

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КРЕМЕНЧУЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ МИХАЙЛА ОСТРОГРАДСЬКОГО



МЕТОДИЧНІ ВКАЗІВКИ
ЩОДО ВИКОНАННЯ ЛАБОРАТОРНИХ РОБІТ
З НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
«КЕРУВАННЯ РОЗРОБКОЮ ПРОГРАМНИХ СИСТЕМ»
ДЛЯ СТУДЕНТІВ ДЕННОЇ ФОРМИ НАВЧАННЯ
ЗІ СПЕЦІАЛЬНОСТІ
123 – «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ»
ОСВІТНЬОГО СТУПЕНЯ «МАГІСТР»
ЧАСТИНА I

КРЕМЕНЧУК 2018

Методичні вказівки щодо виконання лабораторних робіт з навчальної дисципліни «Керування розробкою програмних систем» для студентів денної форми навчання зі спеціальності 123 – «Комп'ютерна інженерія» освітнього ступеня «Магістр». Частина I

Укладачі: д. т. н., проф. М. І. Гученко,
к. т. н. П. П. Костенко,
асист. Н. Л. Сохін

Рецензент к. т. н., доц. О. Г. Славко

Кафедра комп'ютерних та інформаційних систем

Затверджено методичною радою Кременчуцького національного університету імені Михайла Остроградського

Протокол № _____ від _____

Голова методичної ради _____ проф. В. В. Костін

ЗМІСТ

Вступ.....	4
1 Перелік лабораторних робіт.....	5
Лабораторна робота № 1 Розробка технічного завдання.....	5
Лабораторна робота № 2 Структурний підхід до програмування. Стадії «Ескізний проект» та «Технічний проект».....	12
Лабораторна робота № 3 Структурний підхід до програмування. Стадія «Реалізація. Розробка програмного засобу».....	21
2 Критерії оцінювання знань студентів.....	25
Список літератури.....	26

ВСТУП

Предметом навчальної дисципліни «Керування розробкою програмних систем» є методи, призначені для створення проектних основ, моделей та алгоритмів, опису майбутнього проекту, документації, призначеної для розробки програмних комплексів. Під час виконання лабораторних робіт закріплюються знання, отримані на лекціях, де студенти вивчають основні поняття, структуру, вимоги до створення і моделювання комп'ютерних проектів. Студенти вивчають методи відслідковування помилок на етапі створення проекту та методи тестування готових програмних комплексів.

Виконання кожної лабораторної роботи передбачає ознайомлення студента з лабораторним практикумом та його теоретичну підготовку з відповідних розділів дисципліни. До кожного розділу наведено перелік питань для контролю. Об'єкт автоматизації для дослідження визначається викладачами індивідуально. На захист виконаної студентом лабораторної роботи оформлюється окремий звіт, за потреби демонструється електронний документ з виконаним завданням. Підсумкові оцінки, отримані студентом за виконання лабораторних робіт і відповіді на контрольні запитання, враховуються під час обчислення модульних оцінок та семестрової підсумкової оцінки з навчальної дисципліни.

Послідовне виконання лабораторних робіт формує цілісне документальне представлення, що є необхідним для створення автоматизованої системи керування, починаючи з визначення самого проекту, опису робіт, надання моделі проекту, алгоритму виконання програми, розробки програмного модуля, методів відслідковування помилок.

Під час виконання лабораторних робіт студент отримує знання, що розвинуть аналітичні здібності, нададуть змогу правильно оформляти проектну документацію, виступати керівником проекту з впровадження автоматизованих систем. Отримані знання забезпечать достатню кваліфікацію студента як спеціаліста в сфері аналітики і керівника проектів.

1 ПЕРЕЛІК ЛАБОРАТОРНИХ РОБІТ

Лабораторна робота № 1

Тема. Розробка технічного завдання

Мета: навчитися розробляти технічне завдання на створення інформаційної системи.

Короткі теоретичні відомості

1. Зміст та структура технічного завдання

Технічне завдання (ТЗ) – вихідний документ для проектування споруди чи промислового комплексу, конструювання технічного пристрою (приладу, машини, системи керування тощо), розробки автоматизованої системи, створення програмного продукту або проведення науково-дослідних робіт (НДР), відповідно до якого проводиться виготовлення, приймання під час введення в дію та експлуатації відповідного об'єкта.

Згідно з ГОСТ 34.602-89 ТЗ є основним документом, що визначає вимоги і порядок створення (розвитку або модернізації) інформаційної системи, відповідно до якого проводиться її розробка і приймання при введенні в дію.

Технічне завдання на розробку АСУ (ТЗ) має містити такі розділи.

1. Загальні положення, де вказують повну назву системи та її умовне позначення, а також перелік документів, на підставі яких створюється система, термін розробки системи і порядок подання замовнику результатів роботи.

2. Призначення і цілі створення системи. В підрозділі «Призначення системи» вказують вид діяльності, що автоматизується, і перелік об'єктів автоматизації, де буде впроваджена система. В підрозділі «Цілі створення системи» наводять назви і необхідні значення технічних, технологічних, техніко-економічних та інших показників, які мають бути досягнуті в результаті впровадження АСУ.

3. Характеристика об'єкту автоматизації містить стислі відомості про об'єкт автоматизації та відомості про умови експлуатації та навколишнє

середовище.

4. Вимоги до системи складаються з таких підрозділів: вимоги до системи в цілому; вимоги до функцій (задач), які виконує система; вимоги до видів забезпечення.

2. Вимоги до розроблюваної інформаційної системи

Вимоги до розроблюваної інформаційної системи в цілому містять:

- вимоги до структури і функціонування системи;
- вимоги до чисельності і кваліфікації персоналу системи та умов його роботи;
- показники призначення;
- вимоги до надійності;
- вимоги безпеки;
- вимоги до ергономіки та технічної естетики;
- вимоги до експлуатації, технічного обслуговування, ремонту і зберігання компонентів системи;
- вимоги до захисту інформації від несанкціонованого доступу;
- вимоги до збереження інформації в разі аварії;
- вимоги до захисту від впливу зовнішніх чинників;
- додаткові вимоги.

Вимоги до структури і функціонуванню системи містять:

- перелік підсистем, їх призначення і основні характеристики, вимоги до кількості рівнів ієрархії та ступеня централізації системи;
- вимоги до способів і засобів зв'язку для інформаційного обміну між компонентами системи;
- вимоги і характеристики взаємозв'язків системи, що створюється із суміжними системами, а також засобів обміну інформацією;
- вимоги до режимів функціонування системи;
- вимоги до діагностування системи;
- перспективи розвитку та модернізації системи.

Вимоги до чисельності і кваліфікації персоналу системи містять:

– вимоги до чисельності персоналу (або користувачів) системи, вимоги до кваліфікації персоналу, порядку підготовки та контролю його знань та навичок, режиму роботи персоналу.

У вимогах до показників призначення наводять:

– значення параметрів, що характеризують ступінь відповідності системи її призначенню, а саме: ступінь адаптації системи до зміни процесів та методів управління, допустимі межі модернізації системи, імовірно-часові характеристики системи, за яких зберігається цільове призначення системи.

Вимоги до надійності системи містять:

– склад та значення кількісних показників надійності для системи в цілому або її підсистем;

– перелік аварійних ситуацій, відносно яких мають бути регламентовані вимоги до надійності, та значення відповідних показників;

– вимоги до надійності технічних засобів та програмного забезпечення;

– вимоги до методів оцінювання та контролю показників надійності на різних стадіях створення системи відповідно до діючих нормативних документів.

Вимоги безпеки містять:

– вимоги до забезпечення безпеки під час монтажу, налагодження, експлуатації, обслуговування та ремонту технічних засобів системи (захист від дії електричного струму, електромагнітних полів, шуму тощо), вимоги до рівня освітлення, вібраційних та шумових навантажень.

У вимогах до ергономіки та технічної естетики наводять:

– показники АС, що задають необхідну якість взаємодії людини з ПК і комфортність умов роботи персоналу.

Вимоги до експлуатації, технічного обслуговування, ремонту і зберігання компонентів системи складаються з:

– умов і регламенту експлуатації, які мають забезпечити використання

технічних засобів системи з обумовленими технічними показниками, в тому числі види і періодичність обслуговування технічних засобів або допустимість роботи без обслуговування;

- вимог до кількості і кваліфікації обслуговуючого персоналу та режимів його роботи і регламенту обслуговування;

- вимог до складу, розміщення та умов зберігання комплекту запасних виробів та приладів.

У вимогах до захисту інформації від несанкціонованого доступу містяться:

- вимоги, які записані в нормативних документах, що діють в галузі (відомстві) замовника.

У вимогах до збереження інформації в разі аварії наводять:

- перелік подій, які кваліфікуються як аварія, в разі яких має бути збережена інформація.

Вимоги до захисту від впливу зовнішніх чинників складаються:

- з вимог до радіоелектронного захисту засобів АС та вимог щодо стійкості, сталості та міцності відносно негативних зовнішніх чинників.

У вимогах до функцій (задач) наводять:

- по кожній підсистемі перелік функцій, задач або їх компонентів;
- часовий регламент реалізації кожної задачі або функції;
- вимоги до якості реалізації кожної функції (або задачі), до форми подання вихідної інформації, характеристики необхідної точності і часу виконання, вимоги одночасного виконання групи функцій, достовірності видачі результатів;

- перелік і критерії відмов для кожної функції, для якої задаються вимоги до надійності.

У вимогах до видів забезпечення залежно від виду системи наводять вимоги до математичного, інформаційного, лінгвістичного, програмного, технічного, метрологічного, організаційного та інших видів забезпечення

системи.

Для *математичного забезпечення* наводять вимоги до складу, галузі і способів використання у системі математичних методів і моделей, типових алгоритмів і алгоритмів, що необхідно розробити.

Для *інформаційного забезпечення* наводять вимоги:

- до складу, структури і засобів організації даних в системі;
- до інформаційного обміну між системами;
- до інформаційної сумісності із суміжними системами;
- до використання систем керування баз даних;
- до збору, обробки, передачі даних в системі і подання даних;
- до захисту даних від пошкоджень в разі аварії або збою в електроживленні системи;
- до контролю, зберігання, оновлення та відновлення даних.

Для *лінгвістичного забезпечення* наводять вимоги до використання в системі мов програмування високого рівня, мов взаємодії користувачів і технічних засобів системи, а також вимоги до кодування і декодування даних, до мов введення-виводу даних, до способів організації діалогу.

Для *програмного забезпечення* системи наводять перелік необхідних програмних засобів, а також вимоги:

- до незалежності програмних засобів від засобів обчислювальної техніки та операційного середовища;
- до якості програмних засобів, а також до способів його забезпечення і контролю.

Для *технічного забезпечення системи* наводять вимоги:

- до видів технічних засобів, в тому числі видів комплектуючих технічних засобів, програмно-технічних комплексів та інших комплектуючих виробів, які можуть бути використані в системі;
- до функціональних, конструктивних і експлуатаційних характеристик засобів технічного забезпечення системи.

У вимогах до *метрологічного забезпечення* вказують вимоги до метрологічного забезпечення технічних і програмних засобів, які входять до складу вимірювальних каналів системи, і засобів вимірювань, що використовуються під час налагодження і випробувань системи.

Для *організаційного забезпечення* наводять вимоги:

- до структури і функцій підрозділів, що беруть участь в експлуатації системи або забезпечують її експлуатацію;
- до організації функціонування системи;
- до захисту від помилкових дій персоналу системи.

3. Склад і зміст робіт зі створення системи

Розділ технічного завдання «Склад і зміст робіт зі створення інформаційної системи» містить:

- перелік стадій та етапів робіт зі створення системи і терміни їх виконання;
- перелік документів, що подаються після виконання кожного етапу;
- вид і порядок проведення експертизи технічної документації;
- програму робіт із забезпечення надійності системи.

Порядок контролю та приймання системи складається з:

- опису видів, складу, об'єму та методів випробувань системи та її складових частин;
- загальних вимог до прийому робіт за стадіями.

Вимоги до складу та змісту робіт з підготовки об'єкта автоматизації до введення системи в дію містять перелік основних заходів, які необхідно виконати під час підготовки об'єкта автоматизації до введення АС в дію, а саме:

- доведення інформації, що надходить в систему до вигляду, придатного до введення в ПК;
- створення умов функціонування об'єкту автоматизації, за яких гарантується відповідність системи, що створюється, вимогам ТЗ;
- терміни і порядок комплектації штату і навчання персоналу.

Вимоги до документування містять узгоджений із замовником перелік документів, які належить розробити, а також вимоги до складу і змісту цих документів.

Джерела розробки складаються з переліку документів та інформаційних матеріалів (нормативних документів), на підставі яких було розроблено ТЗ і які мають використовуватися під час створення системи.

В додатках до ТЗ наводять:

- розрахунок очікуваної ефективності системи;
- оцінювання науково-технічного рівня системи.

Порядок виконання роботи

1. Отримати у викладача варіант завдання.
2. Розробити технічне завдання для реалізації системи у заданій предметній галузі, що включає такі пункти:
 - загальні положення про розроблювану систему;
 - призначення і мета створення системи;
 - характеристика об'єкту автоматизації;
 - вимоги до системи;
 - технічні вимоги;
 - економічні показники;
 - порядок контролю і приймання об'єкта.

Зміст звіту

1. Назва та мета роботи.
2. Розроблене технічне завдання згідно з варіантом.
3. Письмові відповіді на контрольні запитання.

Контрольні питання

1. Що таке технічне завдання?
2. Наведіть приклади вимог до надійності системи.
3. Перелічіть основні джерела розробки технічного завдання.
4. З яких розділів складається технічне завдання на розробку системи?

5. Згідно з якими стандартами виконується технічне завдання на розробку автоматизованої системи?

6. Що міститься у розділі «Програмне забезпечення» технічного завдання?

7. Що характеризують значення параметрів, які наводять у вимогах до показників призначення?

8. Що входить у розділ «Склад і зміст робіт зі створення системи» технічного завдання?

9. Які вимоги ставлять до математичного забезпечення?

10. З чого складається порядок контролю та приймання системи?

Література: [1–6].

Лабораторна робота № 2

Тема. Структурний підхід до програмування. Стадії «Ескізний проект» та «Технічний проект»

Мета: навчитися розробляти ескізний та технічний проекти на створення інформаційної системи.

Короткі теоретичні відомості

1. Стадія «Ескізний проект»

Розділ «Ескізний проект» складається з 3-х підрозділів:

- документи ескізного проекту;
- документи робочої документації;
- пояснювальна записка до ескізного проекту.

Документи ескізного проекту складаються з переліків вхідних сигналів та даних і переліку вихідних сигналів та даних.

В переліку вхідних сигналів та даних вказують:

- для аналогового сигналу – назву вимірюваної величини, одиниці виміру, діапазон змін, вимоги точності і періодичності змін, тип сигналу;

- для дискретного сигналу – назву, розрядність і періодичність, тип сигналу;

- для сигналу типу «так–ні» – джерело формування та змістовне значення сигналу;

- назву, кодове позначення і значення реквізитів вхідних даних;

- назву і кодове позначення документів, які містять ці повідомлення.

Крім того, необхідно обов’язково вказати, чи будуть вводитися вхідні дані з документів, чи треба буде заповнювати деякі проміжні форми, і надати вигляд цих форм. Якщо вхідні дані вибираються з інших документів, то треба обов’язково вказати, які саме дані.

Перелік вихідних сигналів і документів містить:

- перелік вихідних сигналів з позначенням їх назв, призначення, одиниць виміру і діапазонів значень, способів подання, користувачів інформації;

- перелік вихідних документів з позначенням їх назв, кодових позначень і значення реквізитів, користувачів інформації.

До документів робочої документації належать:

- загальний опис інформаційної системи;

- інструкція користувача системою;

- інструкція з формування і ведення баз даних;

- інструкція з експлуатації комплексу технічних засобів;

- опис технологічного процесу обробки даних.

Пояснювальна записка до ескізного проекту складається з таких документів:

- опис функцій, які автоматизуються, і алгоритму функціонування у вигляді функціональної моделі (в будь-якій нотації);

- опис комплексу задач;

- опис інформаційного забезпечення АСУ;

- опис програмного забезпечення;

- опис математичного забезпечення.

a. Опис інформаційного забезпечення складається з:

– складу інформаційного забезпечення із зазначенням всіх баз даних і наборів даних;

- організації інформаційного забезпечення;
- організації збору і передачі інформації;
- побудови системи класифікації і кодування;
- організації внутрішньої інформаційної бази;
- організації зовнішньої інформаційної бази.

b. Опис програмного забезпечення складається з:

- структури програмного забезпечення;
- функцій частин програмного забезпечення;
- методів і засобів розробки програмного забезпечення;
- операційної системи;
- засобів, що розширюють можливості операційної системи.

c. Опис математичного забезпечення системи наводиться в документі

«Опис алгоритму (проектної процедури)», який складається з таких розділів:

- призначення і характеристика;
- використана інформація;
- результаті рішення;
- математичний опис;
- алгоритм рішення.

Всі документи виконуються відповідно до РД 50-34.698-90. *Додаткові відомості.*

2. Стадія «Технічний проект»

Мета технічного проекту – деталізація рішень Ескізного проекту, що використовуються під час створення інформаційної системи, і остаточне визначення її кошторисної вартості.

Технічне проектування підсистем здійснюється відповідно до затвердженого технічного завдання. Склад документації Технічного проекту

автоматизованих систем регламентує документ РД 50-34.698-90.

Технічний проект програмної системи має, в основному, описувати:

- функції, що виконуються (розв'язувані завдання);
- уточнені структури (форми) вхідних та вихідних даних;
- структуру оброблюваних баз даних;
- структуру розроблюваного програмного забезпечення (програмного засобу);
- алгоритми вирішення завдань;
- склад програмного забезпечення, що використовується (ОС, СКБД та ін.);
- конфігурацію та характеристики технічних засобів, що використовуються.

Під час розробки технічного проекту автоматизованої системи оформлюються:

- відомість технічного проекту (загальна інформація по проекту);
- пояснювальна записка до технічного проекту (вступна інформація, що дозволяє користувачеві швидко освоїти дані по конкретному проекту);
- опис систем класифікації та кодування;
- перелік вхідних даних/документів (перелік інформації, яка використовується як вхідний потік і є джерелом накопичення даних);
- перелік вихідних даних/документів (перелік інформації, яка використовується для аналізу накопичених даних);
- опис програмного забезпечення, що використовується (перелік програмного забезпечення і СКБД, які планується використовувати для створення інформаційної системи);
- опис технічних засобів, що використовуються (перелік апаратних засобів, на яких планується робота проектного програмного продукту);
- проектна оцінка надійності системи (експертна оцінка надійності з виявленням найбільш надійних ділянок програмної системи та її вузьких місць;

– відомість устаткування і матеріалів. Перелік обладнання та матеріалів, які потрібні під час реалізації проекту.

Структурна схема

Структурною називають схему, що відображає склад і взаємодію частин розроблюваного програмного забезпечення. Структурна схема визначається архітектурою розроблюваного ПЗ.

Функціональна схема

Функціональна схема – це схема взаємодії компонентів програмного забезпечення з описом інформаційних потоків, складу даних в потоках та зазначенням використовуваних файлів і пристроїв.

Розробка алгоритмів

Метод покрокової деталізації реалізує спадний підхід до програмування і передбачає покрокову розробку алгоритму.

Структурні карти

Методика структурних карт використовується на етапі проектування ПЗ для того, щоб продемонструвати, як програмний продукт виконує системні вимоги. Структурні карти Константайна призначені для опису відносин між модулями.

Техніка структурних карт Джексона заснована на методі структурного програмування Джексона, який виявляє відповідність між структурою потоків даних і структурою програми. Основну увагу в методі сконцентровано на відповідності вхідних і вихідних потоків даних.

Проектування програмного забезпечення

Проектування програмного забезпечення при структурному підході

Під час проектування складного програмного забезпечення перш за все необхідно визначити структурні компоненти і зв'язки між ними. Отримана в результаті структура ПЗ має бути подана у вигляді структурної або функціональної схем і специфікацій її компонентів.

Структурна схема розроблюваного програмного забезпечення

Структурною називають схему, що відображає склад і взаємодію

керування частин розроблюваного програмного забезпечення.

Структурна схема визначається архітектурою розроблюваного ПЗ.

Розробку структурної схеми програми зазвичай виконують методом покрокової деталізації.

Структурні схеми *пакетів програм* розробляють для кожної програми пакета окремо, оскільки організація програм в пакети не передбачає передачі керування між ними.

Компонентами структурної схеми програмної системи або програмного комплексу можуть слугувати програми, підсистеми, бази даних, бібліотеки ресурсів та ін.

Зазвичай для програмних систем розробляється функціональна схема, яка дає повніше уявлення про проєктоване програмне забезпечення з точки зору взаємодії його компонентів між собою та із зовнішнім середовищем.

Метод покрокової деталізації передбачає, що деякі функції мають бути деталізовані для виконання певних алгоритмічних дій.

Функціональна схема

Функціональна схема (ГОСТ 19.701-90) – це схема взаємодії компонентів програмного забезпечення з описом інформаційних потоків, складу даних в потоках та зазначенням використовуваних файлів і пристроїв.

Для зображення функціональних схем використовують спеціальні позначення, встановлені стандартом. Функціональні схеми більш інформативні, ніж структурні.

Метод покрокової деталізації при складанні алгоритмів

Метод покрокової деталізації реалізує спадний підхід до програмування і передбачає покрокову розробку алгоритму. Можна виділити такі етапи.

1. Створюється опис програми в цілому. Визначаються основні логічні кроки, необхідні для вирішення завдання, навіть якщо поки невідомо, як їх виконати. Ці логічні кроки можуть відображати різні фізичні способи вирішення або можуть бути зручними груповими іменами для тих дій, виконання яких описується досить нечітко. Послідовності кроків, необхідних

для вирішення задачі, записуються на звичайній мові або на псевдокодi.

2. У загальних термінах деталізується опис кроків, введених на етапі.

3. У деталізований опис може входити позначення циклічних структур, у той час як дії в середині циклів можуть, як і раніше, залишатися неясними. Отже, виконуються тільки загальні ескізи складних дій.

4. На цьому і наступних рівнях у вигляді послідовних ітерацій виконуються ті ж дії, що описані на етапі 2. При кожній новій ітерації уточнюються деталі, що залишилися неясними після попередніх ітерацій, і створюються більш точні описи. З кожною виконаною ітерацією невизначені деталі стають все простішими, так що на якомусь етапі можуть бути повністю описані.

5. Розробка завершена: у модульному вигляді отримано опис необхідної програми. Переклад цього опису в програму на конкретній мові програмування має бути досить простим завданням.

Структурні карти Константайна

Методика структурних карт використовується на етапі проектування ПЗ для того, щоб продемонструвати, як програмний продукт виконує системні вимоги. Найбільш часто застосовуються дві техніки: структурні карти Константайна (Constantine), призначені для опису відносин між модулями, і структурні карти Джексона (Jackson), призначені для опису внутрішньої структури модулів.

Структуру програмної системи складають модулі, які в будь-якій мові програмування мають такі загальні характеристики:

- модуль має ім'я, за яким до нього можна звертатися як до єдиного фрагменту;
- модуль складається з безлічі операторів мови програмування, записаних послідовно;
- модуль може приймати і/або передавати дані як параметри в послідовності, що викликається, або пов'язувати дані через фіксовані клітинки або загальні області.

Структурні карти Константайна являють собою модель відносин між модулями програми. Вузли структурних карт відповідають модулям і областям даних, потоки зображують міжмодульні зв'язки. На діаграмі спеціальними вузлами зображуються циклічні та умовні виклики модулів, а потоки проходять через ці спеціальні вузли. Потоки, що зображують міжмодульні зв'язки за даними та керування, також зображуються на діаграмі спеціальними вузлами, а стрілками вказуються напрямки потоків.

На рис. 1 наведені основні компоненти структурних карт Константайна.

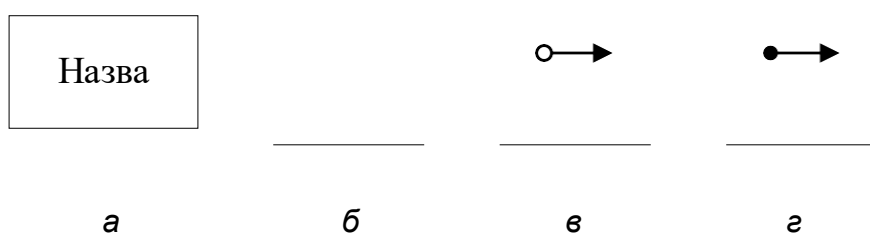


Рисунок 1 – Елементи структурних карт Константайна:

а – модуль, б – виклик модуля; в – зв'язок за даними; г – зв'язок управління

Структурні карти Джексона

Техніка структурних карт Джексона заснована на методі структурного програмування Джексона, який виявляє відповідність між структурою потоків даних і структурою програми. Основну увагу в методі сконцентровано на відповідності вхідних і вихідних потоків даних.

Структури на діаграмах Джексона будуються з чотирьох основних компонентів:

- операція (блок-кодів, що має один вхід і один вихід);
- слідування (послідовне виконання операцій зліва направо);
- вибір (виконання однієї з операцій залежно від виконання умови);
- ітерація (багаторазове виконання блоку).

Порядок виконання роботи

1. На основі розробленого технічного завдання виконати аналіз функціональних і експлуатаційних вимог до програмного продукту.

2. Визначити основні технічні рішення та діаграми потоків даних для вирішуваного завдання.

3. Визначити діаграми «сутність–зв’язок», якщо програмний продукт містить базу даних.

4. Визначити функціональні діаграми та специфікації процесів (задач).

5. Додати словник термінів.

6. Оформити результати, використовуючи MS Office або MS Visio, у вигляді ескізного проекту.

7. На основі ескізного проекту розробити уточнені рішення щодо функцій (завдань) розроблюваного програмного продукту, структур даних, технічних засобів, алгоритмів вирішення.

8. Розробити структурну схему і функціональну схему програмного продукту.

9. Скласти структурну схему у вигляді структурних карт Константайна і структурних карт Джексона.

10. Оформити результати, використовуючи MS Office або MS Visio.

Зміст звіту

1. Назва та мета роботи.

2. Розроблені ескізний та технічний проекти.

3. Письмові відповіді на контрольні запитання.

Контрольні питання

1. Призначення специфікацій програмного продукту.

2. Призначення і суть функціональних діаграм під час аналізу вимог до програмного продукту.

3. Методологія і моделі SADT.

4. Моделі ERM.

5. Призначення і суть діаграм переходів станів.

6. Що має описувати технічний проект програмної системи?

7. Призначення і суть структурної схеми ПЗ.

8. Призначення і суть функціональної схеми ПЗ.

9. Призначення і суть структурних карт Константайна.

10. Призначення і суть структурних карт Джексона.

Література: [1–6].

Лабораторна робота № 3

Тема. Структурний підхід до програмування. Стадія «Реалізація. Розробка програмного засобу»

Мета: розробити діючу версію програми та оформити текст програми відповідно до рекомендацій щодо стилю програмування та оформлення програм.

Короткі теоретичні відомості

Поняття технологічності програмного забезпечення

Під технологічністю розуміють якість проекту програмного продукту, від якого залежать трудові та матеріальні витрати на його реалізацію і наступні модифікації.

Хороший проект порівняно швидко і легко кодується, тестується, налагоджується і модифікується. Технологічність програмного забезпечення визначається пропрацьованістю його моделей, рівнем незалежності модулів, стилем програмування і ступенем повторного використання кодів.

Чим краще опрацьована модель розроблюваного програмного забезпечення, тим чіткіше визначені підзадачі і структури даних, що зберігають вхідні, проміжну і вихідну інформацію, тим простіше їх проектування і реалізація і менше ймовірність помилок, для виправлення яких знадобиться істотно змінювати програму.

Чим вище незалежність модулів, тим їх легше зрозуміти, реалізовувати, модифікувати, а також знаходити в них помилки і виправляти їх.

Стиль програмування, під яким розуміють стиль оформлення програм та їх «структурність», також істотно впливає на читабельність програмного коду і кількість помилок програмування.

Збільшення ступеня повторного використання кодів передбачає як використання раніше розроблених бібліотек підпрограм або класів, так і уніфікацію кодів. Причому для цього критерію ситуація не така однозначна, як у попередніх випадках: якщо ступінь повторного використання кодів підвищується штучно (наприклад, шляхом розробки «суперуніверсальних» процедур), то технологічність проекту може істотно знизитися.

Стиль оформлення програми

З точки зору технологічності хорошим вважають стиль оформлення програми, який полегшує її сприйняття як самим автором, так і іншими програмістами, яким, можливо, доведеться її перевіряти або модифікувати.

Враховуючи, що будь-яку програму неодноразово доведеться переглядати, слід дотримуватися гарного стилю написання програм.

Стиль оформлення програми включає:

- правила іменування об'єктів програми (змінних, функцій, типів, даних тощо);
- правила оформлення модулів;
- стиль оформлення текстів модулів.

Правила іменування об'єктів програми. Вибираючи імена програмних об'єктів слід дотримуватися таких правил:

- ім'я об'єкта має відповідати його змісту;
- якщо дозволяє мова програмування, можна використовувати символ «_» для візуального розділення імен, що складаються з декількох слів;
- необхідно уникати близьких за написанням імен.

Правила оформлення модулів

Кожен модуль має починатися заголовком, який, як мінімум, містить:

- назву модуля;
- короткий опис його призначення;
- короткий опис вхідних і вихідних параметрів із зазначенням одиниць виміру;

- список використовуваних (що викликаються) модулів;
- короткий опис алгоритму (методу) та / або обмежень;
- ПШБ автора програми;
- ідентифікаційну інформацію (номер версії і/або дату останнього коригування).

Стиль оформлення текстів модулів

Стиль оформлення текстів модулів визначає використання відступів, пропусків рядків і коментарів, що полегшують розуміння програми.

Як правило, пропуски рядків і коментарі використовують для візуального розділення частин модуля. Для таких мов, як C++ та Java, використання відступів дозволяє прояснити структуру програми: зазвичай додатковий відступ позначає вкладення операторів мови. Коментувати необхідно цілі виконання тих чи інших дій, а також групи операторів, пов'язані спільною дією, тобто коментарі мають містити деяку додаткову (неочевидну) інформацію.

Для мов низького рівня, наприклад, Асемблера, стиль, який полегшує розуміння, запропонувати важче. У цьому випадку може виявитися доцільним коментувати і блоки операторів, і кожен оператор.

Порядок виконання роботи

1. Спираючись на розроблений в попередній лабораторній роботі технічний проект розробити діючу версію програми.
2. Оформити текст програми відповідно до рекомендацій щодо стилю оформлення.

Зміст звіту

1. Назва та мета роботи.
2. Текст, блок–схема алгоритма та скріншоти розробленої програми.

Контрольні питання

Задаються індивідуально щодо тексту програми.

Література: [4, 5].

2 КРИТЕРІЇ ОЦІНЮВАННЯ ЗНАНЬ СТУДЕНТІВ

У 11-му семестрі студенти виконують 6 лабораторних робіт. Загальна кількість балів, яку отримують студенти за виконання та захист лабораторних робіт, становить 18 балів (максимально по 3 бали на кожен лабораторну роботу).

Шкала оцінювання знань студентів: національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	
		Для іспиту, курсового проекту (роботи), практики	Для заліку
90–100	A	Відмінно	Зараховано
82–89	B	Добре	
74–81	C		
64–73	D	Задовільно	
60–63	E		
35–59	FX	Незадовільно з можливістю повторного складання	Не зараховано з можливістю повторного складання
0–34	F	Незадовільно з обов'язковим повторним вивченням навчальної дисципліни	Не зараховано з обов'язковим повторним вивченням навчальної дисципліни

СПИСОК ЛІТЕРАТУРИ

1. Андон Ф. И. Основы инженерии качества программных систем / Ф. И. Андон. – Издательский Дом «Академперіодика», 2007. – 673 с.
2. Вендров А. М. Проектирование программного обеспечения экономических информационных систем : учебник / А. М. Вендров. – М. : Финансы и статистика, 2005. – 544 с.
3. Вендров А. М. Проектирование программного обеспечения экономических информационных систем / А. М. Вендров. – М. : Финансы и статистика, 2003. – 352 с.
4. Гагарина Л. Г. Технология разработки программного обеспечения : учебное пособие / Л. Г. Гагарина, Е. В. Кокорева, Б. Д. Виснадул. – М. : ИД «ФОРУМ», 2008. – 400 с.
5. Иванова Г. С. Технология программирования : учебник / Г. С. Иванова. – М. : Издательство МГТУ им. Н. Э. Баумана, 2002. – 243 с.
6. Плєскач В. Л. Інформаційні технології та системи / В. Л. Плєскач, Ю. В. Рогущина, Н. П. Кустова. – К. : «КНИГА», 2004. – 520 с.
7. Технологія створення програмних продуктів : лабораторний практикум для студентів напряму підготовки 6. 050101 «Комп'ютерні науки» денної та заочної форм навчання / уклад. В. А. Литвинов, М. В. Гладка, О. А. Хлобистова. – К. : НУХТ, 2014. – 86 с.
8. ДСТУ 2226-93 Автоматизовані системи. Терміни та визначення.
9. ДСТУ 3330-96 Інформаційні технології. Система стандартів з баз даних. Еталонна модель керування даними.
10. ДСТУ 3626-97 Базові програмно-технічні комплекси локального рівня для розосереджених автоматизованих систем керування технологічними процесами. Загальні вимоги.
11. ДСТУ ISO 11442-2:2004 Документація на технічну продукцію. Автоматизоване оброблення технічної інформації. Частина 2. Документація оригіналів.

12. ДСТУ ISO 11442-3:2004 Документація на технічну продукцію. Автоматизоване оброблення технічної інформації. Частина 3. Стадії процесу проектування продукції.

13. ДСТУ ISO 11442-4:2004 Документація на технічну продукцію. Автоматизоване оброблення технічної інформації. Частина 4. Системи керування та пошуку документів.

14. РД 50-680-88 Керівний документ по стандартизації. Методичні вказівки. Автоматизовані системи. Основні положення.

15. РД 50-34.698-90 Автоматизовані системи. Вимоги до змісту документів.

Методичні вказівки щодо виконання лабораторних робіт з навчальної дисципліни «Керування розробкою програмних систем» для студентів денної форми навчання зі спеціальності 123 – «Комп'ютерна інженерія» освітнього ступеня «Магістр». Частина I

Укладачі: д. т. н., проф. М. І. Гученко,

к. т. н. П. П. Костенко,

асист. Н. Л. Сохін

Відповідальний за випуск зав. кафедри «Комп'ютерні та інформаційні системи»
проф. А. В. Луговой

Підп. до др. _____. Формат 60×84 1/16. Папір тип. Друк ризографія.

Ум. друк. арк. _____. Наклад _____ прим. Зам. № _____. Безкоштовно.

Видавничий відділ
Кременчуцького національного університету
імені Михайла Остроградського
вул. Першотравнева, 20, м. Кременчук, 39600