S \left[\left(R_1 + \frac{R'_2 + R'_p}{S} \right)^2 + (X_1 + X'_2)^2 \right]}$$

Криву залежності струму ротора від ковзання при пуску двигуна розраховують по рівнянню

$$I_2 = \frac{K_n \cdot U_{\phi}^2}{E_{2n} \sqrt{\left(R_1 + \frac{R'_2 + R'_p}{S} \right)^2 + (x_1 + x'_2)^2}}$$

Спрощений запис механічної характеристики представляється у вигляді формули Клосса

$$M = \frac{(2 + a \cdot S_{\text{кр}}) \cdot M_{\text{max}}}{\left(\frac{S_{\text{кр}}}{S} \right) + \left(\frac{S}{S_{\text{кр}}} \right) + a \cdot S_{\text{кр}}}$$

Критичне ковзання і параметр а можна визначити в такий спосіб:

$$s_{\text{кр}} = \frac{R'_2 + R'_p}{\sqrt{R_1^2 + (X_1 + X'_2)^2}} \quad a = 2 \frac{R_1}{R'_2 + R'_p}$$

Номінальний лінійний струм статора

$$I_{1H} = \frac{P_H}{\sqrt{3} \cdot U_H \cdot \eta \cdot \cos\varphi_H}$$

Частота обертання ротора при номінальному навантаженні

$$n_H = n_1 \cdot (1 - s_H)$$

Синхронна кутова швидкість обертання магнітного поля

$$\omega_1 = \frac{2\pi \cdot n_1}{60}$$

Перехід від в.о. до Ом здійснюється в такий спосіб

$$U_{\text{ФН}} = \frac{U_H}{\sqrt{3}}$$

$$R_1 = R_1^* \cdot \frac{U_{\text{ФН}}}{I_{1H}}$$

Максимальний електромагнітний обертаючий момент двигуна

Характерні точки - це:

1) Точка ідеального холостого ходу

$$n = n_1, \quad S = 0, \quad M = 0, \quad I_2 = 0$$

2) Точка номінального режиму

$$n = n_n, \quad S = S_n, \quad M = M_n$$

3) Точка пуску двигуна

$$n = 0, \quad S = 1, \quad M = M_{II}$$

Вихідними даними до завдання є наступні величини:

P_H – Номінальна потужність двигуна (кВт);

U_H – Номінальна напруга (В);

U - Напруга живлення (В);

n_1 – Номінальні синхронні оберти машини (об/хв);

η - Номінальний ККД машини;

$\cos\varphi_H$ – Номінальний коефіцієнт потужності;

E_{2H} – Номінальна вторинна ЕДС (В);

S_H – Номінальне ковзання;

R_1^* – Активний опір статорної обмотки двигуна (в. о.);

x_1^* – Індуктивний опір статорної обмотки двигуна (в. о.);

$R_2'^*$ – Наведений активний опір ротора (в. о.);

$x_2'^*$ – Наведений індуктивний опір ротора (в. о.);

R_p' – Опір пускового реостата.

Значення всіх вихідних величин наведені в таблиці

Варіант	3.1	3.2	3.3	3.4	3.5	3.6	3.7	3.8	3.9	3.10
R_1^*	0.028	0.018	0.032	0.019	0.031	0.03	0.029	0.027	0.026	0.025
x_1^*	0.091	0.075	0.084	0.093	0.11	0.12	0.115	0.095	0.097	0.117
$R_2'^*$	0.039	0.025	0.043	0.026	0.042	0.028	0.041	0.027	0.04	0.029
$x_2'^*$	0.11	0.07	0.13	0.08	0.12	0.085	0.095	0.12	0.115	0.09
R_p'	*	0	0	0	*	0	0	0	*	0
P_H	75	*	90	120	100	*	95	110	80	*
U_H	380	600	380	380	380	600	380	380	380	600
n_1	1500	750	*	500	3000	1000	*	750	1500	500
U	380	615	385	*	390	595	375	*	385	610
S_H	0,045	0,05	0,01	0,02	0,012	0,015	0,043	0,041	0,047	0,046
m_1	3	3	3	3	3	3	3	3	3	3
E_{2H}	180	300	160	270	190	290	200	285	185	265
K_n	0,97	0,95	0,96	0,97	0,95	0,96	0,97	0,95	0,97	0,96
H	0,85	0,83	0,86	0,89	0,92	0,94	0,91	0,87	0,86	0,85
$\cos\varphi_H$	0,7	0,85	0,75	0,9	0,73	0,87	0,89	0,78	0,84	0,86

Варіанти завдань до теми 3

Для всіх варіантів значення ковзання S варіювати в межах $0 < S < 1$. Для знаходження характерних точок у ряд значень S потрібно включити значення $S = S_n$, $S = 1$. При побудові графіків функцій $M=f(S)$ і $I_2=f(S)$ урахувати точку ідеального холостого ходу $S=0$.

3.1. Проаналізувати вплив величини опору пускового реостата R_p' на вид статичних характеристик двигуна $M=f(S)$ і $I_2=f(S)$, значення опору пускового реостата варіювати в межах $0 \div 5 \cdot R_1$, ковзання S приймати рівним: 0.01; 0.05; 0.15; 0.25; 0.35; 0.45; 0.7; 0.9.

3.2. Проаналізувати вплив величини номінальної потужності двигуна P_n на вид статичних характеристик двигуна $M=f(S)$ і $I_2=f(S)$, значення номінальної потужності двигуна варіювати в межах $50 \div 100$ кВт із кроком 15кВт, ковзання S приймати рівним: 0.005; 0.06; 0.17; 0.21; 0.35; 0.45; 0.65; 0.9.

3.3. Проаналізувати вплив величини синхронних обертів машини n_1 на вид статичних характеристик двигуна $M=f(S)$ і $I_2=f(S)$, значення величини синхронних обертів приймати рівним: 500,750,1000,1500,3000,4500 об/хв; ковзання S приймати рівним: 0.06; 0.1; 0.15; 0.25; 0.45; 0.55; 0.75; 0.85.

3.4. Проаналізувати вплив величини напруги живлення U на вид статичних характеристик двигуна $M=f(S)$ і $I_2=f(S)$, значення напруги живлення варіювати в межах $100 \div 380$ В с кроком 50 В, ковзання S приймати рівним: 0.02; 0.07; 0.19; 0.28; 0.34; 0.65; 0.85; 0.9.

3.5. Проаналізувати вплив величини опору пускового реостата R_p' на вид статичних характеристик двигуна $M=f(S)$ і $I_2=f(S)$, значення опору пускового реостата варіювати в межах $0 \div 4 \cdot R_1$, ковзання S приймати рівним: 0.03; 0.06; 0.17; 0.21; 0.38; 0.45; 0.65; 0.85.

3.6. Проаналізувати вплив величини номінальної потужності двигуна P_n на вид статичних характеристик двигуна $M=f(S)$ і $I_2=f(S)$, значення номінальної потужності двигуна приймати рівним: 50, 60, 70, 80, 100 кВт, ковзання S приймати рівним: 0.01; 0.07; 0.19; 0.23; 0.37; 0.49; 0.75; 0.85.

3.7. Проаналізувати вплив величини синхронних обертів машини n_1 на вид статичних характеристик двигуна $M=f(S)$ і $I_2=f(S)$, значення синхронних обертів машини варіювати в межах $500 \div 3000$ об/хв, ковзання S приймати рівним: 0.02; 0.09; 0.18; 0.24; 0.38; 0.51; 0.7; 0.9.

3.8. Проаналізувати вплив величини напруги живлення U на вид статичних характеристик двигуна $M=f(S)$ і $I_2=f(S)$, значення напруги живлення приймати рівним: 200, 250, 270, 300, 380 В, ковзання S приймати рівним: 0.01; 0.05; 0.15; 0.25;

0.35; 0.45; 0.7; 0.9.

3.9. Проаналізувати вплив **величини опору** пускового реостата R_p' на вид статичних характеристик двигуна $M=f(S)$ і $I_2=f(S)$, значення опору пускового реостата варіювати в межах $0 \div 5 \cdot R_1$, **ковзання** S приймати **рівним**: 0.03; 0.09; 0.18; 0.26; 0.37; 0.48; 0.75; 0.87.

3.10. Проаналізувати вплив **величини** номінальної потужності двигуна P_n на вид статичних характеристик двигуна $M=f(S)$ і $I_2=f(S)$, значення номінальної потужності двигуна варіювати в межах $60 \div 110$ кВт із кроком 15кВт, **ковзання** S приймати **рівним**: 0.01; 0.05; 0.14; 0.21; 0.33; 0.45; 0.75; 0.9.

6.Методичні вказівки до виконання курсової роботи

Приклад виконання курсової роботи надається для наступної задачі. При пуску двигуна на холостому ході в активному опорі його роторному ланцюгу виділяється тепла енергія що дорівнює кінетичної енергії що приводяться в обертання махових мас, а при пуску під навантаженням кількість виділеної енергії відповідно збільшується. При частих пусках, а також при дуже важких умовах пуску, коли махові маси що приводяться в рух механізмів великі, виникає небезпека перегріву обмоток двигуна. При подачі повної напруги на статор мають місце два несприятливих чинника, а саме:

- велика кратність початкового пускового струму, яка досягає (6-10) I_n ,
- коливальний затухаючий характер пускового моменту двигуна.

Наслідки дії цих факторів: великий початковий пусковий струм викликає значні осідання напруги на живлять шинах підстанції (при сумірною потужності трансформатора і двигуна), що порушує роботу, як інших споживачів, так і самого двигуна (затягування пуску). Великий пусковий струм викликає також значні термічні перевантаження обмотки, наслідком чого може бути прискорене старіння ізоляції, її пошкодження і, як результат, міжвіткове коротке замикання

Значні коливання моменту двигуна на початковому етапі пуску, які можуть перевищувати 4-5 кратне значення номінального моменту, створюють несприятливі умови для роботи механіки (кінематичного ланцюга).

Стійкий момент електродвигуна гірського комбайна визначається по залежності

де K_u , K_{ng} , K_{br} – коефіцієнти, що характеризують умови експлуатації електродвигуна

M_{fmax} – максимальний фактичний момент електродвигуна, Н·М,

M_{max} – максимальний момент двигуна по паспорті, Н·М;

U_{tr} – напруга вторинної обмотки живильного трансформатора, В;

ΔU_n – утрати напруги від сторонніх споживачів, В;

I_K – критичний струм двигуна, А;

$$I_K = \frac{2I_{III} \cdot S_K^{(1/\lambda)}}{1 + S_K^{(1/\lambda)}}$$

I_{III} – початковий пусковий струм (по паспорті), А;

S_K – критичне ковзання;

λ – перевантажувальна здатність двигуна.

$$S_K = S_H \frac{\lambda + \sqrt{(\lambda - 1)(1 + \lambda + 2S_H)}}{1 - 2S_H(\lambda - 1)},$$

S_H – номінальне (по паспорті) ковзання двигуна;

$R_{тр}$, $x_{тр}$ – відповідно активний і індуктивний опір живильного трансформатора, Ом,

r_k , x_k - відповідно активний і індуктивний опір кабелю, Ом/км

L_k – довжина кабелю, км;

r_d , x_d - відповідно активний і індуктивний опір двигуна, Ом

$$r_d = \frac{U_H \cos \varphi_k}{\sqrt{3} I_K} \quad x_d = \frac{U_H \sin \varphi_k}{\sqrt{3} I_K}$$

U_H – номінальна напруга, В;

φ_k – кут зрушення фаз між струмом і напругою, град;

K_B – коефіцієнт вирівнювання двигуном динамічного навантаження, що складає

$$K_B = \frac{1}{\sqrt{\left[1 - \left(\frac{f_H}{f_O}\right)^2\right]^2 + \left(\frac{f_H}{f_O}\right)^2 \frac{T_M}{T}}}$$

Тут f_H – частота динамічного навантаження, Гц;

f_O – власна частота системи, Гц;

$$f_O = \frac{1}{2\pi \sqrt{T \cdot T_M}}$$

$$T_M = \frac{1,2 \cdot GD^2 \cdot n_C S_K}{750 M_{\phi \max}}, c \quad T = \frac{1}{\omega_C S_K}, c$$

ω_C – кругова частота живильної мережі, об/мин;

n_C – синхронна частота обертання ротора двигуна, с-1;

GD^2 – маховий момент ротора двигуна, Нм².

Чисельні значення параметрів для уС++х варіантів теми:

$R_{тр}=0,02$ Ом $x_{тр}=0,455$ Ом $r_k=0,37$ Ом/км $x_k=0,072$ Ом/км

$\varphi_k=0,65$ град $n_C=314$ с-1 $K_B=0,45$ $U_{тр}=690$ В

$\omega_C=1500$ Об/мин $U_H=660$ В $K_U=1,35$ $K_{Hr}=0,75$

Характеристика даних та їх умовні позначення

Табл. 1

Найменування	Позначення в постановці задачі	Позначення в алгоритмі та Excel	Позначення в програмі
Початковий пусковий струм	$I_{нп}$	Inp	Inp
Утрати напруги від сторонніх споживачів	ΔU_n	ΔUn	dUn
Номинальне ковзання двигуна	S_n	Sn	Sn
Перевантажувальна здатність двигуна	Λ	λ	lb
Довжина кабелю	L_k	Lk	Lk
Маховий момент ротора двигуна	GD^2	GD^2	GD2
Частота динамічного навантаження	f_n	fn	fn
Максимальний момент двигуна	M_{max}	Mmax	Mmax
Відповідно активний опір живильного трансформатора	$R_{тр}$	Rтр	Rtr
Відповідно індукційний опір живильного трансформатора	$x_{тр}$	xтр	xtr
Відповідно активний опір кабелю	r_k	rk	rk
Відповідно індуктивний опір кабелю	x_k	xk	xk
Кут зрушення фаз між струмом і напругою	φ_k	φ_k	fik
Синхронна частота обертання ротора двигуна	n_c	nc	nc
Коефіцієнт, що характеризує умови експлуатації електродвигуна	K_{br}	Kbr	Kbr
Напруга вторинної обмотки живильного трансформатора	$U_{тр}$	Uтр	Utr
Кругова частота живильної мережі	ω_c	ω_c	wc
Номинальна напруга	U_n	Un	Un
Коефіцієнт, що характеризує умови експлуатації електродвигуна	K_y	Ky	Ky
Коефіцієнт, що характеризує умови експлуатації електродвигуна	$K_{нг}$	Kнг	Kng
Критичне ковзання	S_k	Sk	Sk
Критичний струм двигуна	I_k	Ik	Ik
Відповідно активний опір двигуна	r_d	rd	rd
Відповідно індуктивний опір двигуна	x_d	xd	xd
Максимальний фактичний момент електродвигуна	$M_{фmax}$	Mфmax	Mfmax
	T	T	T
Власна частота системи	f_o	fo	fo
	T_m	Tm	Tm
Коефіцієнт вирівнювання двигуном динамічного навантаження	K_b	Kb	Kb
Стійкий момент електродвигуна гірського комбайна	M_y	My	My

Блок-схема алгоритму рішення задачі

Блок №1

(Пуск) – блок початку обчислювального процесу;

Блок №2

(Робота з даними) – блок вводу вихідних (заданих) величин, використовуваних у даному розрахунку;

Блок №3

(Блок модифікації) – параметром циклу є зміна i , яка представляє собою порядковий номер елементів одномірного масиву;

Блок №4

(Уведення даних) – блок уведення чисельного значення чергового елемента S_{ni} (номінального ковзання двигуна);

Блок №5

(Процес) – блок виконання арифметичних операцій, обчислюється чергове значення критичного ковзання S_{ki} ;

Блок №6

(Процес) – блок виконання арифметичних операцій, обчислюється чергове значення T_i ;

Блок №7

(Блок модифікації) – параметром циклу є зміна j , яка представляє собою порядковий номер елементів одномірного масиву;

Блок №8

(Уведення даних) – блок уведення чисельного значення чергового елемента I_{nj} (початкового пускового струму);

Блок №9

(Процес) – блок виконання арифметичних операцій, обчислюється чергове значення критичного струму двигуна $I_{ki,j}$;

Блок №10

(Процес) – блок виконання арифметичних операцій, обчислюється чергове значення відповідно активного опору двигуна $g_{d_i,j}$;

Блок №11

(Процес) – блок виконання арифметичних операцій, обчислюється чергове значення відповідно індуктивного опору двигуна $x_{d_i,j}$;

Блок №12

(Процес) – блок виконання арифметичних операцій, обчислюється чергове значення максимального фактичного моменту електродвигуна $M_{фmax_i,j}$;

Блок №13

(Процес) – блок виконання арифметичних операцій, обчислюється чергове значення $T_{m_i,j}$;

Блок №14

(Процес) – блок виконання арифметичних операцій, обчислюється чергове значення власної частоти системи $f_{oi,j}$;

Блок №15

(Процес) – блок виконання арифметичних операцій, обчислюється чергове значення

коефіцієнта вирівнювання двигуном динамічного навантаження K_{vj} ;

Блок№16

(Процес) – блок виконання арифметичних операцій, обчислюється чергове значення стійкого моменту електродвигуна гірського комбайну $Mu_{i,j}$;

Блок№17

(Блок модифікації) – параметром циклу є зміна i , яка представляє собою порядковий номер елементів одномірного масиву;

Блок№18

(Вивід даних) – блок виводу даних, які є результатами обчислень елементів одномірних масивів;

Блок№19

(Блок модифікації) – параметром циклу є зміна j , яка представляє собою порядковий номер елементів одномірного масиву;

Блок№20

(Вивід даних) – блок виводу даних, які є результатами обчислень елементів двомірних масивів;

Блок№21

(Кінець) – блок завершення обчислювального процесу.

Текст програми

```
# include <iostream.h>
# include <conio.h>
# include <math.h>
# include <fstream.h>
int main()
{
  clrscr();
  int i,j,n,m,Inp[8],dUn,GD2,Mmax,nc,Utr,wc,Un,fn;
  double Sn[5],lb,Lk,Rtr,xtr,rk,xk,fik,Kbr,Ky,Knr,Sk[5],T[5],pp;
  double Ik[5][8],Tm[5][8],Mfmax[5][8],rd[5][8],xd[5][8];
  double fo[5][8],Kb[5][8],My[5][8];
  cout.precision(3);
  fstream f1,f2;
  f1.open("data.txt",ios::in);
  f2.open("otvet.txt",ios::out);
  f1>>n; f1>>m; f1>>dUn; f1>>lb; f1>>Lk; f1>>GD2;f1>>fn; f1>>Mmax;
  f1>>Rtr; f1>>xtr; f1>>rk; f1>>xk; f1>>fik;f1>>nc; f1>>Kbr;
  f1>>Utr; f1>>wc; f1>>Un; f1>>Ky; f1>>Knr;
  f2<<"REZULTAT Massiv My"<<"\n";
  for (i=0;i<n;i++)
  { f1>>Sn[i];
    Sk[i]=(Sn[i]*(lb+sqrt((lb-1)*(1+lb+2*Sn[i]))))/(1-2*Sn[i]*(lb-1));
    T[i]=1/(wc*Sk[i]);
  }
  for (j=0;j<m;j++) f1>>Inp[j];
  pp=Mmax*pow((Utr-dUn),2);

  for (i=0;i<n;i++){
    for (j=0;j<m;j++)
    {Ik[i][j]=2*Inp[j]*exp(1/lb*log(Sk[i]))/(1+exp(1/lb*log(Sk[i])));
```

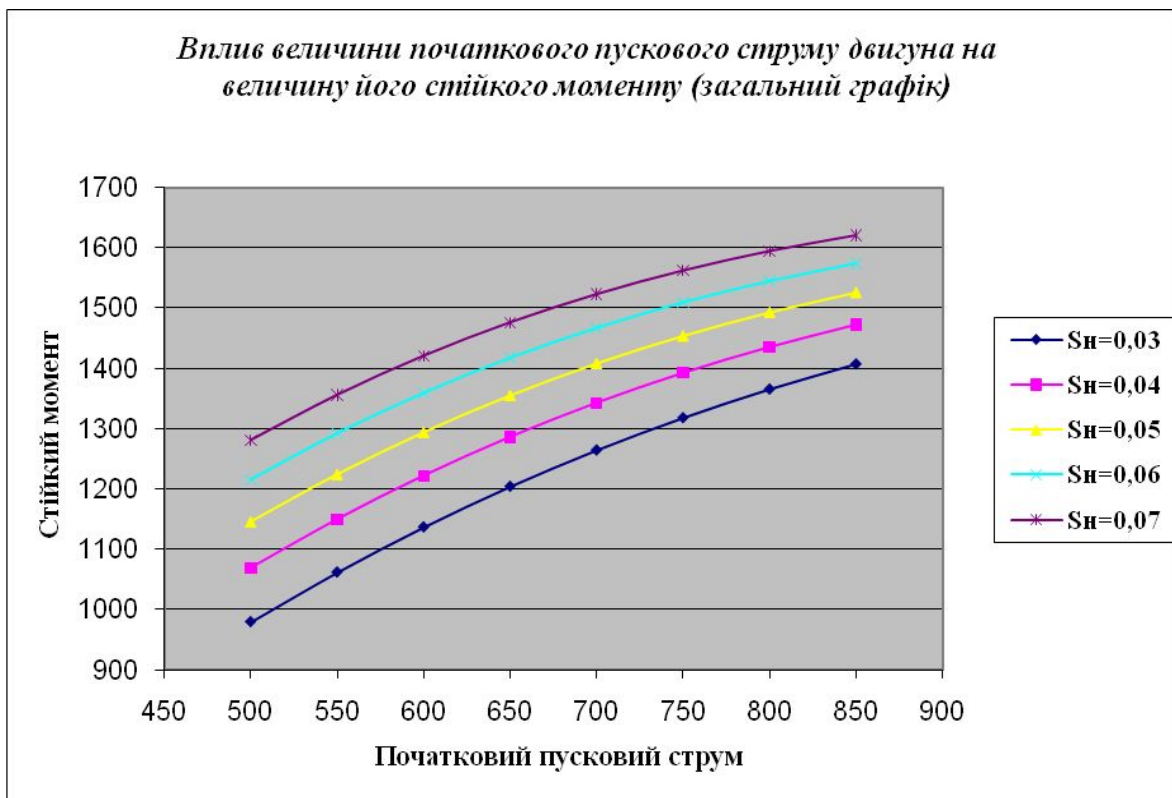
```

rd[i][j]=Un*cos(fik)/(sqrt(3)*Ik[i][j]);
xd[i][j]=Un*sin(fik)/(sqrt(3)*Ik[i][j]);

Mfmax[i][j]=pp/(3*pow(Ik[i][j],2)*pow((Rtr+rk*Lk+rd[i][j]),2)*pow((xtr+xk*Lk+xd[i][j]),2));
Tm[i][j]=(1.2*GD2*nc*Sk[i])/(750*Mfmax[i][j]);
fo[i][j]=1/(2*3.14159*sqrt(T[i]*Tm[i][j]));
Kb[i][j]=1/sqrt(pow((1-pow((fn/fo[i][j]),2)),2)+pow((fn/fo[i][j]),2)*Tm[i][j]/T[i]);
My[i][j]=(Ky*Mfmax[i][j])/(Knr*(1+Kb[i][j]*Kbr));
f2<<My[i][j]<<"\t";
}
f2<<"\n";
}
f2<<"\n";
f1.close();
f2.close();
return 0;
}

```

Аналіз результатів



графік 1.

Отриманий загальний графік впливу початкового пускового струму двигуна на величину його стійкого моменту з урахуванням номінального ковзання двигуна S_n . Значення номінального ковзання двигуна $S_n=0,03;0,04;0,05;0,06;0,07$.

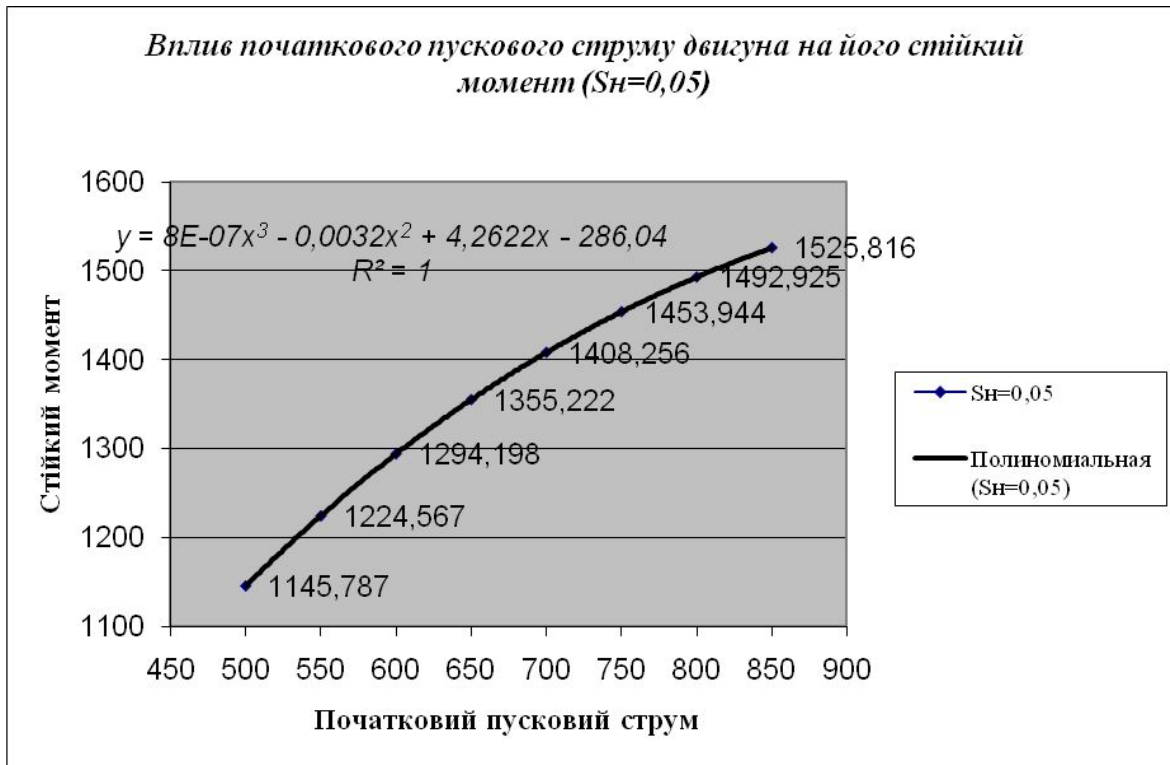


графік 2.

Отриманий графік впливу величини початкового пускового струму двигуна на величину його стійкого моменту з урахуванням номінального ковзання двигуна S_n являє собою поліноміальну 2 степені, зростаючу у міру збільшення початкового пускового струму, залежність виду $y=ax^2+bx+c$, яка при значенні номінального ковзання двигуна $S_n=0,03$ описується рівнянням $y=-0,0014x^2+3,0609x-209,32$.



Отриманий графік впливу величини початкового пускового струму двигуна на величину його стійкого моменту з урахуванням номінального ковзання двигуна S_n являє собою поліноміальну 2 степені, зростаючу у міру збільшення початкового пускового струму, залежність виду $y=ax^2+bx+c$, яка при значенні номінального ковзання двигуна $S_n=0,04$ описується рівнянням $y=-0,0015x^2+3,1236x-126,14$.



Отриманий графік впливу величини початкового пускового струму двигуна на величину його стійкого моменту з урахуванням номінального ковзання двигуна S_n являє собою поліноміальну 3 степені, зростаючу у міру збільшення початкового пускового струму, залежність виду $y=ax^3+bx^2+cx+d$, яка при значенні номінального ковзання двигуна $S_n=0,05$ описується рівнянням $y=8E-07x^3-0,0032x^2+4,2622x-286,04$.



Отриманий графік впливу величини початкового пускового струму двигуна на величину його стійкого моменту з урахуванням номінального ковзання двигуна S_n являє собою поліноміальну 3 степені, зростаючу у міру збільшення початкового пускового струму, залежність виду $y=ax^3+bx^2+cx+d$, яка при значенні номінального ковзання двигуна $S_n=0,06$ описується рівнянням $y=1E-06x^3-0,0036x^2+4,4629x-248,86$.



Отриманий графік впливу величини початкового пускового струму двигуна на величину його стійкого моменту з урахуванням номінального ковзання двигуна S_n являє собою поліноміальну 3 степені, зростаючу у міру збільшення початкового пускового струму, залежність виду $y=ax^3+bx^2+cx+d$, яка при значенні номінального ковзання двигуна $S_n=0,07$ описується рівнянням $y=1E-06x^3-0,0039x^2+4,6585x-216,05$.

Опис контрольного прорахунку

Чисельні значення первинних вихідних даних для обчислення стійкого моменту електродвигуна гірського комбайну знаходяться в наступних комірках:

- D4 – перевантажувальна здатність двигуна;
- E4 – довжина кабелю;
- F4 – маховий момент ротора двигуна;
- G4 – частота динамічного навантаження;
- H4 – втрати напруги від сторонніх споживачів;
- J4 – максимальний момент двигуна;
- A14 – відповідно активний опір живильного трансформатора;
- B14 – відповідно індукційний опір живильного трансформатора;
- C14 – відповідно активний опір кабелю;
- D14 – відповідно індуктивний опір кабелю;
- E14 – кут зрушення фаз між струмом і напругою;
- F14 – синхронна частота обертання ротора двигуна;
- G14 – коефіцієнт, що характеризує умови експлуатації електродвигуна;
- H14 – напруга вторинної обмотки живильного трансформатора;
- I14 – кругова частота живильної мережі;
- J14 – номінальна напруга;
- K14 – коефіцієнт, що характеризує умови експлуатації електродвигуна;
- L14 – коефіцієнт, що характеризує умови експлуатації електродвигуна.

В комірках B22:B26, по формулі
$$=(\$A22*(\$D\$4+((\$D\$4-1)*(1+\$D\$4+2*\$A22))^\wedge{0,5}))/((1-2*\$A22*(\$D\$4-1))$$
 виконується обчислення критичного ковзання S_k при номінальному ковзанні двигуна $S_n=0,03;0,04;0,05;0,06;0,07$ (комірки A22:A26).

В комірках B29:I33, по формулі
$$=(2*B\$28*\$B22^\wedge{(1/\$D\$4)})/((1+\$B22^\wedge{(1/\$D\$4)}))$$
 виконується обчислення критичного струму двигуна I_k при початковому пусковому струмі $I_{np}=500;550;600;650;700;750;800;850$ (комірки B28:I28) і різних значеннях S_n .

В комірках B35:I38, по формулі
$$=(\$J\$4*\text{COS}(\$E\$14))/((3^\wedge{0,5}*B29))$$
 виконується обчислення відповідно активного опору двигуна g_d при різних значеннях S_n і I_{np} .

В комірках B41:I45 по формулі
$$=(\$J\$4*\text{SIN}(\$E\$14))/((3^\wedge{0,5}*B29))$$
 виконується обчислення відповідно індуктивного опору двигуна x_d при різних значеннях S_n і I_{np} .

В комірках B47:I51 по формулі
$$=(\$J\$4*(\$H\$14-\$H\$4)^\wedge{2}/((3*B29^\wedge{2}*((\$A\$14+\$C\$14*\$E\$4+B35)^\wedge{2}*(\$B\$14+\$D\$14*\$E\$4+B41)^\wedge{2})))$$
 виконується обчислення максимального фактичного моменту електродвигуна $M_{фмах}$ при різних значеннях S_n і I_{np} .

В комірках A54:A58 по формулі
$$=1/(\$I\$14*\$B22)$$
 виконується обчислення T при різних значеннях S_n .

В комірках B60:I64 по формулі $= (1,2 * \$F\$4 * \$F\$14 * \$B22) / (750 * B47)$ виконується обчислення T_m при різних значеннях S_n і I_{np} .

В комірках B66:I70 по формулі $= 1 / (2 * \text{ПИ}() * (\$A54 * B60)^{0,5})$ виконується обчислення власної частоти системи f_0 при різних значеннях S_n і I_{np} .

В комірках B72:I76 по формулі $= 1 / ((1 - (\$G\$4 / B66)^2)^2 + (\$G\$4 / B66)^2 * (B60 / \$A54))^{0,5}$ виконується обчислення коефіцієнта вирівнювання двигуном динамічного навантажування K_v при різних значеннях S_n і I_{np} .

В комірках B78:I82 по формулі $= (\$K\$14 * B47) / (\$L\$14 * (1 + B72 * \$G\$14))$ виконується обчислення стійкого моменту електродвигуна гірського комбайну M_u при різних значеннях S_n і I_{np} .

ДОДАТКИ

Д1. Приклад оформлення титульного аркушу

ДЕРЖАВНИЙ ВИЩІЙ НАВЧАЛЬНИЙ ЗАКЛАД
МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Кафедра „Обчислювальної
математики і програмування”

ПОЯСНЮВАЛЬНА ЗАПИСКА

до курсовій роботі з обчислювальної математики і програмування

Тема роботи: „Розрахунок стійкого моменту електродвигуна”

Студента групи ХХ-10
(ПІБ)

Керівник роботи
(ПІБ)

Д2. Приклад оформлення сторінки завдання

ДЕРЖАВНИЙ ВИЩІЙ НАВЧАЛЬНИЙ ЗАКЛАД
МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет Електротехнічний
Кафедра «.....»
Напрямок підготовки електромеханіка

Завдання
на курсову роботу студентіві гр. ХХХ-10
(П І Б)

1. Тема роботи: „Розрахунок стійкого моменту електродвигуна”
2. Термін здачі студентом закінченої роботи 2010 р.
3. Вихідні дані до роботи:
 $500 < I_{\text{нп}} < 850 \text{ А}$, $\Delta U_{\text{п}} = 20 \text{ В}$, $S_{\text{н}} = 0,03; 0,04; 0,05; 0,06; 0,07$, $\lambda = 2,65$,
 $L_{\text{к}} = 0,35 \text{ км}$, $GD^2 = 35 \text{ Нм}^2$, $f_{\text{н}} = 16 \text{ Гц}$

Керівник роботи (П І Б)

Донецьк-2010

ДЗ. Приклад оформлення календарного плану

КАЛЕНДАРНИЙ ПЛАН

Номер етапу	Найменування етапу виконання курсової роботи	Термін виконання	Примітка

Студент

(ПІБ)

Керівник

(ПІБ)

Д4. Приклад оформлення змісту

ЗМІСТ

Вступ	5
1. Постановка задачі	6
2. Алгоритм розв'язання задачі	11
3. Характеристика даних та їх умовні позначення	12
4. Контрольний прорахунок	14
5. Програма та її опис	15
6. Графічний аналіз результатів	18
Висновок	20
Додатки	
Д1. Результати контрольного прорахунку	21

Д5. Приклад оформлення реферату

РЕФЕРАТ

ТАБЛИЦЯ, АЛГОРИТМ, БЛОК-СХЕМА, ГРАФІКИ ЗАЛЕЖНОСТЕЙ, СТІЙКИЙ МОМЕНТ, ПУСКОВИЙ СТРУМ, НОМІНАЛЬНЕ КОВЗАННЯ

сторінок - 27, таблиця - 1, малюнків - 3, джерел - 3, графіків - 6.

Об'єктом дослідження є вплив величини початкового пускового струму ІНП двигуна на величину його стійкого моменту.

У роботі проводиться розрахунок стійкого моменту електродвигуна гірського комбайну, виконується контрольний прорахунок, розробляється алгоритм розрахунку, будується графічний аналіз результатів.

Д6. Приклад оформлення переліку посилань

Список використаної літератури

1. Черняк И.Л., Яруник С.А., Бурчаков Ю.И. Технологія і механізація підземного видобутку вугілля: підручник для вузів. – М.:Надра,1981.
2. Виробничі процеси в очисних вибоях поліпшених шахт: Навчальний посібник для вузів/И.Ф. Ярембаш, В.Д. Мороз, И.С. Костюк. Під загальною редакцією И.Ф. Ярембаша. – Донецьк: Донгту,1999
3. Сергєєв Н.П., Доминин Л.Н. Алгоритмізація і програмування: Навчальний посібник для вузів: - М.:Радіо і зв'язок, 1982. – 232 с.

Д7. Приклад оформлення контрольного прорахунку

Microsoft Excel - Контрольний прорахунок, програма, графіки												
Файл Правка Вид Вставка Формат Сервіс Даньє Окно Справка												
Вихідні дані												
Інп (А)	Sn	λ	Lк (км)	GD ² (Н ² м ²)	fн (Гц)	ΔUn (В)			Mmax (ЭДК-475У5) (Н ² м)			
500	0,03	2,65	0,35	35	16	20			910			
550	0,04											
600	0,05											
650	0,06											
700	0,07											
750												
800												
850												
Rтр (Ом)	xтр (Ом)	гк (Ом/км)	хк (Ом/км)	φк (град)	nc (с ² (-1))	Kвг	Uтр (В)	ωс (Об/мин)	Un (В)	Kу	Kнг	
0,02	0,455	0,37	0,072	0,65	314	0,45	690	1500	660	1,35	0,75	
Обчислення												
Обчислення критичного ковзання Sk												
Sn	Sk											
0,03	0,171											
0,04	0,236											
0,05	0,308											
0,06	0,385											
0,07	0,469											
Обчислення критичного струму двигуна Ік												
Інп	500	550	600	650	700	750	800	850				
Ік	339,102	373,012	406,923	440,833	474,743	508,653	542,563	576,474				
	367,219	403,940	440,662	477,384	514,106	550,828	587,550	624,271				
	390,586	429,645	468,704	507,762	546,821	585,879	624,938	663,997				
	410,876	451,964	493,051	534,139	575,226	616,314	657,402	698,489				
	429,024	471,926	514,829	557,731	600,633	643,536	686,438	729,340				
Обчислення відповідно активного опору двигуна гд												
гд	0,895	0,813	0,745	0,688	0,639	0,596	0,559	0,526				
	0,826	0,751	0,688	0,635	0,590	0,551	0,516	0,486				
	0,777	0,706	0,647	0,597	0,555	0,518	0,485	0,457				
	0,738	0,671	0,615	0,568	0,527	0,492	0,461	0,434				
	0,707	0,643	0,589	0,544	0,505	0,471	0,442	0,416				

Мал. 1. Контрольний прорахунок

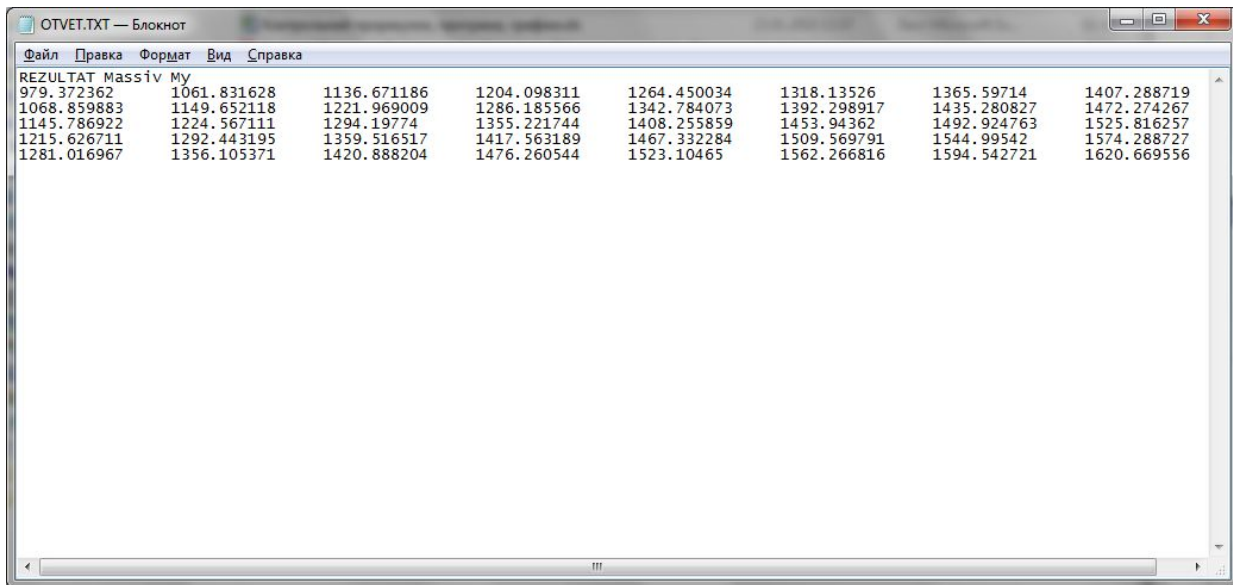
Microsoft Excel - Контрольний прорахунок, програма, графіки											
Файл Правка Вид Вставка Формат Сервіс Данні Окно Справка											Введіть запитання
A	B	C	D	E	F	G	H	I	J	K	L
Обчислення відповідно індуктивного опору двигуна x_d											
41		0,680	0,618	0,567	0,523	0,486	0,453	0,425	0,400		
42	x_d	0,628	0,571	0,523	0,483	0,449	0,419	0,392	0,369		
43		0,590	0,537	0,492	0,454	0,422	0,394	0,369	0,347		
44		0,561	0,510	0,468	0,432	0,401	0,374	0,351	0,330		
45		0,538	0,489	0,448	0,413	0,384	0,358	0,336	0,316		
46		Обчислення максимального фактичного моменту електродвигуна $M_{фmax}$									
47	$M_{фmax}$	806,956	875,107	936,721	992,076	1041,518	1085,428	1124,198	1158,222		
48		863,932	931,559	991,813	1045,132	1091,998	1132,913	1168,371	1198,849		
49		907,839	974,485	1033,129	1084,341	1128,723	1166,870	1199,359	1226,732		
50		943,491	1008,939	1065,887	1115,025	1157,050	1192,644	1222,447	1247,058		
51		973,489	1037,627	1092,859	1139,981	1179,777	1212,998	1240,344	1262,458		
52	Обчислення T										
53	T										
54	0,004										
55	0,003										
56	0,002										
57	0,002										
58	0,001										
59	Обчислення T_m										
60	T_m	0,004	0,003	0,003	0,003	0,003	0,003	0,003	0,003		
61		0,005	0,004	0,004	0,004	0,004	0,004	0,004	0,003		
62		0,006	0,006	0,005	0,005	0,005	0,005	0,005	0,004		
63		0,007	0,007	0,006	0,006	0,006	0,006	0,006	0,005		
64		0,008	0,008	0,008	0,007	0,007	0,007	0,007	0,007		
65	Обчислення власної частоти системи f_0										
66	f_0	41,757	43,485	44,990	46,300	47,440	48,429	49,287	50,027		
67		43,206	44,865	46,294	47,522	48,576	49,477	50,246	50,897		
68		44,291	45,888	47,248	48,405	49,386	50,213	50,907	51,485		
69		45,152	46,692	47,991	49,085	50,002	50,765	51,395	51,910		
70		45,864	47,351	48,595	49,631	50,490	51,196	51,770	52,229		
71	Обчислення коефіцієнта вирівнювання двигуном динамічного навантаження K_v										
72	K_v	1,074	1,074	1,074	1,073	1,073	1,072	1,071	1,070		
73		1,011	1,019	1,024	1,028	1,031	1,033	1,034	1,035		
74		0,947	0,961	0,971	0,978	0,984	0,988	0,991	0,994		
75		0,882	0,900	0,914	0,924	0,932	0,938	0,943	0,946		
76		0,818	0,838	0,854	0,867	0,876	0,884	0,889	0,894		
76											

Мал. 2. Контрольний прорахунок

Microsoft Excel - Контрольний прорахунок, програма, графіки											
Файл Правка Вид Вставка Формат Сервіс Данні Окно Справка											Введіть запитання
A	B	C	D	E	F	G	H	I	J	K	L
Обчислення стійкого моменту електродвигуна гірського комбайна M_y											
78	M_y	979,372	1061,832	1136,671	1204,098	1264,450	1318,135	1365,597	1407,289		
79		1068,860	1149,652	1221,969	1286,186	1342,784	1392,299	1435,281	1472,274		
80		1145,787	1224,567	1294,198	1355,222	1408,256	1453,944	1492,925	1525,816		
81		1215,627	1292,443	1359,517	1417,563	1467,332	1509,570	1544,995	1574,289		
82		1281,017	1356,105	1420,888	1476,261	1523,105	1562,267	1594,543	1620,670		

Мал. 3. Контрольний прорахунок

Д8. Зміст файлу результату.



РЕЗУЛТАТ Massiv My								
979.372362	1061.831628	1136.671186	1204.098311	1264.450034	1318.13526	1365.59714	1407.288719	
1068.859883	1149.652118	1221.969009	1286.185566	1342.784073	1392.298917	1435.280827	1472.274267	
1145.786922	1224.567111	1294.19774	1355.221744	1408.255859	1453.94362	1492.924763	1525.816257	
1215.626711	1292.443195	1359.516517	1417.563189	1467.332284	1509.569791	1544.99542	1574.288727	
1281.016967	1356.105371	1420.888204	1476.260544	1523.10465	1562.266816	1594.542721	1620.669556	

Мал. 4 Файл «Otvet.txt»

