

Національний технічний університет України
“Київський політехнічний інститут”
Факультет інформатики та обчислювальної техніки
Кафедра технічної кібернетики

ПРОГРАМУВАННЯ

Методичні вказівки
до

виконання курсової роботи

Частина 2

“ОБ’ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ”

Київ 2013

Програмування, ч.2. Об'єктно орієнтоване програмування: Метод. вказівки до виконання курсової роботи для студентів 1-го курсу напрямку “Системна інженерія” Уклад. О.І. Лісовиченко, О.П. Сирота – К.: НТУУ-КПІ, 2013 – 34с.

*Рекомендовано кафедрою
технічної кібернетики ФІОТ НТУУ “КПІ”
(протокол №1 від 30.08.2013 р.)*

Навчальне видання

**Програмування ч.2
Об'єктно-орієнтоване програмування**

**МЕТОДИЧНІ ВКАЗІВКИ
ДО ВИКОНАННЯ КУРСОВОЇ РОБОТИ
для студентів 1-го курсу напрямку “Системна інженерія”**

Укладачі *Лісовиченко Олег Іванович, к.т.н., доцент*
 Сирота Олена Петрівна, к.т.н., ст. викладач

Відповідальний
редактор *Ткач Михайло Мартинович, к.т.н., доцент*

Рецензенти *Павлов Олександр Анатолійович, д.т.н., професор*

ЗМІСТ

Вступ	5
1. Цілі і задачі курсової роботи	6
2. Тематика курсових робіт	7
3. Зміст курсової роботи і методичні вказівки по її виконанню	9
3.1. <i>Склад роботи</i>	9
3.2. <i>Зміст пояснювальної записки</i>	9
4. Вимоги до оформлення	12
4.1. <i>Оформлення пояснювальної записки</i>	12
4.2. <i>Оформлення графічного матеріалу</i>	14
5. Особливості складання програм на алгоритмічних мовах	20
5.1. <i>Особливості розробки та тестування програм</i>	20
5.2. <i>Особливості оформлення програм</i>	20
5.3. <i>Резервні копії та захист від комп'ютерних вірусів</i>	21
6. Організація виконання курсової роботи	22
6.1. <i>Видача завдань і затвердження тем</i>	22
6.2. <i>Виконання курсової роботи</i>	22
6.3. <i>Захист курсової роботи</i>	22
Список літератури	24
Основні стандарти	26
Додатки	27

ВСТУП

Основне завдання вивчення курсів “Алгоритмічні мови і програмування”, “Системне програмування” та “Об’єктно-орієнтоване програмування” – розвинути в студентів навички алгоритмізації задач, створення програм високої якості, реалізації їх на ПК, налагодження й одержання необхідного результату за обмежений час. Тому поряд з вивченням теоретичного курсу провідне значення набуває самостійна робота студентів, що передбачає виконання набору домашніх завдань, які допомагають поступово опановувати процес розробки програм на алгоритмічних мовах програмування, лабораторних робіт в учбових лабораторіях, у ході яких освоюється технологія розв’язання задач в операційних системах сучасних ПК.

Реалізація виробничих задач на ПК складається з наступних етапів:

1. Постановка задачі.
2. Побудова математичної моделі.
3. Вибір чисельного методу (алгоритмізація).
4. Програмування алгоритмічною мовою.
5. Налагодження.
6. Усебічна перевірка на контрольних прикладах.
7. Перевірка на реальних даних чи у реальних умовах.
8. Аналіз отриманих результатів.
9. Виправлення виявлених недоліків.

При розбіжності очікуваних результатів з вихідними даними, починаючи з п.5 (налагодження), повертаються на попередні етапи доти, доки ретельна перевірка не підтвердить відповідність отриманих даних початковій практичній задачі.

У процесі виконання домашніх завдань і лабораторних робіт студенти освоюють окремі прийоми та методи розробки програмного забезпечення на відносно простих прикладах. Тому в зазначених навчальних курсах передбачена курсова робота, де на задачах середньої складності освоюються прийоми комплексного налагодження, навички одержання необхідного результату, дослідження ефективності розроблених алгоритмів і програм.

1. ЦІЛІ І ЗАДАЧІ КУРСОВОЇ РОБОТИ

У результаті виконання зазначеної курсової роботи студенти повинні узагальнити і поглибити знання і розвинути навички алгоритмізації та програмування, самостійної інженерної роботи, а також опанувати операційні системи сучасних ПК, які використовуються для керування гнучкими комп'ютеризованими системами і ділянками роботизованих технологічних систем.

У процесі роботи студенти повинні вміти користатися науково-технічною та довідковою літературою, державними стандартами, розвивати творчий підхід до виконуваного завдання. Курсова робота повинна містити елементи наукових досліджень, пов'язаних з аналізом і вибором варіантів алгоритмів, порівнянням даного методу з альтернативними, дослідженням збіжності, точності, визначенням часу розв'язку задач у залежності від вхідних даних.

Деяку частину повинні складати комплексні роботи, де більш складна задача розв'язується невеликою групою студентів (бригадою). Це розвиває навички колективної праці, наближеної до реальних умов реалізації складних програм.

Оскільки дану роботу виконують у II семестрі, вона відрізняється від аналогічних робіт на старших курсах меншим обсягом наукових досліджень і відсутністю обґрунтувань методів і алгоритмів. Від курсових проектів її відрізняє менші обсяги і ступінь опрацювання інженерних рішень. Багато питань методики виконання курсової роботи відбиті в "Загальних вказівках по організації та методиці проведення курсового проектування у вищих навчальних закладах" [16].

2. ТЕМАТИКА КУРСОВИХ РОБІТ

Теми курсових робіт визначаються структурою і спрямованістю навчальних курсів “Алгоритмічні мови і програмування”, “Системне програмування” та “Об’єктно-орієнтоване програмування”, потребами навчального процесу і відбивають практичні вимоги господарства і науки. Вони пов’язані із суміжними дисциплінами (математичний аналіз, лінійна алгебра, чисельні методи) і повинні відбивати реальні напрямки розробки математичного і програмного забезпечення систем керування гнучкими комп’ютеризованими комплексами, що використовуються у науково-дослідних підрозділах кафедри.

Завдання на курсову роботу повинні враховувати індивідуальні інтереси і здібності студента без зниження загальних вимог і містити елементи новини.

У додатку 4 наводиться орієнтовний перелік тем із зазначенням рекомендованої літератури, яки розбитий на 2 підрозділи по складності:

1. теми завдань послабленої складності і розраховані на максимальне оцінювання в діапазоні від В (добре) до Е (задовільно);

2. теми завдань підвищеної складності і розраховані на максимальне оцінювання в діапазоні від А (відмінно) до Е (задовільно) в залежності від ступеню та якості виконання курсової роботи.

3. ЗМІСТ КУРСОВОЇ РОБОТИ І МЕТОДИЧНІ ВКАЗІВКИ ПО ЇЇ ВИКОНАННЮ

3.1. Склад роботи

Курсова робота складається з текстової та графічної документації. Текстова документація включає наступні частини:

	<i>Кількість сторінок</i>
Титульний аркуш	1 с
Завдання	1 с
Пояснювальну записку, що містить у собі:	
Зміст	1 с
Вступну частину	1-2 с
Обґрунтування і вибір алгоритму	4-6 с
Розробка програми	4-6 с
Керівництво користувачу	2-4 с
Керівництво розробнику	2-4 с
Отримані результати:	6-8 с
1. особливості трансляції, компонування і налагодження;	
2. результати рахунку контрольного прикладу;	
3. дослідження програми на прикладах загального вигляду;	
Висновки	1 с
Літературу	1-2 с
Додатки	1-6 с
1. Рисунки (скріншоти) роботи програми	
2. Лістинг (код) програми	
3. Блок-схема роботи програми (формат А1)	
4. Компакт диск:	
1. електронна версія пояснювальної записки;	
2. інсталяція програми;	
3. код програми	
4. електронна версія блок-схеми.)	
Всього	25-45 с

Графічний матеріал являє собою схему чи алгоритм програми на одному аркуші формату А1 [33].

У додатку обов'язково повинен бути наведений текст програми та результати обчислень. Крім того, для перевірки подається програма на машинному носії (дискеті, компакт-диску).

Обсяг програми має бути таким:

- для студентів денної форми навчання – від 1500 операторів;
- для студентів заочної форми навчання – від 800 операторів.

Найменування розділів пояснювальної записки формулюються виходячи з конкретних особливостей розв'язуваної задачі. При необхідності кількість розділів пояснювальної записки може бути збільшена. Допускається також виконання в пояснювальній записці додаткових схем алгоритмів, програм, включення до неї різних таблиць та графіків, що характеризують ефективність розроблених алгоритмів та програм.

3.2. Зміст пояснювальної записки

Пояснювальна записка – це програмний документ, її виклад, повинен бути коротким, ясным і прийнятим в технічній літературі. Опис відомих положень та рішень, що викладені в довідкових матеріалах не допускається. Основне завдання студента під час виконання курсової роботи – показати рівень володіння теоретичним та практичним матеріалами елементів написання коду програм для вирішення поставленої задачі. Тому, у першу чергу, варто відбивати принципові моменти – обґрунтування прийнятих рішень. У пояснювальній записці мають бути висвітлені обсяги виконаної роботи, складність розв'язуваної задачі і використаних методів, елементи новини, рівень практичної і теоретичної підготовки студента.

Пояснювальна записка – це єдиний документ, що характеризується цілісністю викладів від постановки задачі до аналізу отриманих результатів.

Взаємозв'язок окремих частин записки повинен чітко проглядатися; попередні розділи визначають напрямок рішення наступних. Кожен розділ має закінчуватися короткими висновками.

Вступна частина. Висвітлюються загальні питання, пов'язані з проблемою роботи: коротке обґрунтування актуальності теми і постановка задачі розробки:

1. навести аналогічні програми;
2. висвітлити їх недоліки;
3. обґрунтувати актуальність вашої роботи.

Обґрунтування і вибір алгоритму. На підставі математичної задачі, вхідної інформації, вимог до результатів рішення, наявних обчислювальних ресурсів і операційної системи, порівнюються альтернативні методи рішення (похибка, швидкість збіжності, кількість обчислювальних операцій) і вибирається найкращий (раціональний) з них. Якщо метод рішення заданий, його необхідно порівняти з іншими можливими, відзначивши переваги і недоліки, область застосування.

Враховуючи подання даних у конкретній ЕОМ та операційній системі, обсяг вхідної інформації і її розміщення, а також вимоги до представлення результатів, розробляється алгоритм рішення поставленої задачі. Для зручності налагодження алгоритм повинен будуватися за модульним принципом. Тому рекомендується спочатку представити його збільшено, виділивши повторювані процедури чи характерні частини. Усвідомивши взаємозв'язок цих блоків, розробляються більш деталізовані алгоритми окремих складових. При цьому ступінь деталізації повинна бути розумною, достатньою для розуміння алгоритму в цілому. Очевидні фрагменти можна детально не проробляти. Якщо ж існують неясності чи виникають сумніви в послідовності обчислень, ці частини рекомендується проробити детальніше з виконанням відповідних схем.

Якщо виконується комплексна тема чи частина складної програмної системи, спочатку варто визначити призначення і функції всієї системи. Далі розробити комплекс задач, що реалізують відповідні функції, і привести повний перелік задач системи. Після, цього доцільно виділити задачі, що підлягають розробці, обґрунтувати методи й алгоритми їхнього рішення.

Розробка програми. У цьому розділі знаходиться опис програми і процесу її розробки, достатні для розуміння її структури й особливостей використання.

Відповідно до [25] опис програми містить у собі такі підрозділи:

- загальні відомості;
- функціональне призначення;
- опис логічної структури;

використовувані технічні засоби;
виклик і завантаження;
вхідні дані;
вихідні дані.

У “загальних відомостях” вказуються позначення і найменування програми, програмне та апаратне забезпечення, необхідне для її функціонування (операційна система, зовнішні пристрої), мови програмування, на яких написана програма.

“Функціональне призначення” – дані про клас розв’язуваних задач або про функції, що виконуються даною програмою в деякій програмній системі, а також необхідні відомості про обмеження в застосуванні.

Оскільки алгоритм рішення задачі і використані методи обчислень були зазначені в попередньому розділі, у підрозділі “Опис логічної структури” відбиваються особливості побудови програми з описом функцій її складових частин і зв’язків між ними, а також зв’язок даної програми з іншими програмами. Необхідно охарактеризувати окремі процедури (внутрішні функції) і програмні модулі, специфіку їхньої побудови (особливості передачі параметрів, використання пам’яті і т.п.). Якщо в програмі використовуються макрокоманди, їх варто охарактеризувати аналогічно. Описується застосування системних засобів (системних макрокоманд, підпрограм, засобів операційної системи). Опис логічної структури супроводжується посиланнями на текст програми вихідною мовою.

Для визначення можливості використання розробленої програми вказується тип ПК, обсяг оперативної пам’яті і набір необхідних зовнішніх пристроїв.

Далі варто описати послідовність виклику (установки) програми з відповідного носія даних на ПК (для мов програмування низького рівня необхідно описати особливості трансляції, компонування і завантаження).

Наприкінці розділу описуються вхідні і вихідні дані програми. Відбивається характер, організація і попередня підготовка вхідних даних, використані формати перетворення і способи кодування даних. При цьому програма повинна бути стійкою до змін розмірності і допускати можливість зміни діапазону вхідних даних, що призводить до завдання деякої максимальної розмірності масивів у головній програмі, введенню конкретних значень границь масивів, використанню в модулях-процедурах масивів регульованих розмірів. Вхідні дані, як правило, повинні вводитися в процесі рішення задачі чи зберігатися в окремих файлах.

Аналогічно описуються і вихідні дані: відзначається характер і організація, формати і, при необхідності, спосіб кодування.

Текст програми, особливо на мовах програмування низького рівня, повинен містити в собі достатню кількість коментарів, необхідних для розуміння її структури, особливостей експлуатації й обмежень на застосування. Виведення проміжних і остаточних результатів також повинне супроводжуватися інформацією, що не тільки поліпшує їхнє сприйняття, але й містить вказівки щодо дій оператора в особливих випадках.

Керівництво користувачу. У цьому розділі визначається послідовність виконання інсталяції, якщо є необхідність, а також алгоритм виконання функцій програми. Описується структура функціональних можливостей програми. Викладається алгоритм розв’язування задач. Необхідні скріншоти (копії екранних форм) додаються безпосередньо в текст роз’яснення або виносяться окремо в додаток і тоді на них в керівництві робляться посилання.

Керівництво розробнику. У цьому розділі визначається перелік функцій (процедур, класів і т.д.), їх функціональне призначення. Дається перелік необхідних для роботи бібліотек. Якщо є необхідність, описують структури вхідних та вихідних даних.

Отримані результати. У цьому розділі визначаються особливості етапів трансляції, компонування і налагодження програми, перевірки правильності її роботи на контрольних прикладах, а також дослідження її ефективності на більш загальних вхідних даних.

Необхідно зазначити файли вхідних та вихідних даних, їхні розміри, порядок трансляції, допущені помилки, їх аналіз і усунення. Можна відзначити системні програми, що використовувались при розробці, редагуванні і налагодженні.

Аналогічно вказуються особливості компонування програми, помилки і їхнє усунення. Відзначаються використані прийоми налагодження (спеціальні дебагери чи ж додаткові виводи даних в характерних частинах програми). Підтвердженням правильності роботи програми є розрахунок на контрольному прикладі з заздалегідь відомими результатами. Якщо програма припускає кілька варіантів обчислювальних процедур у залежності від вихідних даних, варто випробувати і відповідний набір контрольних прикладів. Якщо не вдається отримати повного збігу результатів, необхідно виявити причину таких розбіжностей і намітити можливі шляхи їх подолання.

Крім контрольних прикладів, програма повинна бути випробувана на вхідних даних загального вигляду, що відбивають сутність даного алгоритму і мають відносно велику розмірність. При цьому необхідно провести дослідження збіжності методу, чи досягається точність, швидкодію в залежності від обсягу вихідних даних. Ці результати бажано подати у вигляді графіків, таблиць, лістингів результатів окремих ітерацій і т.п. Крім того, можна досліджувати можливості розв'язку задачі на конкретній конфігурації ПК із заданим обсягом пам'яті та інших ресурсів.

Висновки. Коротко зазначаються експлуатаційні характеристики програми (обсяг оперативної пам'яті, що використовує програма, швидкодія), обмеження на її використання з огляду на вхідні дані і їх обсяг. Наводиться висновок щодо правильності її роботи і вірності одержаних результатів. Якщо ж поставленої мети досягти не вдалося, вказуються можливі причини і визначаються шляхи подолання ускладнень. При виявленні цікавих наукових результатів намічається перспектива їхнього використання чи продовження досліджень.

4. ВИМОГИ ДО ОФОРМЛЕННЯ

4.1. Оформлення пояснювальної записки

4.1.1. Пояснювальна записка і креслення оформляються згідно [45, 76] виключно українською мовою.

Слід використовувати наступні відступи від краю аркуша: верхній - **25мм**, нижній - **25мм**, лівий - **25мм**, правий - **10мм**. Основний текст набирається шрифтом **Times new Roman** розмір **14**, міжстроковий інтервал – півтора (**1,5**). Для спеціальних символів та формул необхідно використовувати редактори формул MathType та Microsoft Equation. Всі рисунки, створені в Microsoft Word, повинні бути згруповані в один об'єкт. Перша сторінка пояснювальної записки – титульний аркуш, що оформлюється згідно додатку 1. Після цього міститься завдання на курсову роботу (додаток 2,3), бланк (або електронна версія) якого видається викладачем.

Допускається виконання пояснювальної записки у рукописному, машинописному та електронному виглядах.

Пояснювальна записка виконується на окремих аркушах: сам текст, а далі внизу проставляється маленька рамка, в якому вписується номер аркуша та шифр документа. Рамки (границі) формату сторінки документа наносити обов'язково.

Курсова робота передбачає виконання двох програмних документів: пояснювальної записки і схеми алгоритму чи програми. Згідно [22, 46] кожному програмному документу присвоюється шифр, що має наступну структуру:

xx.xx.xxxxx-xx xx,

де в перших двох позиціях вказується код країни, у наступних двох – код організації-розробника, у п'ятьох наступних позиціях – реєстраційний номер; далі – номер редакції (дві позиції) і номер документа (дві позиції).

Відповідно до [46, 76] слід використовувати наступний шифр: у першій позиції – UA; у наступних двох – шифр спеціальності (ІК); далі в чотирьох позиціях – номер залікової книжки; п'ята позиція – нуль; номер редакції – 01; пояснювальна записка – 01, а схема чи алгоритм програми – 02.

4.1.2. Текст пояснювальної записки поділяється на розділи, кількість яких визначається конкретною задачею. Розділи повинні мати порядкові номери, позначені арабськими цифрами з крапкою. Перший розділ – “Обґрунтування і вибір алгоритму”. Найменування розділів повинні бути короткими, визначатися темою роботи, відповідати змісту частини роботи і записуватися великими літерами з нової сторінки.

Розділи можуть складатися з підрозділів, що мають порядкові номери в межах розділу. Повний номер підрозділу складається з двох чисел – номера розділу і підрозділу, розділених крапкою. Наприкінці номера також ставиться крапка. Найменування підрозділу записується малими літерами. Між підрозділами додатково пропускається один інтервал.

При необхідності, розділи можна поділяти на пункти і підпункти. Кожен пункт нумерується в межах підрозділу. Загальний номер пункту складається з номера розділу, підрозділу і пункту, розділених крапками. Пункт починається з абзацу і може мати заголовок, що записується малими літерами. Між пунктами додаткових пропусків не робиться.

Підкреслювати і переносити слова в заголовках не допускається. Крапка наприкінці заголовка не ставиться, а якщо заголовок складається з двох речень, їх розділяють крапкою.

Текст пояснювальної записки повинен бути коротким та чітким, виключати можливість суб'єктивного тлумачення. Термінологія і визначення повинні відповідати нормам встановленим у стандартах, а при їхній відсутності – загальноприйнятим. Якщо в записці використовується специфічна термінологія або скорочення, повинен бути наведений перелік прийнятих термінів і скорочень з роз'ясненнями.

4.1.3. Умовні літерні позначення фізичних величин повинні відповідати встановленим стандартам. При розрахунках необхідно користатися одиницями СІ. При визначенні одиниць виміру величин необхідно дотримуватися вимог стандарту [35].

Формули нумеруються арабським цифрами в межах розділу з правого боку аркуша на рівні формули в круглих дужках. Посилання в тексті на формулу наводять у круглих дужках, наприклад: “дисперсія помилки визначається у відповідності з формулою (2.3)”.

Позначення величин і параметрів у тексті доцільно пояснити. Значення символів і числових коефіцієнтів, що входять у формулу, вказуються відразу під нею. Значення кожного символу наводять з нового рядка в послідовності, прийнятій у формулі. Перший рядок розшифровки повинен починатися зі слова “де”. Формула розглядається як звичайний елемент речення. З урахуванням цього необхідно ставити розділові знаки після формул.

4.1.4. Текст пояснювальної записки може бути ілюстрований необхідними схемами, малюнками, таблицями та графіками.

Ілюстрації виконуються на аркуші паперу формату А4, допускаються малюнки формату А3 (297х420 мм), використання інших форматів небажано. Забороняється використовувати вирізки з інших джерел. Всі ілюстрації повинні мати номери в межах розділу або написи, що пояснюють їхній зміст, і розміщуватися після відповідного посилання на них у тексті.

Згідно [29] найменування розміщують над пояснюючим малюнком, дані – під ним. Номер ілюстрації розташовують нижче пояснюючих даних.

Графіки будуються в масштабі на звичайному аркуші чи міліметровці. По координатних осях вказуються відповідні величини і їхні одиниці виміру. Якщо на одному графіку наведено кілька кривих, бажано зобразити їх різними типами ліній або різними кольорами.

У пояснювальній записці на кожен малюнок обов’язково повинно бути посилання в тексті.

Схеми алгоритмів і програм можуть міститися на декількох аркушах. На першому аркуші вказується номер малюнка, найменування і номер аркуша, на наступних аркушах найменування можна не проставляти.

4.1.5. Цифровий матеріал рекомендується оформляти у вигляді таблиць і наводити тільки значущі цифри. Не допускаються похилі лінії в заголовках таблиць. Для зручності посилань графі нумерують. При переносі таблиці на інший аркуш заголовок розташовують тільки над першою частиною.

Графу “№ п/п” у таблицю не включають. При необхідності нумерації даних порядкові номери вказують у першому ліворуч стовпці перед їх найменуванням. Якщо числові дані в графах таблиці мають різні одиниці, їх вказують у заголовку кожної графи. Ставити лапки замість повторюваних цифр, знаків та математичних символів не допускається. Якщо цифрові чи інші дані в таблиці не вказуються, у графі ставиться риска. Для скорочення тексту заголовків і підзаголовків граф окремі поняття замінюють літерними позначеннями, якщо вони пояснені в тексті чи зазначені на малюнках. Таблиці нумерують арабськими цифрами в межах розділів. Над правим верхнім кутом поміщають напис з номером, наприклад: таблиця 2.1.

4.1.6. Окремі ілюстрації, таблиці, допоміжний текст, лістинги програм рекомендується розташовувати в додатку з посиланням на них в основному тексті. Кожен додаток треба починати з нової сторінки з зазначенням в правому верхньому куті слова “Додаток” і тематичного заголовка. При наявності в тексті більш ніж одного додатка усі вони нумеруються арабськими цифрами.

Нумерація сторінок пояснювальної записки і додатків повинна бути наскрізна.

4.1.7. У пояснювальній записці має бути наведений список використаної літератури. До нього потрібно обов'язково включати використані звіти, проекти, технічну документацію і т.п.

Перелік літератури оформляється згідно [34].

Відомості про книги (монографії, підручники, довідники) включають у себе: прізвище та ініціали авторів, назву книги, місце видання, видавництво (чи видання), рік видання і загальна кількість сторінок. Якщо авторів більше трьох, допускається вказувати прізвища тільки перших трьох з них і слова “та ін.”. Назву книги необхідно наводити в тому вигляді, у якому вона надрукована на титульному аркуші. Місце видання наводиться цілком у називному відмінку. Найменування видавництва наводиться без лапок.

Відомості про статтю з періодичного видання містять у собі: прізвище та ініціали автора, назву статті, найменування видання (журналу), номер, рік випуску, номера першої й останньої сторінок, на яких розташована стаття. Назву статті вказують так, як вона зазначена в оригіналі. Видавництво вказується тільки для разових збірників.

Відомості про проектну й іншу технічну документацію повинні містити: заголовок, вид документації; організацію, що випустила документ, місто і рік випуску.

Відомості про іноземні джерела вказуються мовою оригіналу, при цьому найменування на японському й інших рідких мовах записують у українській транскрипції.

Літературні джерела вказуються або в порядку вживання, або за алфавітом. Іноземні видання містяться після українських.

У тексті пояснювальної записки обов'язково повинні бути посилання на використану літературу. При цьому вказується в квадратних дужках номер джерела відповідно до списку, наприклад: “значення коефіцієнта K_n згідно [63] приймається рівним x ”.

Посилання на неопубліковані матеріали забороняються.

4.2. Оформлення графічного матеріалу

Схеми алгоритмів і програм повинні бути виконані у суворій відповідності до стандартів [26, 31, 33].

Креслення повинні бути виконані олівцем на щільному білому папері формату А1. Допускається машинне виконання документів.

Згідно [26] для позначення операторів використовуються наступні символи:

- 1) процес – виконання операції чи групи операцій, у результаті яких змінюється значення, форма подання чи розташування даних (рис.1,а);
- 2) рішення – вибір напрямку виконання алгоритму в залежності від деяких змінних умов (рис.1,в);
- 3) визначений процес – використання раніше створених або окремо описаних алгоритмів та програм (рис.1,е);
- 4) ввід/вивід (рис.1,б);
- 5) початок/кінець, переривання обробки даних (рис.1,к);
- 6) цикл – циклічне повторення одного або кількох блоків та операторів (рис.1,г,д).

Для кращого читання схем можна використовувати також символи:

з'єднувач (рис.1,і);

коментар (рис.1,з).

При виконанні схем алгоритмів основний розмір a повинен бути обраний з ряду 10, 15 або 20 мм. Допускається збільшити це значення на число, кратне 5. Розмір b складає $1,5 \cdot a$ (при ручному виконанні схем допускається $b = 2 \cdot a$).

Символи нумеруються порядковими номерами або координатами зон (по горизонталі – арабські цифри, по вертикалі – великі літери латинського алфавіту).

При ручному виконанні схеми в її межах допускається застосовувати не більше двох суміжних типорозмірів символів.

Лінії потоків повинні бути паралельні лініям зовнішньої рамки схеми (рис.2,а).

Напрямок лінії потоку згори донизу і зліва направо стрілкою можна не позначати. В інших випадках стрілки обов'язкові.

Відстань між окремими символами схеми повинна бути не менше 10 мм.

Для пояснення процесів всередині символів розміщують короткі записи. Для зручності читання схем використовують коментарі до символів та ліній потоку.

За наявності у схемі або алгоритмі циклічних ділянок (циклів) слід використовувати символи початку та кінця циклу (рис.1,г,д). В середині символів початку та кінця циклу обов'язково вказується ідентифікатор циклу. Для циклів з передумовою умова виходу з циклу вказується в символі початку циклу, для циклів з післяумовою – в символі кінця циклу. Приклад оформлення циклу з передумовою наведено на рис.2,б.

У межах одного аркуша лінії потоку можна розривати із застосуванням з'єднувачів, усередині яких міститься їхній унікальний ідентифікатор (рис.3,а).

Якщо ж лініями потоку з'єднують символи на різних аркушах, так само як і в попередньому випадку використовується з'єднувач, усередині якого міститься унікальний ідентифікатор. Якщо ж до одного символу підходить кілька ліній потоку, усі вони перелічуються в коментарі (рис.3,б).

Можливі варіанти відображення ходу рішення при кількості можливих виходів два, три і більше показані на рис.4.

Для складних алгоритмів рекомендується використовувати схеми різного ступеня деталізації. При цьому в збільшеному вигляді частина алгоритму може позначатися одним символом, усередині якого проводиться додаткова горизонтальна лінія, над якою міститься ідентифікатор програми, що буде далі деталізована (XB4), а праворуч – номер листа (015) і координати зони (B3), де розміщується символ “Початок або кінець”. Усередині символу “Початок або кінець”, що позначає початок програми, що деталізується, вказується її ідентифікатор (рис.5).

Аркуш креслення повинен бути заповнений, тому на одному аркуші рекомендується розміщувати приблизно 30-35 символів.

Формат аркуша обводиться тонкою лінією. Рамка робочого поля обводиться основною лінією і розташовується на відстані 5 мм зверху, знизу та праворуч і 20 мм ліворуч.

Схеми алгоритмів і програм згідно [23] повинні мати основний напис (рис.7) і додаткові граfi (рис.6). В залежності від орієнтації аркуша ці написи розташовуються так, як показано в на рис.8.

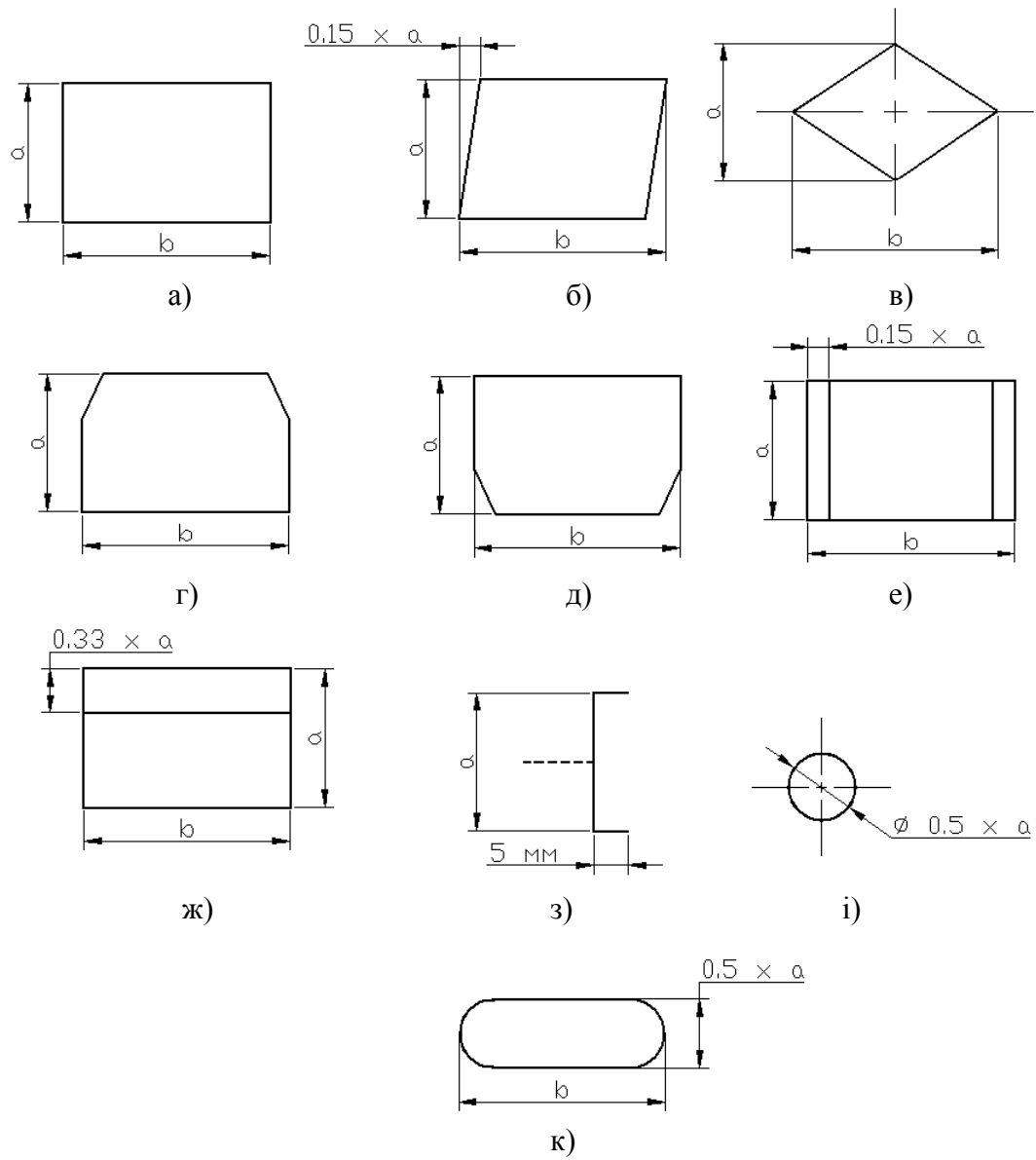


Рис.1 Основні символи

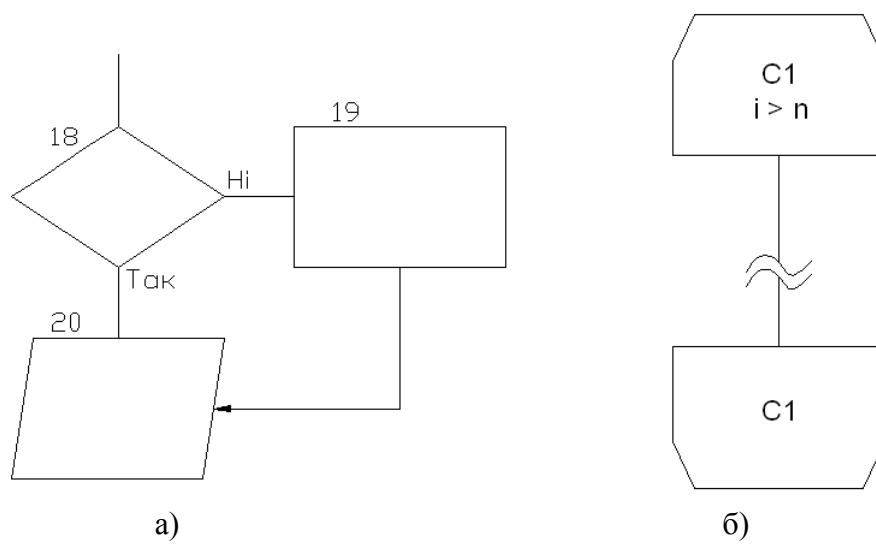


Рис.2 Типові ділянки схем

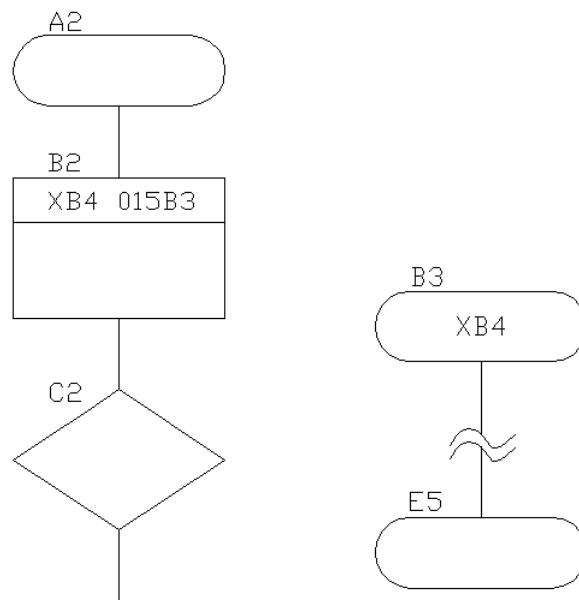


Рис.5 Використання різних ступенів деталізації

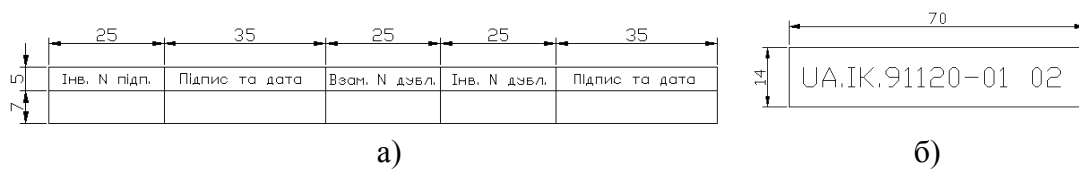


Рис.6 Допоміжні графи



Рис.7 Основний напис для форматів більших за A4

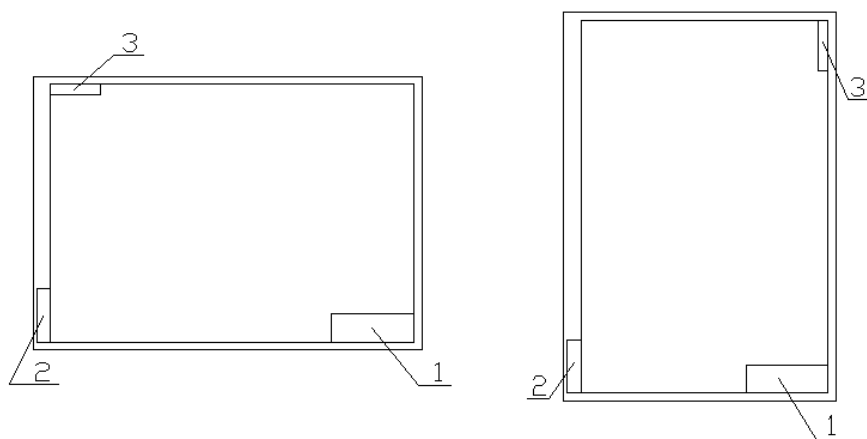


Рис.8 Розташування основного напису та допоміжних граф
(1 – рис.7; 2 – рис.6,а; 3 – рис.6,б)

5. ОСОБЛИВОСТІ СКЛАДАННЯ ПРОГРАМ НА АЛГОРИТМІЧНИХ МОВАХ

Залежно від характеристик ПК, наявного програмного та апаратного забезпечення, а також від характеру самої задачі (обчислювальна зі складною логікою або пов'язана з аналізом окремих бітів або байтів даних) обирається мова програмування. Можуть зустрітися задачі, програми яких складаються з модулів, виконаних на різних мовах.

До програмування варто приступати тільки після появи чіткого розуміння алгоритму. У загальному випадку за трудомісткістю алгоритмізація складає приблизно 40%, програмування 20% і налагодження 40% часу. Найбільші труднощі в студентів пов'язані з розробкою алгоритму. Тому в процесі налагодження програми, як правило, коректується і сам алгоритм.

5.1. Особливості розробки та тестування програм

Під час розробки, тестування та налагодження програмної частини курсового проекту рекомендується використовувати інтегровані середовища розробки програмного забезпечення (особливо в тому випадку, коли для розробки програми обрано мову програмування високого рівня). При розробці програмного забезпечення, що має працювати під керівництвом операційної системи MS DOS можна використовувати Borland C++, Borland Turbo Pascal, Microsoft Turbo Basic. При розробці ж візуального програмного забезпечення для ОС Windows широко використовуються Borland Delphi та пакет Microsoft Visual Studio. Вказані програми мають широкі можливості щодо набору та редагування тексту програми та програмних модулів, мають гнучкі можливості у навігації крізь текст програми, вбудовані компілятори, дебагери та інші утиліти, корисні під час написання програми.

5.2. Особливості оформлення програм

Складні програмні системи розробляються колективом програмістів і, як правило, згодом модифікуються. Тому кожна програма повинна бути добре описана, так щоб її зміг зрозуміти і використовувати інший розробник.

Розрізняють зовнішню і внутрішню програмну документацію. У першій відбиваються загальні характеристики програми для тих, хто може і не бачити самого тексту програми. Це в основному ті відомості, що викладаються в пояснювальній записці до курсової роботи. Внутрішня документація реалізується через коментарі в тексті програми.

Програма належної якості повинна містити коментарі до всієї програми й окремих її частин. Пояснення до всієї програми повинні включати:

- ім'я програми і її назва;
- ім'я програміста;
- дату розробки (номер версії);
- стиснутий виклад алгоритму;
- особливості вхідних даних;
- імена і призначення основних змінних;
- викликувані підпрограми;
- особливі чи аварійні ситуації.

Програма повинна містити коментарі як до окремих її частин (розміром близько 30 команд), так і до логічно завершених груп інструкцій (5-6 команд). Кожну інструкцію пояснювати недоцільно.

Для наочності всі речення програми не слід починати з однієї і тієї ж позиції, а використовувати відступи. Наприклад, заголовок і заключний рядок циклу друкувати з

7 позиції, а тіло – розміщати правіше. Аналогічно можна розміщати внутрішні цикли і коментарі.

5.3. Резервні копії та захист від комп'ютерних вірусів

Враховуючи досить великий обсяг курсової роботи та бажання студентів виконувати текстову та графічну її частини на ПК рекомендується регулярно виконувати резервні копії важливих для роботи даних на змінні носії інформації (дискети, компакт-диски). Ці резервні копії дають можливість продовжити виконання курсової роботи у разі виходу з ладу ПК або жорсткого диску, на якому зберігалися дані курсової роботи.

З метою запобігання розповсюдження комп'ютерних вірусів та захисту інформації від наслідків їх роботи при появі симптомів зараження слід виконати сканування на наявність вірусів виконавчих модулів та документів даного ПК. До основних симптомів можна віднести такі факти: деякі програми не можуть бути запущені, після запуску програм операційна система “зависає”, не вдається завантажити операційну систему, некоректна робота програм та інші.

6. ОРГАНІЗАЦІЯ ВИКОНАННЯ КУРСОВОЇ РОБОТИ

6.1. Видача завдань і затвердження тем

Завдання на курсову роботу необхідно видати на самому початку семестру. При цьому варто провести вступну лекцію, дати загальну характеристику роботи: призначення, вихідні дані, приблизний обсяг і зміст окремих частин пояснювальної записки, графічної частини, перелік методичної літератури, приблизний графік, а також вимоги щодо порядку виконання і розклад консультацій.

Видача окремого завдання здійснюється в ході індивідуальної бесіди керівника зі студентом, що є необхідною умовою, яка забезпечує успіх подальшого керівництва розробкою. При цьому необхідно з'ясувати ступінь підготовленості та зацікавленості студента у виконанні завдання, дати рекомендації з урахуванням особливостей теми, науково-теоретичної і практичної підготовленості, зазначити необхідну літературу і довідкові матеріали, уточнити порядок виконання і календарний графік.

У результаті бесіди тема завдання може бути скоректована чи змінена, що відбивається в бланку завдання, який підписується як керівником, так і виконавцем роботи.

6.2. Виконання курсової роботи

Курсова робота повинна виконуватися переважно на ПК. Як виняток, дозволяється робота в з використанням інших методів і технологій (рукописний варіант, креслення на кульмані та ін.).

З огляду на обмежену пропускну здатність учбових лабораторій, необхідно, щоб робота всіх студентів була організована рівномірно. Починаючи з 2-3 тижня семестру, слід контролювати завантаженість лабораторій для того щоб не створювати перевантажень наприкінці семестру.

Відповідно до розкладу проводяться консультації (приблизно 3 години на тиждень), що можуть мати як індивідуальний, так і груповий характер.

Консультації не повинні перетворюватися в натаскування або репетиторство, а сприяти розвитку самостійного, продуманого й обґрунтованого, планомірного виконання роботи. Не слід давати готових рецептів, а постановкою навідних запитань, указівкою додаткової літератури допомогти студенту зрозуміти допущені помилки і знайти правильний шлях вирішення ускладнень. Щоб навчити студента пояснювати свою роботу, необхідно вимагати від нього доказів і обґрунтування найбільш важливих моментів роботи. У ході таких бесід студенти знайомляться з роботою своїх товаришів, допомагають один одному.

Для виключення штурмівщини керівник повинний переглянути виконання частини роботи: алгоритм, програму, рішення контрольних прикладів, дослідження ефективності програми. Після перевірки результатів одного етапу роботи викладач візує їх і дозволяє перейти до наступного. Рекомендується вивішувати графік ходу робіт, де варто відбивати виконання окремих етапів кожним студентом.

Оскільки окремі завдання можуть мати неоднакову складність, у процесі роботи можна вносити додаткові задачі: збереження вхідних даних і результатів на зовнішніх запам'ятовуючих пристроях, використання засобів операційної системи і т.п.

6.3. Захист курсової роботи

Після виконання всіх пунктів завдання пояснювальна записка, підписана студентом, схема програми (алгоритм) разом із програмою на носії (дискеті чи компакт-диску) здаються керівнику. Після перевірки працездатності програми і її ефективності в учбовій лабораторії, виконання усіх вимог до пояснювальної записки і графічної частини керівник допускає роботу до публічного захисту, про що робиться

запис у пояснювальній записці. При виявленні помилок чи зауважень робота повертається на доробку.

Прийом роботи проводиться комісією з 2-3 чоловік за участю керівника в присутності автора й інших студентів групи. Це дозволяє ознайомити групу з роботами її членів, а також виробити єдині вимоги до роботи в керівника.

Захист полягає в короткій доповіді (до 7 хв.) і відповідях на питання, що можуть задавати всі присутні.

За підсумками захисту виставляється оцінка, що повинна враховувати обґрунтованість обраного алгоритму, працездатність програми, її ефективність, стійкість до зміни вхідних даних, якість оформлення самої програми, пояснювальної записки і креслення, самостійність і обсяг проробленої роботи.

Студент, що не представив у встановлений термін курсову роботу чи не захистив її з неповажних причин, вважається таким, що має академічну заборгованість [16].

Після захисту рекомендується проводити заключну бесіду керівника й аналізувати кращі і погані роботи з указівкою типових помилок і недоліків, а також організаційних недоглядів.

Кращі роботи представляються на факультетський та інститутський конкурс.

Усі роботи здаються до архіву кафедри, де зберігаються на протязі трьох років, а їх електронні копії (на компакт дисках) – на протязі 5-ти років.

СПИСОК ЛІТЕРАТУРИ

1. Айвор Хортон Microsoft Visual C++ 2005: базовый курс = Beginning Visual C++ 2005. — М.: «Диалектика», 2007. —1152с.
2. Ахо, Альфред, В., Хопкрофт, Джон, Ульман, Джеффри, Д. Структуры данных и алгоритмы. — Издательский дом «Вильямс», 2000. —384с.
3. Баркакати Программирование игр для Windows на Borland C++. — М.: Бином, 1994.-512 с.
4. Бартків А. Б. та ін. Turbo Pascal: алгоритми і програми. — К.: Вища школа, 1992. — 247 с.
5. Бейко И.В., Бублик Б.Н., Зинько П.Н. Методы и алгоритмы решения задач оптимизации.- Высшая школа, 1983г. - 512 с.
6. Бендат Дж., Персол А. Прикладной анализ случайных данных. — М.: Мир, 1989. — 540 с.
7. Березин Н. С., Жидков Н. П. Методы вычислений, т. I. — М.: Наука, 1966. — 632 с.
8. Блейхут Р. Быстрые алгоритмы цифровой обработки сигналов. — М.: «Мир», 1989. — 448 с.
9. Бор К. Практическое руководство по сплайнам. — М.: Радио и связь, 1985. — 304с.
10. Брайан Фафенбергер. Толковый словарь по компьютерным технологиям и Internet. - К.: Диалектика, 1996.- 479с.
11. Брандт З. Статистические методы анализа наблюдений. — М.: Мир, 1975. — 312 с.
12. Василенко О.Н. Теоретико-числовые алгоритмы в криптографии. — Москва: МЦНМО, 2003. —328с.
13. Вентцель Е.С. Исследование операций: задачи, принципы, методология.- М.: Наука-1988.-208с.
14. Вирт Н. Алгоритмы+структуры данных=программы. М.: Мир,1985.-406 с.
15. Воеводин В. В., Кузнецов Ю. А. Матрицы и вычисления. — М.: Наука, 1984. — 320 с.
16. Высшая школа. Сборник основных постановлений, приказов и инструкций. Под ред. Войленко Е. И., ч. I. — М.: Высшая школа, 1978. — 399 с.
17. Гилл Ф. и др. Практическая оптимизация. — М.: Мир, 1985.- 252с.
18. Главацкая А. П. Методы и алгоритмы вычислительной математики. — М.: Радио и связь, 1999. — 408 с.
19. Горелик А.Л., Гуревич И.Б., Скрипкин В.А. Современное состояние проблемы распознавания: Некоторые аспекты. - М.: Радио и связь (Кибернетика), 1985.- 161с.
20. ГОСТ 19.001-77. ЕСПД. Общие положения.
21. ГОСТ 19.101-77. ЕСПД. Виды программ и програмных документов.
22. ГОСТ 19.103-77. ЕСПД. Обозначения программ и програмных документов.

23. ГОСТ 19.104-78. ЕСПД. Основные надписи.
24. ГОСТ 19.401-78. ЕСПД. Текст программы. Требования к содержанию и оформлению.
25. ГОСТ 19.402-78. ЕСПД. Описание программы.
26. ГОСТ 19.701-90. Схемы алгоритмов, программ, данных и системы. Обозначения условные и правила выполнения.
27. ГОСТ 2.004-88 ЕСКД. Общие требования к выполнению конструкторских и технологических документов на печатающих и графических устройствах вывода ЭВМ.
28. ГОСТ 2.104-68. ЕСКД. Основные надписи.
29. ГОСТ 2.105-95. ЕСКД. Общие требования к текстовым документам.
30. ГОСТ 2.106-96. ЕСКД. Текстовые документы.
31. ГОСТ 2.109-73. ЕСКД. Основные требования к чертежам.
32. ГОСТ 2.201-80. ЕСКД. Обозначение изделий и конструкторских документов.
33. ГОСТ 2.301-68. ЕСКД. Форматы.
34. ГОСТ 7.1-84. Система стандартов по информации, библиотечному и издательскому делу. Библиографическое описание документа. Общие требования и правила оформления.
35. ГОСТ 8.417-81.ГСИ. Величины физических величин.
36. Грегори К. Использование Visual C++ 5. – М.: Вильямс, 1999. – 864с.
37. ГСТУ 3-71-24-94. Специфікація програмних виробів. Вимоги змісту та оформлення.
38. Гутер Р. С., Резниковский П. Г. Программирование и вычислительная математика, ч. II. – М.: Наука, 1971.-433с.
39. Демидович Б. П., Марон И. А. Основы вычислительной математики. – М.: Наука, 1966. – 664 с.
40. Дональд Кнут Искусство программирования, том 1. Основные алгоритмы = The Art of Computer Programming, vol. 1. Fundamental Algorithms. — 3-е изд. — М.: «Вильямс», 2006. — С. 720с.
41. Дональд Кнут Искусство программирования, том 2. Получисленные методы = The Art of Computer Programming, vol. 2. Seminumerical Algorithms. — 3-е изд. — М.: «Вильямс», 2007. — С. 832с.
42. Дональд Кнут Искусство программирования, том 3. Сортировка и поиск = The Art of Computer Programming, vol. 3. Sorting and Searching. — 2-е изд. — М.: «Вильямс», 2007. — С. 824с.
43. Дональд Кнут Искусство программирования, том 4, выпуск 3. Генерация всех сочетаний и разбиений = The Art of Computer Programming, Volume 4, Fascicle 3 : Generating All Combinations and Partitions. — М.: «Вильямс», 2007. —208с.
44. Дональд Кнут Искусство программирования, том 4, выпуск 4. Генерация всех деревьев. История комбинаторной генерации = The Art of Computer Programming, Volume 4, Fascicle 4: Generating All Trees — History of Combinatorial Generation. — М.: «Вильямс», 2007. —160с.

45. ДСТУ 3008-95. Документація. Звіти у сфері науки і техніки. Структура і правила оформлення.
46. ДСТУ ISO 3166-1-2000. Коды назв країн світу.
47. Дьяконов В. П. MATLAB 6.5 SP1/7.0 + Simulink 5/6/ Обработка сигналов и проектирование фильтров.— М.: СОЛОН-Пресс, 2005. - 676с.
48. Дьяконов В. П. Справочник по алгоритмам и программам на языке БЕЙСИК для ПЭВМ. – М.: Наука, 1989.-240с.
49. Зайченко Ю.П. Исследование операций, 1988.-191с.
50. Зайченко Ю.П., Шумилова С.А. Исследование операций, сб. задач. 1988.-237с.
51. Зелинский К. Х., Игнатенко В. И., Коц А. П. Компьютерные методы прикладной математики, ч. I. – К., 1999. – 352 с.
52. Иванов В. В. Методы вычислений на ЭВМ. Справочник. – К.: Наукова думка, 1986.- 584 с
53. Калиткин Н. Н. Численные методы. – М.: Наука, 1978. – 512с.
54. Кнут Д. Искусство программирования для ЭВМ. Т. 3. Сортировка и поиск. М. Мир. 1978. - 844с.
55. Копчёнова Н. В., Марон И. А. Вычислительная математика в примерах и задачах. – М.: Наука, 1972. – 367 с.
56. Краскевич В.Е. Численные методы в инженерных расчетах. — Киев: Вища школа, 1986. •— 263 с.
57. Крейг Арнуш. Borland C++: освой самостоятельно. – М.: Бином, 1997.-543с.
58. Кудрявцев Е.М. Исследование операций в задачах, алгоритмах и программах.- Москва, "Радио и связь" 1984.-345с.
59. Ливингстон Брайан, Таррот Пол Секреты Microsoft Windows Vista = Windows Vista Secrets. — М.: «Диалектика», 2007. — С. 456.
60. Липский Комбинаторика для программистов . – М.: Мир, 1995. —200с..
61. Ляшенко М.Я., Головань М.С. Чисельні методи. - К.: Либідь, 1996. - 288 с.
62. Майника Э. Алгоритмы оптимизации на сетях и графах. – М.: Мир, 1981. – 323 с.
63. Мак-Кракел Д., Дорн У. Численные методы и программирование на Фортране. – М.: Мир, 1977. – 584 с.
64. Маликов В. Т., Кветный Р. Н. Вычислительные методы и применение ЭВМ. – К.: Вища школа, 1989. – 213 с.
65. Мудров А. Е. Численные методы для ПЭВМ на языках БЕЙСИК, ФОРТРАН и ПАСКАЛЬ. – Томск: Раско, 1991. – 272 с.
66. Муртаф Математическое программирование.
67. Ортега Дж., Пул У. Введение в численные методы решения дифференциальных уравнений. – М.: Наука, 1986. – 288 с.
68. Ортега Дж., Рейнболт. Итерационные методы решения нелинейных систем уравнений со многими неизвестными. – М.: Мир, 1975. - 278 с.

69. Паппас Крис, Мюррей Уильям Visual C++ 6. Руководство разработчика. – Киев.: BHV, 2000. - 624.
70. Поллард Дж. Справочник по вычислительным методам статистики. - М.: Финансы и статистика. - 1982. - 343 с.
71. Проценко В.С., Чаленко П.Й., Ставровський А.Б. Техніка програмування мовою Сі.-К.: Либідь, 1993.-224 с.
72. Пьюполо Джон. OLE: создание элементов управления. – Киев.: BHV, 1997. – 243с.
73. Рабинер Л., Гоулд Б. Теория и применение цифровой обработки сигналов. — М.: «Мир», 1978. — 848 с.
74. Рейнгольд Э., Нивергельт Ю. Комбинаторные алгоритмы. – Издательство: Мир, 1980. – 476с.
75. Самарский А.А. Введение в численные методы решения дифференциальных уравнений– М.: Наука, 1986. - 269 с.
76. Стандарты КПИ 2.001-83. Курсовые проекты. Требования к оформлению документации.
77. Тосс В. Энциклопедия Visual C++ 5. – АСТ, Астрель, 2008. – 320 с.
78. Фадеев Д.К. Вычислительные методы линейной алгебры, 1980
79. Форсайт Дж., Моулер К., Малкольм М. Машинные методы математических вычислений. – М.: Мир, 1986. – 276 с.
80. Хемминг Р.В. Численные методы для научных работников и инженеров. – М.: Наука, 1972. – 400 с.
81. Химмельблау Д. Прикладное нелинейное программирование. – М.: Мир, 1975. – 534 с.
82. Холзнер . Учебный курс Visual C++ 6.- Издательство: Питер, 2007. – 570с.
83. Шуп Т. Е. Прикладные численные методы в физике и технике. – М.: Высшая школа, 1990. – 225 с.
84. Янг Д. Visual C++ 6. – Киев.: BHV, 1999. – 560 с.

ДОДАТКИ

Додаток 1

Національний технічний університет України
“Київський політехнічний інститут”
кафедра Технічної кібернетики

Розробка програми обробки зображення

УА.ІК.91120-01 01

Курсова робота
дисципліна
“Об’єктно-орієнтоване програмування”

Керівник:
доц. Лісовиченко О.І.

Виконавець:
Миронюк О.П.

“Захист дозволено”

гр. ІК-91

“ ____ ” _____ 2010 р.

/підпис керівника/

/підпис/

Зхищено з оцінкою

Національний технічний університет України “КПІ”

(назва вищого навчального закладу)

Кафедра технічної кібернетики

Дисципліна об’єктно-орієнтоване програмування

Курс I Група ІК-91 Семестр 2

ЗАВДАННЯ

на курсову роботу студента

Миронюка Олександра Петровича

(прізвище, ім’я, по батькові)

1. Тема роботи Розробка програми обробки зображення

2. Строк здачі студентом закінченого проекту (роботи) 1.05.10

3. Вихідні дані до проекту (роботи) мова програмування
Borland C++

Технічна література

4. Зміст розрахунково – пояснювальної записки (перелік питань, які підлягають розробці)

Розробка алгоритму роботи програми

Архітектура програми

5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)

Блок-схема алгоритму роботи програми (Формат А1)

6. Дата видачі завдання 12.02.2010

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Найменування етапів курсової роботи	Строк вико- нання етапів роботи	Примітки
1.	Вивчення та аналіз алгоритму обробки зображення	19.02.2010	
2.	Визначення структури програмних модулів	26.02.2010	
3.	Розробка інтерфейсу	21.03.2010	
4.	Оптимізація програми	4.04.2010	
5.	Налагодження та тестування програми	10.04.2010	
6.	Оформлення роботи	14.04.2010	
7.	Здача на перевірку (електронна версія)	19.04.2010	
8.	Виправлення недоліків та доробка програми	24.04.2010	
9.	Захист	01.05.2010	

Студент _____ Миронюк О. П.
(підпис студента) (прізвище, ім'я, по батькові студента)

Керівник _____ Лісовиченко О.І.
(підпис викладача) (прізвище, ім'я, по батькові викладача)

“01” травня 2010р.

складність 3 - 4

№	Назва	Пояснення	Література
1.	Метод Гаусса розв'язання систем лінійних алгебраїчних рівнянь(СЛАР)	Розробка програми розв'язання СЛАР з можливістю роботи з файлами матриць та результатів розв'язку.	5, 6, 9, 10, 14, 15, 17
2.	Метод обертання розв'язання СЛАР	Розробка програми розв'язання СЛАР з можливістю роботи з файлами матриць та результатів розв'язку.	65
3.	Метод Холеського розв'язання СЛАР (L-U алгоритм)	Розробка програми розв'язання СЛАР з можливістю роботи з файлами матриць та результатів розв'язку.	5, 6
4.	Метод квадратного кореня розв'язання СЛАР	Розробка програми розв'язання СЛАР з можливістю роботи з файлами матриць та результатів розв'язку.	5, 6, 10
5.	Метод прогонки розв'язання СЛАР	Розробка програми розв'язання СЛАР з можливістю роботи з файлами матриць та результатів розв'язку.	7,17
6.	Уточнення коренів СЛАР	Розробка програми розв'язання СЛАР з можливістю роботи з файлами матриць та результатів розв'язку.	5, 6, 10
7.	Ітераційні методи розв'язання СЛАР	Розробка програми розв'язання СЛАР з можливістю роботи з файлами матриць та результатів розв'язку.	5, 6
8.	Обчислення зворотної матриці	Розробка програми обчислення зворотної матриці з можливістю роботи з файлами матриць та результатів розв'язку.	5, 6
9.	Обчислення визначника матриці	Розробка програми обчислення визначника матриці з можливістю роботи з файлами матриць та результатів розв'язку.	
10.	Обчислення власних значень та власних векторів матриць методами: Левер'є-Фадсєва	Розробка програми обчислення матриць з можливістю роботи з файлами матриць та результатів розв'язку.	13, 17
11.	Обчислення власних значень та власних векторів матриць методами: Данилевського-Якобі	Розробка програми обчислення власних значень та власних векторів матриці з можливістю роботи з файлами матриць та результатів розв'язку.	17, 18
12.	Обчислення власних значень та власних векторів матриць методами: Крилова	Розробка програми обчислення власних значень та власних векторів матриці з можливістю роботи з файлами матриць та результатів розв'язку.	18
13.	Обчислення власних значень та власних векторів матриць методами: L-R	Розробка програми обчислення власних значень та власних векторів матриці з можливістю роботи з файлами матриць та результатів розв'язку.	13, 15, 17, 47, 49, 51
14.	Обчислення власних значень та власних векторів матриць методами: Q-R	Розробка програми обчислення власних значень та власних векторів матриці з можливістю роботи з файлами матриць та результатів розв'язку.	13, 15, 17, 47, 49, 51
15.	Розв'язання алгебраїчних рівнянь з комплексними коренями	Розробка програми розв'язання алгебраїчних рівнянь з комплексними коренями з можливістю роботи з файлами (запис та зчитування рівнянь) та результатів розв'язку.	15, 17, 19, 39, 55, 63
16.	Розв'язання систем нелінійних рівнянь (СНР) методом Зейделя	Розробка програми розв'язання СНР з можливістю роботи з файлами (запис та зчитування рівнянь) та результатів розв'язку.	17, 65
17.	Розв'язання СНР методом Ньютона	Розробка програми розв'язання СНР з можливістю роботи з файлами (запис та зчитування рівнянь) та результатів розв'язку.	17, 51, 65
18.	Наближення функцій методом найменших квадратів	Розробка програми наближення функцій з можливістю роботи з файлами (запис та	19

№	Назва	Пояснення	Література
		зчитування функцій) та результатів розв'язку.	
19.	Наближення функцій сплайнами	Розробка програми наближення функцій з можливістю роботи з файлами (запис та зчитування функцій) та результатів розв'язку.	3, 17
20.	Чисельне інтегрування різними методами	Розробка програми реалізації одного методу чисельного інтегрування функцій з можливістю роботи з файлами (запис та зчитування функцій) та результатів розв'язку.	17, 19
21.	Розв'язання систем звичайних диференціальних рівнянь методами: Рунге-Кутта четвертого порядку	Розробка програми розв'язання системи звичайних диференціальних рівнянь з можливістю роботи з файлами (запис та зчитування рівнянь) та результатів розв'язку.	14, 17, 50, 65
22.	Розв'язання систем звичайних диференціальних рівнянь методами: Кутта-Мерсона з автоматичним вибором кроку	Розробка програми розв'язання системи звичайних диференціальних рівнянь з можливістю роботи з файлами (запис та зчитування рівнянь) та результатів розв'язку.	13, 65
23.	Розв'язання систем звичайних диференціальних рівнянь методами: Адамса	Розробка програми розв'язання системи звичайних диференціальних рівнянь з можливістю роботи з файлами (запис та зчитування рівнянь) та результатів розв'язку.	5, 6, 17, 18
24.	Розв'язання звичайних диференціальних рівнянь другого порядку методом Мілна	Розробка програми розв'язання системи звичайних диференціальних рівнянь з можливістю роботи з файлами (запис та зчитування рівнянь) та результатів розв'язку.	5, 6, 65
25.	Пошук екстремуму функції методами: Ньютона	Розробка програми пошуку екстремуму функції з можливістю роботи з файлами (запис та зчитування функції), а також результатів розв'язку.	17, 18, 56
26.	Пошук екстремуму функції методами: б) Найшвидшого спуску	Розробка програми пошуку екстремуму функції з можливістю роботи з файлами (запис та зчитування функції), а також результатів розв'язку.	17, 18, 56
27.	Пошук екстремуму функції методами: Сполучених градієнтів	Розробка програми пошуку екстремуму функції з можливістю роботи з файлами (запис та зчитування функції), а також результатів розв'язку.	17, 18, 56
28.	Пошук екстремуму функції методами: Флетчера-Рівса	Розробка програми пошуку екстремуму функції з можливістю роботи з файлами (запис та зчитування функції), а також результатів розв'язку.	17, 18, 56
29.	Пошук екстремуму функції методами: Деформованого багатогранника	Розробка програми пошуку екстремуму функції з можливістю роботи з файлами (запис та зчитування функції), а також результатів розв'язку.	56, 57
30.	Статистичний аналіз	Розробка програми реалізації одного з методів прикладного статистичного аналізу даних.	58, 59
31.	Розробка програм аналізу та генерації псевдовипадкових чисел	Розробка програми генерації псевдовипадкових чисел в залежності від вибраної функції розподілення з можливістю роботи з файлами (запис та зчитування функції), а також результатів генерації.	41,42,43,44
32.	Переведення числа в різні системи счислення (2-у, 8-у, 10-у, 16-у)	Розробка програми переведення числа в різні системи счислення з можливістю роботи з файлами (запис та зчитування масива чисел), а також результатів розв'язку.	41,42,43,44

складність 4- 5

№	Назва	Пояснення	Література
33.	Обчислення власних значень та власних векторів матриць методами: а) Левер'є-Фадєєва; б) Данилевського-Якобі в) Крилова г) L-R д) Q-R	Розробка пакету програм для обчислення власних значень та власних векторів матриць методами за вибором користувача. Виконати порівняння методів.	5, 6, 9, 10, 14, 15, 17, 65
34.	Розв'язання систем звичайних диференціальних рівнянь методами: а) Рунге-Кутта четвертого порядку; б) Кутта-Мерсона з автоматичним вибором кроку; в) Адамса	Розробка пакету програм для обчислення систем звичайних диференціальних рівнянь методами за вибором користувача. Виконати порівняння методів.	5, 6, 9, 10, 14, 15, 17, 65
35.	Пошук екстремуму функції методами: а) Ньютона б) Найшвидшого спуску в) Сполучених градієнтів г) Флетчера-Рівса д) Деформованого багатогранника	Розробка пакету програм для пошуку екстремуму функції методами за вибором користувача. Виконати порівняння методів.	5, 6, 9, 10, 14, 15, 17, 65
36.	Розробка драйверів нестандартних пристроїв	Розробка драйверів для нестандартних пристроїв з використанням різних каналів передачі даних	
37.	Розробка резидентних програм	Розробка резидентних програм різного призначення.	
38.	Трьохвимірна графіка	Розробка пакету програм створення 3D об'єктів з переліком їх властивостей.	
39.	Використання програмних модулів, написаних різними мовами програмування		
40.	Учбові та навчаючі програми	Розробка мультимедійного пакету програм для підвищення ефективності засвоєння інформації студентами під час навчання.	
41.	Задачі на графах	Розробка програми рішення пакету задач з використанням графів.	41,42,43,44, 61
42.	Управління базами даних	Розробка пакету програм підтримки електронної бази даних	
43.	Обробка зображень	Розробка пакету програм обробки зображень	
44.	Перетворення інформації для різних застосувань	Розробка пакету програм конвертації інформаційних ресурсів для різних застосувань	
45.	Захист інформації та архівування	Розробка пакету програм захисту (авторизація) та архівування (декілька алгоритмів) інформації	
46.	Підтримка роботи систем клієнт-сервер	Розробка програм передачі та обробки даних дистанційним способом (COM – технологія)	
47.	Мультимедійний довідник класів MFC (OWL) (анімація + гіпертекст)	Інтерактивний режим відображення дерева ієрархії класів OWL BC++ або MFC Visual C++, піддерев заданого рівня, структури заданого класу, приклади програм використання заданого класу.	
48.	Мультимедійний довідник по створенню анімації на C++ (OWL) (анімація + гіпертекст)	Інтерактивний режим відображення основних концепцій розробки анімаційних програм, алгоритмів і кодів програм, приклади реалізації анімацій.	
49.	Мультимедійний довідник прикладу задач і програм Visual	Розробка пакету програм і технології їх створення в середовищі Visual C++ як	

№	Назва	Пояснення	Література
	C++ (OWL) (анімація + гіпертекст)	самовчитель по Visual C++. Використовувати технологію візуального і невізуального проектування програм.	
50.	Створення Internet-магазину	Електронна база даних товарів і послуг, які можна отримати дистанційним способом через провайдерів internet (COM – технологія)	
51.	Мультимедійний конспект по програмуванню на мові Pascal (анімація + гіпертекст)	Гіпертекстовий документ, теоретичний, довідковий, практичний матеріал, що містить питання по програмуванню на мові Паскаль із запуском прикладів у вигляді файлів *.exe, анімацією.	
52.	Використання засобів мультимедіа для програмування видео- і звуко- ефектів	Інтерактивний режим відображення основних концепцій використання засобів мультимедіа для програмування видео- і звуко- ефектів, приклади реалізації.	
53.	Моделювання роботи засідань на прикладі Верховної Ради	Програма веде облік тривалості виступів депутатів і їх черговості, підрахунок голосів при голосуванні і видає результат голосування, видає документ заданої користувачем форми та ін.	
54.	Графічний редактор	Багатовіконне середовище відображення і формування графічних об'єктів	
55.	Моделювання роботи аеропорту	Програма-диспетчер, що управляє процесом зльоту і посадки літаків, реєструє аварійні ситуації, видає інформацію користувачу про проведені посадки, зльоти і стан польоту літаків.	
56.	Моделювання роботи залізничного вокзалу	Програма-диспетчер, що управляє процесом приходу та відправлення потягів, реєструє аварійні ситуації, видає інформацію користувачу про проведення посадки на потяги, стан потягів.	
57.	Програма тестування комп'ютерів	Багатовіконна оболонка для відображення складу і характеристик ПК – аналог Everest	
58.	Каталог літератури по сучасним мовам програмування	Багатовіконна оболонка для відображення переліку, здійснення гнучкого пошуку, обробки для статистики і т.і.	
59.	Каталог літератури по Hardware & software	Багатовіконна оболонка для відображення переліку, фірм-продавців, здійснення гнучкого пошуку, обробки для статистики, і т.і.	
60.	Програма моделювання виконання математичних операцій	Виконання векторних, матричних, арифметичних розрахунків, побудова графіків, діаграм та ін.	
61.	Англо-російський (український) перекладач	Перекладач-підрядковий переклад з узгодженням відмінків в пропозиціях. Об'єм словника від 2000 слів. Здійснює переклад окремих слів або цілих фраз. Запрошуються режими висновку перекладу (нове вікно або в старому вікні).	
62.	Программ перевірки правопису та граматики	Інтерактивна оболонка читання файлів, пошуку помилок в тексті і їх коректування та вибору зі словнику.	
63.	Довідник по методах і алгоритмах оптимізації функцій, що не диференціюються, і відшукування	Інтерактивна багатовіконна оболонка, яка демонструє вибір задач, введення початкових даних, вибір методу і результатів рішення.	

№	Назва	Пояснення	Література
	точок перегибу.		
64.	Программа роботи з матрицями	Матричні перетворення: складання, перемноження, ділення, піднесення до степеня, пошук зворотньої матриці та ін.	
65.	Программа побудови графіків по введеним функціям	Інтерактивна багатівіконна оболонка, яка демонструє вибір функції, введення початкових даних, вибір методу і виду побудови графіків.	
66.	Бібліотека програмних засобів рішення інтегральних і диференціальних рівнянь.	Інтерактивна багатівіконна оболонка рішення інтегральних і диференціальних рівнянь різними методами по вибору користувача (Ейлера, Рунге-Кутта і ін.).	
67.	Електронний записник	Інтерактивна багатівіконна оболонка з можливостями записника, телефонного довідника. Передбачити можливість шифрування та архівування.	
68.	Електронний щоденник робочого дня, тижня	Інтерактивна багатівіконна оболонка з можливостями ділового щоденника з відображенням переліку запланованих справ, виконаних заходів, ін. Передбачити можливість шифрування та архівування.	
69.	Бібліотека програмних засобів рішення систем лінійних рівнянь	Інтерактивна багатівіконна оболонка рішення систем лінійних рівнянь будь-якої розмірності, різними методами по вибору користувача (Гауса, Зейделя, Крамера і ін.). Передбачити довільну розмірність задачі.	
70.	Мультимедійний довідник по розробці Web-сторінок	Розробка пакету програм і технології їх створення в середовищі Visual C++ як самовчитель по розробці Web-сторінок. Використовувати технологію візуального і невізуального проектування програм.	
71.	Программа побудови розкладу занять на факультеті	Інтерактивна багатівіконна оболонка побудови розкладу занять на факультеті з можливістю пріоритетів користувача та створення звітів в Excel.	
72.	Захист інформації та шифрування	Інтерактивна багатівіконна оболонка з можливістю шифрування інформації та її захист. Передбачити можливість перевірки цілісності об'єкту.	
73.	СУБД	Інтерактивна багатівіконна оболонка з можливостями організації БД та роботи з нею, використовуючи мову програмування Visual C++. Передбачити можливість шифрування та архівування	
74.	Створення програми роботи з системою клієнт-сервер	Інтерактивна багатівіконна оболонка з можливостями організації роботи з системою клієнт-сервер Visual C++. Передбачити можливість шифрування та паролей.	
75.	Створення програми роботи з тривимірною графікою	Інтерактивна багатівіконна оболонка з можливостями роботи з тривимірною графікою, використовуючи мову програмування Visual C++. Передбачити можливість доповнення графічних примітивів.	
76.	Створення програми аналізу та генерації псевдовипадкових чисел	Інтерактивна багатівіконна оболонка з можливостями аналізу та генерації псевдовипадкових чисел, використовуючи мову програмування Visual C++. Передбачити можливість вибору та доповнення методів.	