

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЗАПОРІЗЬКА ПОЛІТЕХНІКА»

МЕТОДИЧНІ ВКАЗІВКИ
до лабораторних робіт
з дисципліни
ТЕХНОЛОГІЇ ПРОГРАМУВАННЯ
для студентів спеціальності
152 „Метрологія та інформаційно-вимірювальна техніка“
денної й заочної форм навчання

Методичні вказівки до лабораторних робіт з дисципліни „Технології програмування“ для студентів спеціальності 152 „Метрологія та інформаційно-вимірювальна техніка“ денної й заочної форм навчання / Укл.: В.І. Рева. – Запоріжжя: НУ «Запорізька політехніка», 2021. – 40 с.

Укладач: В.І. Рева, доц., канд. фіз.-мат. наук

Рецензент: О.В. Василенко, доц., канд.техн.наук

Відповідальний за випуск: А.В. Коротун, доц., канд.фіз.-мат.наук

Затверджено
на засіданні кафедри
мікро- та наноелектроніки
Протокол № 7
від “ 26 “ листопада 2020 р.

Рекомендовано до видання
НМК ФРЕТ
Протокол № 4
від “ 10 “ грудня 2020 р.

ЗМІСТ

ВСТУП	5
ПІДГОТОВКА ДО ПРОВЕДЕННЯ ЛАБОРАТОРНОЇ РОБОТИ ТА ОФОРМЛЕННЯ ЗВІТУ	6
1 ЛАБОРАТОРНА РОБОТА № 1 «ПОПЕРЕДНЄ ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ»	7
1.1 Порядок виконання роботи і звітність	7
1.2 Теоретичні відомості	7
1.3 Завдання	8
1.4 Контрольні запитання	9
2 ЛАБОРАТОРНА РОБОТА № 2 «РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ»	10
2.1 Порядок виконання роботи і звітність	10
2.2 Теоретичні відомості	10
2.3 Контрольні питання	11
3 ЛАБОРАТОРНА РОБОТА № 3 «ПОБУДОВА ФУНКЦІОНАЛЬНОЇ СХЕМИ СИСТЕМИ ПЗ»	12
3.1 Порядок виконання роботи і звітність	12
3.2 Теоретичні відомості	12
3.3 Контрольні запитання	13
4 ЛАБОРАТОРНА РОБОТА № 4 «ЗОВНІШНЄ ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ»	13
4.1 Порядок виконання роботи і звітність	13
4.2 Теоретичні відомості	13
4.2.1 Основні правила організації діалогу програми з користувачем	14
4.3 Контрольні запитання	16
5 ЛАБОРАТОРНА РОБОТА № 5 «РОЗРОБКА АРХІТЕКТУРИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ»	16
5.1 Порядок виконання роботи і звітність	16
5.2 Теоретичні відомості	16
5.2.1 Загальні правила структурної побудови програмних модулів	17
5.2.2 Правила зв'язку програмних модулів за управлінням	18
5.2.3 Правила зв'язку програмних модулів за інформацією	18
5.3 Контрольні запитання	19
6 ЛАБОРАТОРНА РОБОТА № 6 «ОПИС АЛГОРИТМУ»	20

6.1 Порядок виконання роботи і звітність	20
6.2 Теоретичні відомості	20
6.3 Контрольні запитання	21
7 ЛАБОРАТОРНА РОБОТА № 7 «ЗАПИС ТЕКСТІВ ПРОГРАМ НА АЛГОРИТМІЧНІЙ МОВІ ВИСОКОГО РІВНЯ»	21
7.1 Порядок виконання роботи і звітність	21
7.2 Теоретичні відомості	22
7.3 Загальна організація програми і її запис	22
7.4 Контрольні запитання	23
8 ЛАБОРАТОРНА РОБОТА № 8 «ПОКРОКОВА РОЗРОБКА ПРОГРАМИ»	23
8.1 Порядок виконання роботи і звітність	23
8.2 Теоретичні відомості	23
8.3 Контрольні питання	24
9 ЛАБОРАТОРНА РОБОТА № 9 «ТЕСТУВАННЯ ТА НАЛАГОДЖЕННЯ РОЗРОБЛЕНОЇ ПРОГРАМИ»	24
9.1 Порядок виконання роботи і звітність	24
9.2 Теоретичні відомості	24
9.3 Контрольні питання	26
10 ЛАБОРАТОРНА РОБОТА № 10 «СКЛАДАННЯ ДОКУМЕНТА «КЕРІВНИЦТВО КОРИСТУВАЧА»»	27
10.1 Порядок виконання роботи і звітність	27
10.2 Теоретичні відомості	27
10.3 Контрольні запитання	29
Додаток А Приклад оформлення титульної сторінки	30
Додаток Б Варіанти задач	31
Додаток В Додаткові питання	39

ВСТУП

Помилки у задумі і проектуванні програм та неякісна документація призводять до вельми відчутного марнування часу та ресурсів. З цього приводу існує красномовна американська статистика «якщо розміру великої програми поставити у відповідність час, необхідний для її розробки, і число зайнятих в ній програмістів, то виявиться, що все відбувається так, немов кожен програміст створює 5 команд в день, а решту часу пише команди, які потім визнаються марними або неправильними, і розшукує зроблені помилки».

Software Engineering Institute (SEI) в Університеті Карнегі-Меллон у Піттсбурзі встановив деякі градаційні рамки, які дозволяють кожному користувачеві і виробнику віднести себе до однієї з п'яти категорій по відношенню до проектування та розробки програмного забезпечення. При обстеженні ряду приватних фірм і державних установ отримані наступні результати:

1. **Хаотичний.** Погане управління порядком. Відсутність управління операціями. Висока собівартість проектів і проблеми з плануванням. Відсутність управління технічною стороною реалізації проектів, невикористання нових засобів і технологій. Під цю категорію підпадають від 74% до 86% всіх розробників програмного забезпечення.
2. **Повторюваний.** Переоцінка вартості, планування, зміна вимог, огляд стану справ та інше повторюються від проекту до проекту. Використовуються стандартні методи розробки програмного забезпечення. Вартість і планування проектів під контролем, В даній категорії від 22% до 23% всіх розробників.
3. **Визначений.** Процес розробки визначено в термінах технічного стандарту розробки програмного забезпечення, включаючи проектування, ліцензування коду і навчання. Тільки від 1% до 4% розробників досягли даного рівня.
4. **Керований.** Процес визначений, оцінений і добре керований. Використовуються спеціальні засоби для контролю і управління процесом розробки і для підтримки збору та аналізу даних. Ведеться всеосяжний аналіз даних про проект, зібраних за допомогою оглядів і тестування.

Практично 0% розробників досягли даного рівня компетенції.

5. **Оптимізований.** Досягнутий високий ступінь управління процесом. Організація концентрує зусилля на оптимізації окремих операцій. Вичерпний аналіз допущених і запобігання можливих помилок постійно веде до вдосконалення процесу. Даного рівня досягли 0%.

ПІДГОТОВКА ДО ПРОВЕДЕННЯ ЛАБОРАТОРНОЇ РОБОТИ ТА ОФОРМЛЕННЯ ЗВІТУ

Лабораторна робота включає самостійне опрацювання теоретичного матеріалу, вивчення документації з апаратних засобів, методик проведення розрахунків, програмування і моделювання, обробку й інтерпретацію результатів. При проведенні лабораторного практикуму необхідно:

- підготуватися до експрес-опитування з теоретичного матеріалу, необхідного для виконання роботи (щоб отримати допуск до лабораторної роботи);
- підготувати і оформити план виконання лабораторної роботи;
- після виконання лабораторної роботи провести необхідні розрахунки і оформити звіт, який повинен містити:

- 1) титульний аркуш (див. Додаток А);
- 2) мету роботи;
- 3) опис словами;
- 4) алгоритм роботи у вигляді блок схеми;
- 5) необхідні математичні розрахунки роботи пристрою;
- 6) текст розробленої програми;
- 7) основні результати моделювання пристрою;
- 8) стислі висновки по роботі, в яких:
 - вказати, що досліджувалось;
 - вказати методику (чи метод) дослідження;
 - вказати особливості алгоритму роботи пристрою;
 - визначити характер роботи пристрою при моделюванні;

- відзначити подібність і відмінність результатів моделювання від теоретичних даних;
- здати звіт з виконаної лабораторної роботи та відповісти на запитання викладача (отримати залік за виконану роботу).

1 ЛАБОРАТОРНА РОБОТА № 1

«ПОПЕРЕДНЄ ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ»

Мета роботи: Проведення попереднього проектування конкретної програми; складання переліку вимог і функціональних характеристик розроблюваної програми; розробка документу «Постановка завдання».

1.1 Порядок виконання роботи і звітність

Під час виконання лабораторної роботи необхідно визначити потребу в програмному виробі, його призначення та основні функціональні характеристики; скласти перелік вимог до нього.

Робота повинна бути оформлена у вигляді документа «Постановка завдання».

1.2 Теоретичні відомості

Визначення повного комплексу вимог до програмного виробу є первинним завданням його розробки. Неякісне визначення вимог призводить до створення програмного виробу, яке буде правильно вирішувати невірні сформульоване завдання, а програмний продукт не буде відповідати справжнім потребам замовника.

Тому при визначенні вимог до програмного виробу потрібно дотримуватися максимально можливої акуратності і точності, щоб потім ці вимоги можна було транслювати в проект, що розробляється

з мінімальним числом помилок. Вимоги задаються природною (для цієї задачі) мовою і повинні бути дуже точно сформульовані.

Вимоги оформлюються в вигляді документа, в якому письмово викладається те, що буде, і що не буде зроблено при випуску програмного виробу. У навчальному закладі такий документ називається "Постановка завдання".

1.3 Завдання

Постановка завдання пишеться на природній мові в термінах зрозумілих і користувачеві і розробнику програмного забезпечення і може містити наступні розділи:

1. Заголовок програми.

2. Умови завдання.

Формулюється умова завдання, короткий опис програми, що розробляється, її призначення та необхідні уточнення.

3. Початок/закінчення роботи.

Вказується місяць і рік початку/закінчення розробки програми.

4. Підстава для розробки програми.

Підставою для розробки програми може бути замовлення користувача, завдання адміністрації навчального закладу, контракт навчального закладу з іншою організацією та ін.

5. Коротка характеристика об'єкта розробки.

Описується об'єкт розробки: як вирішується поставлене завдання в даний час без програми, що розробляється і яка частина ручної роботи буде замінена програмою.

6. Користувач.

Вказуються користувачі програми.

7. Мета і призначення розробки.

8. Основні вимоги.

Описуються вимоги користувача до розроблюваної програми.

Тут же з точки зору користувача слід детально перерахувати функції програми.

9. Вхідна інформація.

Перераховуються всі вхідні дані програми з точки зору їх змісту і призначення - звіти, файли, записи, поля даних, таблиці ... Їх можливі носії і засоби відображення інформації і т.д.

10. Вихідна інформація.

Описуються вихідні дані так само, як в пункті 9.

11. Вимоги до апаратного та програмного забезпечення.

Описується конфігурація апаратури і програмного забезпечення, в яких розробляється програма може працювати, інші програмні продукти, від яких вона залежить.

12. Зовнішні обмеження.**13. Ефективність.**

Цілі продуктивності, такі, як часові і об'ємні характеристики, пропускна здатність, використання ресурсів тощо.

14. Безпека даних від несанкціонованого доступу.**15. Ергономічні характеристики.**

Ергономічними характеристиками виробу є такі властивості, які забезпечують надійність, комфорт і продуктивність роботи користувачів і операторів. Ергономіка (грец.) - праця + закон - галузь знання, що вивчає трудові процеси з метою створення найкращих умов праці.

16. Мобільність.

Описуються вимоги і цілі забезпечення перенесення програмного продукту з одних робочих умов в інші.

17. Окупність капіталовкладень.

Визначається прибуток, яку дасть створення програмного продукту в поняттях, відповідних цільовим призначенням організації.

18. Інші угоди сторін.**19. Термінологія.**

Чітко визначається вся термінологія, яка може виявитися специфічною для даної розробки.

1.4 Контрольні запитання

Попереднє проектування ПЗ

1. Предмет "Технологія програмування", основні поняття.
2. Попереднє проектування (Системний аналіз).
3. Постановка цілей проекту і продукту.
4. Визначення вимог.
5. Документ "Постановка завдання". ("Угода про вимоги")

2 ЛАБОРАТОРНА РОБОТА № 2 «РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ»

Мета роботи: Визначення етапів розробки конкретної програми; розробка календарного плану створення конкретної програми.

2.1 Порядок виконання роботи і звітність

Під час виконання лабораторної роботи необхідно детально проаналізувати етапи розробки конкретної програми (її життєвий цикл), починаючи від виникнення потреби в ній до повного припинення її використання внаслідок її морального старіння або втрати необхідності вирішення відповідних завдань.

Робота повинна бути оформлена у вигляді Табл. 2.1.

Таблиця 2.1 – Календарного плану розробки програми

№	Найменування етапу розробки програми	Термін виконання		Примітки
		Початок	Кінець	

2.2 Теоретичні відомості

Узагальнена модель життєвого циклу програми може виглядати так:

1. Системний аналіз (попереднє проектування ПП)
 - а) Дослідження
 - б) Здійсненність
 - Експлуатаційна
 - Економічна
 - Комерційна
2. Проектування програми
 - а) конструювання програми
 - функціональна декомпозиція задач
 - розробка архітектури системи
 - зовнішнє проектування програми

- розробка архітектури програми
- проектування бази даних
- б) програмування
 - внутрішнє проектування форм і модулів
 - визначення властивостей об'єктів і кодування
 - налагодження форм і модулів
 - компоновка форм і модулів в програму
- в) налагодження програми в цілому
- 3. Оцінка (випробування) програми
- 4. Використання програмного виробу

2.3 Контрольні питання

Розробка програмного забезпечення.

1. Технологія розробки програмного забезпечення.
2. Вимоги, що пред'являються до «ідеальної» технології розробки програмного забезпечення.
3. Об'єктно - орієнтовані технології розробки ПЗ.
4. Програмне забезпечення як виріб.
5. Проектування ПЗ. Огляд етапів проектування ПЗ (життєвий цикл).

3 ЛАБОРАТОРНА РОБОТА № 3

«ПОБУДОВА ФУНКЦІОНАЛЬНОЇ СХЕМИ СИСТЕМИ ПЗ»

Мета роботи: проведення функціональної декомпозиції розв'язуваної задачі; побудова функціональної схеми.

3.1 Порядок виконання роботи і звітність

Під час виконання лабораторної роботи необхідно провести функціональну декомпозицію розв'язуваної задачі, побудувати відповідну схему. Робота повинна бути оформлена у вигляді специфікації, що містить функціональну схему розв'язуваної задачі.

3.2 Теоретичні відомості

Проектування програмного забезпечення часто починається з функціональної декомпозиції розв'язуваної задачі.

Функціональна декомпозиція задачі являє собою ієрархічне розбиття складної задачі на ряд простіше вирішуються невеликих завдань, які, в свою чергу, поділяються на підзадачі до тих пір, поки кожна необхідна деталь в ній не буде визначена досить ясно.

Концепція ієрархічної декомпозиції настільки природна, що ми не завжди в змозі усвідомити, як часто нам доводиться використовувати її на практиці. Вона впливає з людської потреби мати справу з цілком визначеним числом дискретних джерел інформації, що піддається управлінню і проводити «відсікання» інформації до тих пір, поки число дискретних джерел не стане приблизно рівним семи.

Суворі ієрархічна декомпозиція підпорядковується правилам:

1. На кожному рівні ієрархії завдання повинна мати закінчений вигляд на даному рівні деталізації;
2. На будь-якому рівні ієрархії кожне розбиття повністю охоплює окреме завдання (функцію), яка відповідає певному рівню деталізації.

3.3 Контрольні запитання

Побудова функціональної схеми системи ПЗ.

1. Методи проектування програмного забезпечення.
2. Висхідне і нисхідне проектування програмного забезпечення.
3. Об'єктно-орієнтоване проектування програмного забезпечення.
4. Функціональна декомпозиція системи програмного забезпечення.
5. Надійність програмного забезпечення.

4 ЛАБОРАТОРНА РОБОТА № 4 «ЗОВНІШНЄ ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ»

Мета роботи: Проведення зовнішнього проектування конкретної програми; розробка взаємодії з користувачем, програми що розробляється: сценарій, екранні форми, набір підказок, тощо.

4.1 Порядок виконання роботи і звітність

Під час виконання лабораторної роботи необхідно описати очікувану поведінку розроблюваної програми з точки зору зовнішнього по відношенню до нього спостерігача (зазвичай - користувача), тобто здійснити "побудову" зовнішніх взаємодій майбутнього програмного продукту з користувачем без конкретизації його внутрішнього устрою.

Робота повинна бути оформлена у вигляді зовнішньої специфікації.

4.2 Теоретичні відомості

Зовнішнє проектування мало, чим пов'язане (якщо пов'язано взагалі) з програмуванням; конкретніше воно стосується розуміння

обстановки, проблем і потреб користувача, психології спілкування людини з машиною. Ця сторона зовнішнього проектування стає все більш значною у міру того, як застосування ЕОМ все більше починає зачіпати користувачів, не знайомих із програмуванням.

Результати зовнішнього проектування програми відображаються у зовнішній специфікації, в якій може бути представлено опис наступних зовнішніх аспектів програми:

- організація діалогу програми з користувачем;
- склад меню, підменю ...;
- опис дій функціональних клавіш;
- усі екранні форми або протокольні екранні повідомлення;
- повідомлення, що видаються користувачеві під час проведення сеансу роботи програми і відповіді, що видаються користувачем на них;
- повідомлення про помилки;
- підказки користувачеві, організація "допомоги";
- структура і організація баз даних;
- опис і підготовка вхідних даних;
- вихідні друковані форми;
- інші зовнішні спряження програми.

Зовнішня специфікація повинна бути написана зрозумілою користувачеві і розробнику мовою для зменшення ймовірності можливих непорозумінь. Причому, перевірку коректності та повноти специфікації необхідно проводити ще до початку програмування.

4.2.1 Основні правила організації діалогу програми з користувачем:

1. Узгоджуйте спосіб взаємодії програми з користувачем, з його підготовкою та рівнем, з обмеженнями, в умовах яких він працює.

2. Вихідні дані повинні видаватися програмою в необхідній формі та обов'язково з коментарями. Не можна, наприклад, видавати їх у вигляді числа, а тим більше - у вигляді набору чисел.

3. Забезпечте концептуальну цілісність для різних типів ввідних/вивідних повідомлень. Наприклад, всі повідомлення виводу на екран, звіти повинні мати однакові формати, стиль і скорочення.

4. Намагайтеся, щоб користувач вводив дані з клавіатури як можна менше. Буде краще, якщо йому буде дана можливість вибору

даних, що вводяться в вигляді меню, що виключить помилки введення користувачем. Повідомлення, що вводяться користувачем, повинне бути якомога коротше, але не настільки, щоб зникла їх осмисленість.

5. Забезпечте засоби "допомоги" - спеціальний набір функцій (підказки) з надання допомоги споживачеві, якщо той заплутався або забуде якесь правило взаємодії.

6. Намагайтеся, щоб програма не дратувала користувача. Уникайте образливих повідомлень. Спілкуйтеся з користувачем на його мові, а не на жаргоні програмістів.

7. Пам'ятайте про дизайн екрану. З естетично оформленим екраном приємніше працювати. Екранна форма може бути різноманітною.

8. Намагайтеся на кожне вхідне повідомлення видавати будь-яке повідомлення. Програма повинна приймати будь-які дані, що вводяться. Якщо дані не є тим, що програма вважає допустимим, то вона повинна інформувати про це користувача.

9. Спроектуйте програму так, щоб користувач в будь-який момент роботи з нею міг закінчити цю роботу або перейти в попередній стан. Передбачається, що в першому випадку програма успішно завершить свою роботу (закриє відкриті файли, очистить змінні пам'яті і т.д.)

10. Помилки користувача повинні виявлятися негайно.

11. Не прагніть виправляти вхідне повідомлення користувача.

Наприклад, у медичній інформаційній системі користувач випадково натискає на зайву клавішу, внаслідок чого вхідний повідомлення набуває вигляду "Ретіловий спирт" замість повідомлення "Етиловий спирт". Система виправляє це повідомлення на "Метилловий спирт". Відомо, що етиловий спирт п'янить, а метиловий спирт вбиває.

12. Будь-які дії користувача, як правильні, так і неправильні, повинні контролюватися програмою. Як негативний приклад можна навести програму, яка може раптом аварійно, передчасно закінчити свою роботу.

4.3 Контрольні запитання

Зовнішнє проектування ПЗ.

1. Методична, технологічна, інструментальна і організаційна підтримка процесу проектування ПЗ.
2. Конструювання програмного забезпечення.
3. Зовнішнє проектування ПЗ.
4. Правила організації діалогу ПЗ з користувачем.
5. Створення інтерфейсу програми.

5 ЛАБОРАТОРНА РОБОТА № 5 «РОЗРОБКА АРХІТЕКТУРИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ»

Мета роботи: Розробити архітектуру програмного виробу.

5.1 Порядок виконання роботи і звітність

Під час виконання лабораторної роботи необхідно розробити архітектуру розроблюваної програми: спроектувати структуру всіх його компонент, її об'єктно - модульно - ієрархічна побудова.

Робота повинна бути оформлена у вигляді специфікації, що містить архітектуру розроблюваної програми.

5.2 Теоретичні відомості

Типова архітектура програми:

1. Рівень – центральний диспетчер
2. Рівень – місцеві диспетчери
3. Рівень – функціональні програми
4. Рівень – функціональні програми
5. Рівень – стандартні програми, бібліотеки програм

5.2.1 Загальні правила структурної побудови програмних модулів

1. Кожен модуль характеризується функціональною закінченістю, автономністю і незалежністю в оформленні від модулів, які його використовують і які він викликає. Високий ступінь незалежності модулів можна досягти за допомогою двох методів оптимізації:

- посиленням внутрішніх зв'язків в кожному модулі, тобто реалізувати окремі функції окремими модулями (висока міцність модуля).

- ослабленням взаємозв'язку між модулями, застосовуючи формальний механізм передачі параметрів (слабке зчеплення модулів).

2. Застосовуються стандартні правила організації зв'язків з управління та інформації з іншими модулями (дивись далі).

3. Комплекси програм розробляються у вигляді сукупності невеликих за кількістю (до 100) програмних модулів, пов'язаних ієрархічним чином, що дає можливість повністю і відносно просто усвідомити функцію і правила роботи окремих частин і комплексу програм в цілому.

4. Як правило, модуль містить від 10 до 1000 виконуваних операторів мови високого рівня. Розміри модуля впливають на ступінь незалежності програми, легкість її читання і тестування.

5. Модуль міцний. Міцність модуля вимірюється його внутрішніми зв'язками. Модуль - це замкнута програма, яка виконує одну або кілька функцій, володіє деякою логікою.

6. Модуль передбачуваний, тобто модуль, робота якого не залежить від передісторії його використання. Модулі не повинні зберігати ніяких "спогадів" про свій попередній виклик.

7. Визначено структуру прийняття рішень. Бажано, щоб ті модулі, на які прямо впливає прийняте рішення, були підлеглими (викликаються) по відношенню до приймаючого рішення модулю.

8. Обсяг даних, на які модуль може посылатися, повинен бути зведений до мінімуму.

9. Внутрішня процедура (або підпрограма) - це замкнута програма, фізично розташована в модулі, що її викликав. Їх слід уникати, тому що їх важко ізолювати для автономного тестування і вони не можуть бути викликані з модулів, відмінних від тих, які їх

фізично містять. Коли виникає потреба у внутрішній процедурі, проектувальник повинен розглянути можливість оформлення її у вигляді окремого модуля.

10. У параметри процедури слід включати тільки ті змінні, через які йде обмін інформацією з іншими програмними одиницями. Інші змінні - це внутрішня справа процедури. Процедури, які видають в якості результату тільки одне значення, оформляються як функції. Функція зручніше у використанні, так як її результат безпосередньо можна використовувати в арифметичному і / або в логічному виразах.

5.2.2 Правила зв'язку програмних модулів за управлінням

1. Передача управління викликаному модулю завжди здійснюється через його початок, тобто через його перший оператор.

2. Вихід з викликаного модуля завжди відбувається через його природне закінчення, тобто після нормального його завершення.

3. Після закінчення виконання викликаного модуля управління передається в викликаючий модуль, на оператор, наступний за оператором виклику.

4. Модулі нижчих рівнів або одного рівня ієрархії можуть викликатися для виконання тільки модулями вищих рівнів, тобто модулі нижчих рівнів не можуть викликати модулі вищих рівнів, а модулі одного рівня - викликати один одного.

5. Якщо все ж необхідно виконати модуль з деякою внутрішньою точкою, то виклик все одно здійснюється стандартним чином (через його перший оператор), а точка початку задається у вигляді параметра. При цьому на початку викликаємого модуля повинен стояти перемикач, який забезпечує передачу управління програмою до його внутрішніх точок по параметру, вказаному при зверненні до модуля.

6. У кожному модулі повинна бути передбачена можливість підключення контрольних і налагоджувальних засобів; оператори, що реалізують ці засоби, зазвичай зосереджуються в кінці модуля.

5.2.3 Правила зв'язку програмних модулів за інформацією

1. Інформація зон глобальних змінних доступна для використання будь-яким модулем, що входять в комплекс програм або

в групу програм відповідно до області дії зони глобальних змінних, тобто глобальні змінні, можуть бути доступні не для всього комплексу програм, а лише для зазначеної в описі групи модулів.

2. Локальні змінні доступні лише в межах того модуля, в якому вони визначені або оголошені.

3. Для взаємодії викликаючих і викликаємих модулів створюються зони обмінних змінних, інформація з яких доступна лише модулям, що безпосередньо пов'язані по управлінню.

4. Після закінчення роботи викликаємого модуля вважається, що відповідні регістри не містять інформації, що є результатом його роботи. Забороняється їх використовувати в викликаємому модулі.

5. Інформація, що знаходиться в регістрах викликаючого модулю, при виклику повинна бути збережена на період виконання викликаємого модулю і відновлена при поверненні управління в викликаючий модуль. Збереження регістрів може здійснювати як викликаючий так і викликаємий модуль.

5.3 Контрольні запитання

Розробка архітектури ПЗ.

1. Конструювання архітектури ПЗ.
2. Типова архітектура програмного забезпечення.
3. Загальні правила структурної побудови ПЗ.
4. Правила зв'язку програмних модулів з управлінням.
5. Правила зв'язку програмних модулів за інформацією.

6 ЛАБОРАТОРНА РОБОТА № 6

«ОПИС АЛГОРИТМУ»

Мета роботи: Розробка алгоритму розв'язання задачі; запис алгоритму в блок схемної формі.

6.1 Порядок виконання роботи і звітність

Під час проведення лабораторної роботи необхідно розробити алгоритм вирішення конкретного завдання і представити його в блок-схемної формі

Робота повинна бути оформлена у вигляді специфікації, що містить опис алгоритму конкретної програми.

6.2 Теоретичні відомості

Представлений в будь-якій формі алгоритм пишеться на природній (розмовній) мові, використовуючи терміни галузі, в якій вирішується завдання і не повинен містити програмістські терміни; тим більше алгоритм не може містити оператори мови програмування. Текст алгоритму повинен бути читабельним без додаткових пояснень автора.

В алгоритмі не показують опис, тому що опис - це особливості мови програмування, а не алгоритму. Алгоритм не може бути "прив'язаний" до конкретної мови програмування. В алгоритмі описуються конкретні дії - "що робиться", а не "як це робиться".

У блок-схемі зазвичай пишуть "скупі" тексти. Тому її доповнюють поясненнями. У поясненні до схеми описують функціональні дії всього алгоритму в цілому, дії окремих частин схеми і в разі крайньої необхідності - дія окремого блоку. Описувати кожний блок схеми безглуздо. Пояснення повинні містити деяку додаткову інформацію, а не перефразування алгоритму.

Основні вимоги до блок-схеми:

- ❖ схема виконується з дотриманням умов ДУСТу;
- ❖ запис в блоках повинен бути словесним або математичним, а не у вигляді операторів мови програмування;

- ❖ управління за схемою має в основному йти вниз (вправо), повертаючись назад тільки в циклах;
- ❖ альтернативно виконувані гілки повинні розміщуватися паралельно;
- ❖ змінні повинні бути визначені в будь-якому блоці;
- ❖ вхідні та вихідні блоки процедур повинні містити, відповідно, вхідні і вихідні (формальні) параметри;
- ❖ блоки можна об'єднувати в більші пунктирними лініями, які потрібно коментувати - описати їх призначення;
- ❖ на кожному підпрограмі (модуль) складається окремий схема.

6.3 Контрольні запитання

Опис алгоритму.

1. Програмування. Проектування програмного модуля.
2. Планування програмування.
3. Алгоритм. Форми опису алгоритмів.
4. Основні вимоги до блок-схеми.
5. Проектування логіки програмного модуля.

7 ЛАБОРАТОРНА РОБОТА № 7 «ЗАПИС ТЕКСТІВ ПРОГРАМ НА АЛГОРИТМІЧНІЙ МОВІ ВИСОКОГО РІВНЯ»

Мета роботи: Запис текстів програм на алгоритмічній мові високого рівня, використовуючи правила хорошого стилю програмування.

7.1 Порядок виконання роботи і звітність

Під час проведення лабораторної роботи необхідно написати програму на конкретній алгоритмічній мові програмування для вирішення конкретного завдання, використовуючи прийоми і методи

програмування, які використовують, досвідчені програмісти, щоб отримати правильні, надійні, ефективні, зручні для застосування і легко читаємі програми.

У звіті повинні бути представлені листинги текстів програм.

7.2 Теоретичні відомості

Програма повинна бути складена таким чином, щоб її могли прочитати в першу чергу люди, а не машини.

Програма — це документ для подальшого використання, навчальний матеріал по кодування алгоритмів і засіб для подальшої розробки досконаліших програм.

Як правило, до розробки програми приступають зі скромними цілями. Надалі її можливості розширюють. Програми, що важко читаються модифікувати складно. Особливо, якщо це доводиться робити не автору програми.

Програма повинна передавати логіку і структуру алгоритму настільки, наскільки це можливо. Її кодують просто і раціонально. Слід уникати всіляких програмістських трюків, тому що чим їх більше, тим важче буде розібратися в логіці програми самому автору, а будь-хто інший це зробити не зможе. На ранніх етапах розробки складної програми краще без коливань переписати заново її громіздкі блоки, якщо це веде до її спрощення.

7.3 Загальна організація програми і її запис

Також як поділ великого твору на глави і параграфи полегшує читання, так і розбиття великої програми на параграфи, розділи (підпрограми і модулі), шляхом виділення логічних одиниць покращує сприйняття програми; допомагає уникнути одноманітності і добре організувати матеріал. Назва розділу відображає його мету.

Структура програми добре реалізується з деяким зміщенням. Для ЕОМ форма запису програми байдужа: зсув рядків призначене для зручності читання програми людиною. Зсув рядків не повинно бути великим: двох-трьох позицій цілком достатньо, щоб виділити структуру програми.

7.4 Контрольні запитання

Запис текстів програм на алгоритмічній мові високого рівня.

1. Стиль програмування.
2. Типова структура програмного модуля.
3. Загальна організація програми і її запис.
4. Читабельність програми. Правила запису структурних конструкцій. Коментарі. Блок коментарів. Ідентифікація імен змінних, функцій, процедур, файлів.
5. Малий програмістський стандарт. Правила скорочень імен.

8 ЛАБОРАТОРНА РОБОТА № 8 «ПОКРОКОВА РОЗРОБКА ПРОГРАМИ»

Мета роботи: Освоїти метод покрокової розробки програм.

8.1 Порядок виконання роботи і звітність

Під час виконання лабораторної роботи необхідно здійснити покрокову розробку конкретної програми.

Робота повинна бути оформлена у вигляді специфікації, що містить опис покрокової розробки конкретної програми.

8.2 Теоретичні відомості

Покрокова розробка (покрокова деталізація) програми це простий процес, який передбачає первинне вираження логіки модуля в термінах гіпотетичної (умовної) мови дуже високого рівня з подальшою деталізацією кожної речення в термінах мови більш низького рівня, до тих пір, поки, нарешті, не буде, досягнутий рівень мови програмування.

Переваги покрокової деталізації полягає в тому, що цей метод дозволяє проектувальнику впорядкувати свої міркування - на кожному кроці деталізації вирішується елементарна задача.

8.3 Контрольні питання

Покрокова розробка програми.

1. Структурне програмування.
2. Метод покрокової розробки програм.
3. Оптимізація програм. Основні прийоми оптимізації програм.
4. Коректність (правильність) програми.
5. Ефективність програми. Ефективність або удобочитаємость?

9 ЛАБОРАТОРНА РОБОТА № 9 «ТЕСТУВАННЯ ТА НАЛАГОДЖЕННЯ РОЗРОБЛЕНОЇ ПРОГРАМИ»

Мета роботи: Здійснити тестування і налагодження розробленої раніше конкретної програми на алгоритмічній мові високого рівня.

9.1 Порядок виконання роботи і звітність

Під час виконання лабораторної роботи необхідно скласти набір тестів до розробленої раніше програми і провести її налагодження. Складений набір тестів необхідно представити в звіті.

9.2 Теоретичні відомості

Тестування – це процес виконання програми з метою визначення місця некоректного її функціонування. Воно включає навмисне конструювання важких наборів вхідних даних, що створюють найбільші можливості для відмови програмного виробу. Тестування є основним методом виявлення помилок в програмі. Результати тестування є вихідними даними для налагодження.

Налагодження програми – це етап її розробки, на якому усуваються недоліки щойно створеної програми.

Якщо програма правильно поводитьися для солідного набору тестів, немає підстав, стверджувати, що в ній немає помилок. Просто невідомо, коли вона не працює і можна говорити лише про деяке рівні її правильності.

Тести, які не сприяють виявленню помилок і тільки підтверджують коректність функціонування програми, є неефективними, тому що призводять до безглузвих витрат ресурсів і часу.

Тест – це прорахований вручну або іншим способом приклад, проміжні та кінцеві результати якого використовуються для контролю правильності (живучості) програмного виробу. Тест складається з вихідних даних і тих значень, які повинні видати налагоджувальні друку при роботі за цим тестом. Ці значення повинні бути записані в точності в тому вигляді, в якому їх повинна видати ЕОМ. Ці значення бажано отримати будь-яким шляхом, але не тим, який реалізований в програмі, тому що в останньому випадку можна не помітити помилки в алгоритмізації.

Комплект тестів повинен бути таким:

- ❖ щоб перевірити всі варіанти зовнішнього ефекту програми і варіанти її внутрішньої роботи алгоритму;
- ❖ щоб всі гілки алгоритму були пройдені, хоча б по одному разу;
- ❖ щоб проконтролювати граничні і вироджені випадки.

Тестові дані повинні підбиратися таким чином, щоб програміст мав можливість обчислити правильний результат ще до початку тестування.

Процес тестування програми можна розділити на три етапи:

1. Перевірка в нормальних умовах.
2. Перевірка в екстремальних умовах.
3. Перевірка у виняткових ситуаціях.

Кожен з цих трьох етапів перевірки повинен гарантувати отримання вірних результатів при правильних вхідних даних і видачу повідомлень про помилки при неправильних вхідних даних.

Перевірка в нормальних умовах

Випадки, коли програма повинна працювати з усіма можливими вихідними даними, надзвичайно рідкісні. Зазвичай мають місце конкретні обмеження на область зміни даних, в якій програма повинна

зберігати свою працездатність. Програма повинна видавати правильні результати для характерних сукупностей вихідних даних.

Перевірка в екстремальних умовах

Тестові дані цього етапу включають граничні значення області зміни вхідних змінних, які повинні сприйматися програмою як правильні дані. Типові приклади дуже великі числа, дуже малі числа або відсутність інформації.

Перевірка у виняткових ситуаціях.

Використовуються вихідні дані, значення яких лежать за межами допустимої області їх зміни. Найгірша ситуація коли програма сприймає невірні дані як правильні і видає невірний, але правдоподібний результат. Програма повинна відкидати будь-які дані, які вона не в змозі обробляти вірно.

Приклад тестів.

Нехай потрібно обчислити довжину діагоналі паралелепіпеда за формулою $d = \sqrt{a^2 + b^2 + c^2}$. Необхідно сформулювати тестові дані для нормальних, екстремальних і виняткових умов (Табл. 9.1).

Таблиця 9.1 – Тестові дані для нормальних, екстремальних і виняткових умов.

Сторони паралелепіпеда			Довжина діагоналі	Примітка
1	1	1	Хороший нормальний тест	
1	2	3	Тест в нормальних умовах	
0	0	0	Результат повинен бути рівний нулю	
0	1	2	Не паралелепіпед. Що станеться?	
1	0	3	Невірні дані	
2	1	0	Невірні дані	
1	-6	3	Невірні дані	
A	B	C	Невірні дані	

9.3 Контрольні питання

Тестування та налагодження програми.

1. Помилки – їх причини появи і наслідки. Класи помилок.

2. Первинні та вторинні помилки. Синтаксичні та семантичні помилки. Захисне програмування. Програмування без помилок.
3. Тестування ПЗ. Поняття тесту. Комплект тестов. Приклад тесту.
4. Методи тестування. Етапи тестування. Аксиоми тестування.
5. Налаштування ПЗ. Етапи, методи і інструменти налаштування програм.

10 ЛАБОРАТОРНА РОБОТА № 10

«СКЛАДАННЯ ДОКУМЕНТА «КЕРІВНИЦТВО КОРИСТУВАЧА»»

Мета роботи: Скласти документ «Керівництво користувача» до розробленої раніше програми.

10.1 Порядок виконання роботи і звітність

Під час виконання лабораторної роботи необхідно скласти документ «Керівництво користувача» до розробленої раніше програми.

Робота повинна бути оформлена у вигляді документа «Керівництво користувача».

10.2 Теоретичні відомості

Формальні вимоги до документації програмного забезпечення описані в ЕСПД (Єдина система програмної документації), неформально: склад документації до програмного забезпечення складається з опису його зовнішнього ефекту і опису його внутрішнього устрою.

Перша частина документації, так звана «Інструкція користувачеві» або «Керівництво користувача» призначена для того,

хто збирається використовувати програмне забезпечення (для користувача), не вникаючи в подробиці його внутрішнього устрою.

Друга частина - «Керівництво програміста» необхідна при модифікації програмного забезпечення або при необхідності виправити в ньому помилку.

В цілому, документація до програмного забезпечення може містити нижче перераховані відомості:

1. Найменування ПЗ і опис завдання, яке воно вирішує.
2. Область застосування ПЗ.
3. Режим роботи ПЗ, повідомлення, що видаються по ходу його роботи, відповіді користувача на них (якщо це необхідно).
4. Вихідні дані, необхідні для роботи ПЗ, а також результати, що їм видаються;
5. Правила підготовки вихідних даних на зовнішніх носіях (якщо вони застосовуються) і вид видаваної інформації;
6. Опис структури даних. Для будь-якої змінної описується її призначення, атрибути (тип, розмір масиву і т.д.), структура інформації в ній, якщо вона не очевидна. Опис змінних мас починається з тих, які служать вихідними даними і результатами.
7. Описи форм, об'єктів. Опис властивостей форм і об'єктів.
8. Тексти програм, процедур (у вигляді роздруківки EOM) з коментарями.
9. Тести.
10. Інструкція (керівництво) користувача.

Інструкція по використанню програми (або просто «Інструкція користувача», або «Керівництво для користувача») – це витримка з повної документації, призначена для експлуатації програми. Вона являє собою незалежний документ для користувача програми, в якому описується: що робить програма і як ним користуватися.

«Інструкція користувача» повинна містити всю необхідну для користувача інформацію і повинна бути йому зрозуміла без додаткових матеріалів (без звернення до інших специфікацій). Отже, необхідна для цієї інструкції інформація переписується повністю з відповідних специфікацій.

Перша частина інструкції є описовою і повинна містити:

- ❖ найменування програми;
- ❖ короткий опис програми;
- ❖ перелік виконуваних програмою функцій;

- ❖ коротку характеристику методу (або методів) вирішення поставленого завдання, його переваги та недоліки;
- ❖ повне бібліографічне посилання на повний опис методу;
- ❖ опис вхідних та вихідних даних;
- ❖ опис структури бази даних (якщо вона є), всіх її таблиць в словесній (вербальній) формі.

Друга частина документа повинна описувати порядок роботи з програмою. Вона повинна містити опис всіх режимів роботи програми, а також зміст усіх роздруківок і діагностичних повідомлень, які видаються по ходу виконання програми.

Слід пам'ятати, що користувач за своєю кваліфікацією не є програмістом і тому його робота з програмою описується на зрозумілій йому мові і досить докладно, а саме:

- ❖ як запустити програму;
- ❖ як продовжити роботу з програмою (описується докладний інтерактивний режим роботи користувача з програмою);
- ❖ підготовка і введення вихідних даних в програму;
- ❖ як реагувати на запити програми;
- ❖ як вести роботу в виняткових ситуаціях;
- ❖ як реагувати на помилки;
- ❖ як відновити роботу програми в разі аварійного його завершення;
- ❖ як отримати необхідний результат;
- ❖ як правильно закінчити роботу з програмою (запланований програмою вихід);
- ❖ інші відомості, необхідні користувачу програми.

10.3 Контрольні запитання

Складання документа «Керівництво користувача».

1. Використання програмного забезпечення.
2. Супровід програмного забезпечення.
3. Розробка програмного забезпечення «Під ключ».
4. Документація до програмного забезпечення.
5. Інструкція по використанню програмного забезпечення.

Додаток А
Приклад оформлення титульної сторінки

Міністерство освіти і науки України
Національний університет «Запорізька політехніка»

Кафедра МіНЕ

ЛАБОРАТОРНА РОБОТА
з дисципліни
„ТЕХНОЛОГІЇ ПРОГРАМУВАННЯ“

(тема роботи)

Виконав:
студент ____ курсу, група РТ-__

(Ініціали, Прізвище)

Прийняв:
(посада)

(Ініціали, Прізвище)

20 __ р.

Додаток Б

Варіанти задач

1. З заданого масиву чисел, вивести спочатку негативні значення, потім нуль, а потім позитивні в зростаючому порядку.
2. Згенерувати такі комбінації: 1111, 1112, 1113, 1121, 1122, 1123, 1131, 1132, 1133, 1211, ..., 3333.
3. Знайти корені рівняння $y = x^2 - 2$ на заданому відрізку методом половинного ділення. Намалювати графік функцій.
4. Знайти найбільший спільний дільник заданих N чисел.
5. Обчислити визначник N -го порядку ($N \leq 5$), користуючись формулою розкладання визначника по i -му рядку і знаючи формулу обчислення визначника 2-го порядку.
6. На місцевості є N населених пунктів, пронумерованих від 1 до N ($N \leq 10$). Деякі з пунктів з'єднані між собою дорогами. Інформація про дороги задається в вигляді послідовності пар чисел i, j ($i < j$), що вказують, що i -й і j -й пункти з'єднані дорогою, ознака кінця цієї послідовності - пара нулів. Визначити, чи можна потрапити по цих дорогах з першого пункту в n -ий.
7. Реалізувати алгоритм так званої «швидкого сортування» Хоара: є два покажчика i і j , причому спочатку $i = 1$, а $j = N$ (номеру останнього елемента).
 Порівнюємо $a[i]$ і $a[j]$, і якщо обмін не потрібний, то зменшимо j на 1 і повторюємо цей процес.
 Після першого обміну збільшимо i на 1 та будемо продовжувати порівняння, збільшуючи i , поки не відбудеться ще один обмін.
 Тоді знову зменшимо j і т.д., тобто будемо «спалювати свічку з обох кінців», поки не стане $i = j$. В результаті отримаємо, що зліва від $a[i]$ виявилися тільки менші елементи, а справа - тільки великі (тим самим елемент $a[i]$ виявиться на своєму остаточному місці).
 Продовжити вищеописаний метод для лівої і правої частин масиву до тих пір, поки в підмасиві не залишиться тільки один елемент.
8. Знайти найменше спільне кратне заданих N чисел.
9. У заданому рядку перевірити чергування букв і цифр. Вивести рядок, що складається з одиниць і нулів (1 - буква, 0 - цифра).

10. Сформувати послідовність рядків:

A
BB
CCC
DDDD

.....
ZZZZZZZ 26 разів

11. Сформувати все перестановки без повторень деякого масиву значень. Наприклад якщо заданий масив, що містить символи ABC, то необхідно сформувати такі комбінації: ABC, ACB, BAC, BCA, CAB, CBA.
12. Обчислити функцію Аккермана (m і n - натуральні числа):
- $$A(n,m) = \begin{cases} m+1 & , \text{ якщо } n=0 \\ A(n-1,1) & , \text{ якщо } n > 0 \text{ и } m=0 \\ A(n-1, A(n,m-1)) & , \text{ якщо } n > 0 \text{ и } m > 0 \end{cases}$$
13. Є відомості про ріст N учнів одного класу (дані вводяться в алфавітному порядку прізвищ учнів). Визначити середній ріст учнів, чиї прізвища розташовані в журналі між прізвищами учнів з мінімальним і максимальним ростом (зростання цих учнів теж враховувати), вважаючи, що найвищий і найнижчий учні в класі єдині.
14. Кіт Матроскін і Шарик загадували парні і непарні числа в довільному порядку, поки не скінчилося місце на печі, де вони записували ці числа. Визначити, яких чисел було загадане більше: парних або непарних, якщо останнім було записано число 0 (при підрахунку цього числа не враховувати). Скільки парних чисел передували першому непарному?
15. У коробці перемішалися кубики. На всіх гранях кожного кубика намальовані однакові букви або цифри. Потрібно розкласти їх в різні коробки: кубики з цифрами помістити в одну, а з буквами - в іншу. Вивести на екран вміст всіх трьох коробок (що спочатку лежало в першій коробці, потім роздрукувати вміст отриманих двох коробок).
16. Наочно продемонструвати графіки функцій: $y = a \sin (bx + c) + d$, змінюючи крок аргументу і кордони відрізка. Провести дослідження функції на парність, спадання і зростання. Знайти коріння і екстремальні точки.

17. Сформувати всілякі варіанти розстановки m ($m > 3$) ферзів на шахівниці $m \times m$ клітин, при яких ферзі не б'ють один одного.
18. В арифметичному виразі $1 * 2 * 3 * 4 * 5$ замість зірочок розставити арифметичні операції $+$, $-$, $*$, $/$ так, щоб вийшло число 543.
19. Фунтик і Хрюша вирішили порівняти свої книжки «по товщині». Вони склали список, вказуючи разом з назвами число сторінок в кожній книзі. Водночас з'ясувалося, що всі книжки мали різну кількість сторінок, деякі книги були як в бібліотеці Фунтика, так і в бібліотеці у Хрюші. Допоможіть Фунтику знайти саму «товсту» його книжку серед тих, яких немає у Хрюші. (Книжки порівнювати за кількістю сторінок).
20. У двох спортсменів були гирі для тренувань, на кожній було вказано її вага. Щоразу після чергового заняття вони становили їх по зростанню ваг гир. Потім вирішили об'єднати свої гирі, зберігаючи звичне розташування гирь. Потрібна допомога в їх розміщенні (відомий вага кожної гирі у вихідних наборах). Надрукувати порядок гирь у вихідних наборах і в отриманому.
21. Щоб закріпити поняття «парне» і «непарне», хлопчик записував на папері різні цілі числа, а потім парні розфарбовував червоним олівцем, непарні - синім. Після розфарбовування він розрізав смужку паперу на окремі числа і вирішив скласти візерунок: побудувати їх в лінію в такому порядку: червоний квадратик, синій квадратик, червоний, синій і т.д. Решта числа поклав в кінець отриманого візерунка. Скласти програму відображення отриманої послідовності чисел на екрані у відповідному кольорі.
22. Дано дійсні числа $a(1) \dots a(50)$. Вони визначають інтервали на числової осі $[a(1), a(2)]$, $\dots$, $[a(49), a(50)]$. Визначити, чи мають всі інтервали загальні точки? Якщо так, то вказати яку-небудь з цих точок.
23. Учасники гри «О! Щасливчик» розсідаються на пронумерованих стільцях навколо великого круглого столу. За допомогою константи рахунку починається їх відлік за годинниковою стрілкою. Гравець, на якого потрапляє константа рахунку, зобов'язаний звільнити місце. Відлік гравців триває до тих пір, поки не залишаться двоє людей. При відомому числі гравців N і константи рахунку C визначити номери стільців, які потрібно зайняти гравцям, щоб потрапити в число двох «щасливчиків».

24. Необхідно серед введених N цілих чисел розпізнати ступеня двійки (наприклад, 1, 2, 4, 8, 16 і т.д.).
25. На пронумерованих дискетах записано різну кількість файлів, і тільки дві дискети мають однакове число файлів. Знайти їх номери.
26. Кожен учень 8 класу знайшов на археологічних розкопках по одній глиняній табличці з зашифрованими на них числами (на кожній табличці по два числа.). Коли вдалося розшифрувати ці числа, то виявилось, що на деяких табличках сума двох чисел є простим числом, а на інших - складовим. Визначити, скільки табличок було зі складовими числами.
27. Є масив N натуральних чисел. Визначити, скільки в ньому чисел:
 - а) різних;
 - б) різних парних;
 - в) різних простих.
28. Таблиця кола футбольного чемпіонату задана квадратної матрицею розміру $N \times N$. Виграш - 2 очки, нічия - 1 очко, програш - 0 очок.

Надрукувати номер команди, яка зайняла перше місце, номера команд, які пройшли чемпіонат без поразок, підрахувати кількість очок у кожній команді, визначити номер команди - аутсайдера (при рівній кількості очок у двох команда переважно визначається з результату гри між цими командами) .
29. Три мушкетери отримали відомості, що 12 вкрадених алмазних підвісок зберігаються в десяти різних замках Англії, причому, в кожному замку їх по 12 штук (частина з них справжні, а відсутні замінені фальшивими). Відомо, що у справжніх підвісок по 13 алмазів, у підроблених же різну кількість каменів, але не 13. Допоможи мушкетерам знайти всі 12 справжніх алмазних підвісок королеви.
30. У залі для глядачів N рядів по M місць в кожному. Інформація про продані квитки зберігається в двох матрицях: в першій матриці у відповідній клітинці вказана ціна квитка за дане місце, а в другій - продано квиток на це місце чи ні (квиток проданий - «1», не проданий - «0»). Підрахувати загальну виручку за продані квитки.

31. На м'ячиках були написані різні літери і цифри. Необхідно відібрати ті м'ячики, які зустрічаються по одному разу. Отриманий набір надрукувати.
32. З елементів заданого масиву створити новий масив, відібравши в нього ті числа з вихідного масиву, які є ступенем заданого числа R . Числа в новому масиві не повинні повторюватися.
33. Надрукувати таблицю множення в системах числення
 - а) десяткової;
 - б) вісімкової;
 - в) шестнадцатеричної.
34. У квадратній цілочисельній матриці порядку N знайти однакові елементи, які зустрічаються одночасно як вище, так і нижче головної діагоналі (саму головну діагональ не розглядати).
35. Нехай задано N кіл на площині координатами їх центру X , Y та радіусом R . Чи є серед даних кіл дві попарно перетинаються? Якщо так, то надрукувати їх параметри (номера, координати і радіуси).
36. Рядок символів являє собою запис арифметичного виразу. Перетворити цифри, що стоять підряд в натуральні числа і зробити необхідні обчислення:
 - а) виконати дії по порядку;
 - б) дії $+$, $-$, $*$, $/$ виконуються відповідно пріоритету цих операцій;
 - в) в запису може бути присутнім одна пара або навіть делулька пар круглих дужок, що змінює порядок виконання операцій.
37. Дано N рядків тексту кирилицею. Скласти програму зашифрування, а потім її розшифрування, користуючись таким простим правилом: кожна буква замінюється на наступну за алфавітом (при цьому буква «я» замінюється на букву «а»). Ускладнений варіант: буква замінюється на іншу, яка слідує за нею через N позицій (заміна також циклічна).
38. На метеорологічній станції в комп'ютер введені відомості про середньодобовій температурі за березень. Скласти програму, яка б визначала:
 - а) кількість днів, коли температура була нижче 0°C ;
 - б) суму позитивних температур;
 - в) середню температуру місяці;

г) день, коли температура найближче підходило до середньомісячної.

39. Розрахувати таблицю значень і побудувати графік функції одного аргументу $y = f(x)$ на заданому відрізку по введеної формулою, змінюючи крок аргументу і границі відрізка.
40. У літерному файлі lit.txt знайти символ, що повторюється максимальне число раз. Надрукувати цей символ і кількість його повторень.
41. Обчислити заданий інтеграл методами Трапецій і Сімпсона. В обох випадках вивести також кількість розбиття відрізка інтегрування для досягнення однієї і тієї ж точності обчислення інтеграла.
42. Наочно продемонструвати графіки функцій: $y=ax^2+bx+c$, змінюючи крок аргументу і границі відрізка. Провести дослідження функції на парність, спадання і зростання. Знайти коріння і екстремальні точки.
43. Маленький Мук пробував друкувати в текстовому редакторі. Визначити, скільки різних символів він надрукував на екрані, а також літеру, надруковану максимальне число разів. Вивести на екран отриманий результат.
44. Нехай задано N кіл на площині координатами їх центру X , Y та радіусом R . Знайти номери всіх відокремлених кіл, тобто таких, які не мають спільних точок з жодною з інших кіл, які не лежать цілком всередині і не вміщають у собі будь - яке з інших кіл.
45. Даний рядок тексту з букв і цифр. Не користуючись стандартною процедурою мови програмування для перекладу чисел в число, організувати свою процедуру перекладу розташованих в ряд цифр в цілі числа і знайти їх середнє арифметичне. Надрукувати отримані числа і результат.
46. У зачарованому замку в парадному залі підлога викладена мозаїчними мармуровими плитами, що мають форму неправильних восьмикутників, всього в залі було 57 таких плит. Якщо наступити на плитку з найменшою площею, то двері залу відкриються. Кожна плита задана координатами своїх вершин. Знайти номер цієї потайний плити.
47. У трьох таблицях записані оцінки за семестр в трьох восьмих класах А, В і С. Кількість учнів у класах Ма, Mb, Mc (рядків) різняться, кількість предметів - N (стовпців) однакове. Надруку-

- вати ту таблицю, в якій найбільшу кількість рядків з усіма п'ятірками (тобто більше відмінників) і, якщо таких таблиць декілька, надрукувати їх все.
48. У стіні є прямокутний отвір розмірами X і Y . пролізе в дану дірку цегла, з розмірами A , B і C , якщо проштовхувати його таким чином, щоб сторони цегли були паралельні сторонам отвору?
 49. Дама здавала в багаж коробку і порожній валізу з відповідними розмірами X , Y , Z і A , B , C . Чи вдасться заощадити місце в багажному відділенні, помістивши цю коробку в валізу, якщо сторони коробки і валізи будуть паралельні?
 50. Одна дама вирішила купити новий м'який куточок (диван, 2 крісла, журнальний столик). Відома вартість кожного виробу окремо. Чи вистачить у неї грошей на покупку всього набору, якщо немає, то що зможе вона купити із запропонованого (вивести всі варіанти набору).
 51. Білосніжка на Різдво вирішила купити гномам нові сорочки. Вона визначила розміри кожного: R_1 , R_2 , R_3 , R_4 , R_5 , R_6 , R_7 . В магазині їй запропонували сорочки трьох розмірів: I - 25 - 35 см, II - 36 - 45 см і III - 46 - 55 см. Скільки Білосніжка купить сорочок I - го, II - го і III - го розмірів?
 52. Координати будь-якого поля шахівниці можуть бути задані парою натуральних чисел, що не перевершують восьми. Дано чотири числа (K , L і M , N), що задають координати двох клітин (при рахунку зліва направо і знизу вгору). визначити:
 - а) чи є ці два поля шахівниці полями одного кольору?
 - б) якщо на першому полі розташований ферзь, то чи загрожує він фігурі противника, розташованої на другому полі?
 - в) якщо на першому полі розташований кінь, то чи загрожує він фігурі противника, розташованої на другому полі?
 53. За даним тризначного числа N знайти і надрукувати всі нові тризначні числа, побудовані з тих же цифр, а також виділити з них ті числа, які виявилися більше вихідного.
 54. Обчислити корені заданої функції на заданому відрізку методом хорд. Намалювати графік функцій.
 55. За весь навчальний рік вчителька поставила в восьмому класі K двійок, L трійок, M четвірок і N п'ятірок. Який вийшов середній

- бал за кожен триместр, якщо половина двійок і трійок були виставлені в першому триместрі, інша половина - порівну в другому і третьому? Четвірки і п'ятірки розподілилися так: в III - му триместрі - половина, а в I-му і в II-м триместрах порівну.
56. У масиві містяться відомості про ріст " n " учнів одного класу (дані вводяться в алфавітному порядку). Визначити середнє ріст учнів, чиї прізвища розташовані між прізвищами учнів з мінімальним і максимальним ростом (їх ріст також враховується).
57. Сформувати матрицю $m \times n$ елементів, заповнивши її цілими числами від -9 до 9 випадковим чином. Відсортувати елементи кожного рядка з першого до передостаннього по зростанню. Останній елемент кожного рядка замінити сумою позитивних елементів цього рядка.
58. Заповнити матрицю $n \times n$ випадковими елементами і виконати наступні дії:
- а) Знайти суму елементів матриці по його периметру.
 - б) Знайти суму позитивних елементів.
 - в) Знайти середнє арифметичне елементів випадкового рядку.
59. Є m різних предметів, їх вага і вартість. Визначити, які предмети можна покласти в рюкзак ємністю до 50 кг так, щоб загальна вартість предметів була мінімальною.

Додаток В

Додаткові питання

1. Надійність програмного забезпечення.
2. Програмні вироби з малою та великої тривалістю експлуатації.
3. Системний аналіз (попереднє проектування).
4. Конструювання програмного забезпечення.
5. Програмування програмного забезпечення.
6. Оцінка (випробування) програмного забезпечення.
7. Використання (експлуатація та супровід) ПЗ.
8. Розбиття програмних проєктів на групи з точки зору розробки вимог до них.
9. Документ «Угода про вимоги».
10. Проблеми проектування великих програмних засобів.
11. Система автоматизованого проектування ПЗ (САПР ПЗ).
12. Об'єктно - орієнтоване проектування ПЗ.
13. Моделювання системи. Модель системи. Об'єктна модель. Об'єкт, клас, властивості об'єктів, операції, методи. Узагальнення і успадкування.
14. Основні принципи проектування програмного забезпечення.
15. Конструювання об'єктної моделі.
16. Документ «Зовнішня специфікація».
17. Система автоматизації програмування (САП).
18. Документ «Внутрішня специфікація».
19. Оптимізація ефективності експлуатації ПЗ.
20. Об'єктно-орієнтоване програмування.
21. Поняття об'єктно-орієнтованого програмування: об'єкт, клас, подія, метод, інкапсуляція, успадкування, поліморфізм, форма.
22. Розподіл операцій по класах.
23. Об'єктно-орієнтовані мови програмування.
24. Візуальне програмування. Інтегроване середовище розробки програми. Майстер розробки програми. Елементи управління додатком.
25. «Альфа» і «Бета» тестування програмного забезпечення.
26. Система автоматизації налагодження програмного забезпечення.

27. Оцінка (випробування) програмного забезпечення.
28. Попередні і спільні випробування ПО.
29. Властивості якісного програмного забезпечення.
30. Проблема оцінки якості програмних продуктів.
31. Визначення якості програмного забезпечення.
32. Якісні характеристики ПО: зрозумілість, завершеність, осмисленість, мобільність, корисність, машинезалежність, надійність, структурованість, ефективність, точність, доступність, модифіцируемість, відкритість, комунікативність, інформативність, розширюваність, врахування людського фактора.