

ЛАБОРАТОРНА РОБОТА № 1. Етапи розробки програмного забезпечення при структурному підході до програмування. Стадія «Технічне завдання»

Мета роботи: ознайомитися з правилами написання технічного завдання.

Теоретична частина. Розробка технічного завдання

Технічне завдання являє собою документ, в якому сформульовані основні цілі розробки, вимоги до програмного продукту, визначені терміни і етапи розробки та регламентований процес випробувань. У розробці технічного завдання беруть участь як представники замовника, так і представники виконавця. В основі цього документа лежать вихідні вимоги замовника, аналіз передових досягнень техніки, результати виконання науково-дослідних робіт, передпроектних досліджень, наукового прогнозування і т. п.

Порядок розробки технічного завдання

Розробка технічного завдання виконується в наступній послідовності. Перш за все, встановлюють набір функцій, що виконуються, а також перелік і характеристики вихідних даних. Потім визначають перелік результатів, їх характеристики і способи подання.

Далі уточнюють середовище функціонування програмного забезпечення: конкретну комплектацію і параметри технічних засобів, версію операційної системи і, можливо, версії і параметри іншого встановленого програмного забезпечення, з яким взаємодіятиме програмний продукт.

У випадках, коли програмний додаток збирає та зберігає деяку інформацію або включається в управління яким-небудь технічним процесом, необхідно також чітко регламентувати дії програми у разі збоїв обладнання та енергопостачання.

1. Загальні положення

- 1.1. Технічне завдання оформляють відповідно до ГОСТ 19.106-78 на аркушах формату А4 і А3 по ГОСТ 2.301 - 68, як правило, без заповнення полів аркуша. Нумери аркушів (сторінок) проставляють в верхньої частини аркуша над текстом.

- 1.2. Лист затвердження і титульний лист оформляють відповідно до ГОСТ 19.104-78. Інформаційну частину (анотацію і зміст), лист реєстрації змін допускається в документ не включати.
- 1.3. Для внесення змін і доповнень до технічного завдання на наступних стадіях розробки програми або програмного додатку випускають доповнення до нього. Узгодження і затвердження доповнення до технічного завдання проводять у тому ж порядку, що встановлений для технічного завдання.
- 1.4. Технічне завдання повинно містити наступні розділи:
 - Вступ;
 - Найменування та область застосування;
 - Підстава для розробки;
 - Призначення розробки;
 - Технічні вимоги до програми або програмного додатку;
 - Техніко-економічні показники;
 - Стадії і етапи розробки;
 - Порядок контролю та приймання;
 - Додатки.

Залежно від особливостей програми або програмного додатку допускається уточнювати зміст розділів, вводити нові розділи або об'єднувати окремі з них. При необхідності допускається в технічне завдання включати додатки.

2. Зміст розділів

2.1. Вступ повинен включати коротку характеристику області застосування програми або програмного додатку, а також об'єкта (наприклад, системи), в якому передбачається її використання. Основне призначення вступу - продемонструвати актуальність даної розробки і показати, яке місце ця розробка займає в ряду подібних.

2.2. У розділі «Найменування та область застосування» вказують найменування, коротку характеристику області застосування програми або програмного додатку і об'єкта, в якому використовують програму або програмний додаток.

2.3. У розділі «Підстава для розробки» повинні бути зазначені:

- документ (документи), на підставі якого ведеться розробка. Таким документом може служити план, наказ, договір і т. п.

- організація, що затвердила цей документ, і дата його затвердження;
- найменування і (або) умовне позначення теми розробки.

2.4. У розділі «Призначення розробки» повинно бути вказано функціональне та експлуатаційне призначення програми або програмного додатку.

2.5. Розділ «Технічні вимоги до програми або програмного додатку» повинен містити такі підрозділи:

- вимоги до функціональних характеристик;
- вимоги до надійності;
- умови експлуатації;
- вимоги до складу і параметрів технічних засобів;
- вимоги до інформаційної та програмної сумісності;
- вимоги до маркування та упаковки;
- вимоги до транспортування і зберігання;
- спеціальні вимоги.

- У підрозділі «Вимоги до функціональних характеристик» повинні бути вказані вимоги до складу виконуваних функцій, організації вхідних і вихідних даних, тимчасові характеристики і т. п.
- У підрозділі «Вимоги до надійності» повинні бути вказані вимоги до забезпечення надійного функціонування (забезпечення сталого функціонування, контроль вхідної і вихідної інформації, час відновлення після відмови і т. п.).
- У підрозділі «Умови експлуатації» повинні бути вказані умови експлуатації (температура навколишнього повітря, відносна вологість і т. п. Для обраних типів носіїв даних), при яких повинні забезпечуватися задані характеристики, а також вид обслуговування, необхідну кількість і кваліфікація персоналу.
- У підрозділі «Вимоги до складу і параметрів технічних засобів» вказують необхідний склад технічних засобів із зазначенням їх технічних характеристик.
- У підрозділі «Вимоги до інформаційної та програмної сумісності» повинні бути вказані вимоги до інформаційних структур на вході і виході, методи вирішення, вихідні коди, мови програмування. При необхідності необхідно забезпечувати захист інформації та програм.

- У підрозділі «Вимоги до маркування та упаковки» в загальному випадку вказують вимоги до маркування програмного додатку, варіанти і способи упаковки.
 - У підрозділі «Вимоги до транспортування і зберігання» повинні бути вказані умови транспортування для програмного додатку, місця зберігання, умови зберігання, умови складування, терміни зберігання в різних умовах.
 - У розділі «Техніко-економічні показники» повинні бути зазначені: орієнтовна економічна ефективність, економічні переваги розробки в порівнянні з кращими вітчизняними і зарубіжними зразками або аналогами.
- 2.6. У розділі «Стадії та етапи розробки» встановлюють необхідні стадії розробки, етапи і зміст робіт (перелік програмних документів, які повинні бути розроблені, узгоджені та затверджені), а також, як правило, терміни розробки і визначають виконавців.
- 2.7. У розділі «Порядок контролю та приймання» повинні бути вказані види випробувань і загальні вимоги до приймання роботи.
- 2.8. У додатках до технічного завдання при необхідності наводять:
- перелік науково-дослідних та інших робіт, що обґрунтовують розробку;
 - схеми алгоритмів, таблиці, описи, обґрунтування, розрахунки та інші документи, які можуть бути використані при розробці;
 - інші джерела розробки.

У випадках, якщо які-небудь вимоги, передбачені технічним завданням, замовник не пред'являє, слід у відповідному місці зазначити «Вимоги не пред'являються».

Порядок виконання роботи

1. Розробити технічне завдання на програмний продукт (див. Варіанти завдань у додатку 1).
2. Оформити роботу відповідно до ГОСТ 19.106-78.
3. Здати і захистити роботу.

Міністерство освіти і науки України
Вінницький національний технічний університет
Кафедра програмного забезпечення

ЗАТВЕРДЖУЮ

«____» _____ 20__ р

XX

ТЕХНІЧНЕ ЗАВДАННЯ

Листів 3

Керівник _____

Виконавець _____

Вінниця, 20__

1. Вступ

Технічне завдання поширюється на розробку програми сортування одновимірного масиву методами «бульбашки», «прямого вибору», Шелла і «швидкого сортування», призначеної для використання школярами старших класів при вивченні курсу шкільної інформатики.

2. Підстава для розробки

1. Програма розробляється на основі навчального плану кафедри «Програмного забезпечення».
2. Найменування роботи:
«Програма сортування одновимірного масиву».
3. Виконавець: компанія XXXXXXXX.
4. Співвиконавці: ні.

3. Призначення

Програма призначена для використання школярами при вивченні теми «Обробка одновимірних масивів» в курсі «Інформатика».

4. Вимоги до програми або програмного додатку

1. Вимоги до функціональних характеристик
 1. Програма повинна забезпечувати можливість виконання таких функцій:
 - введення розміру масиву і самого масиву;
 - зберігання масиву в пам'яті;
 - вибір методу сортування;
 - висновки текстового опису методу сортування;
 - висновки результату сортування.
 2. Вихідні дані:
 - розмір масиву, заданий цілим числом;
 - масив.
 3. Організація вхідних і вихідних даних

Вхідні дані надходять з клавіатури.

Вихідні дані відображаються на екрані і при необхідності виводяться на друк.

2. Вимоги до надійності

Передбачити контроль введеної інформації.

Передбачити блокування некоректних дій користувача при роботі з системою.

Передбачити проміжкове копіювання даних для їх тимчасового збереження

Передбачити наявність «підказок» при роботі із додатком

3. Вимоги до складу і параметрів технічних засобів.

Система повинна працювати на IBM-сумісних персональних комп'ютерах.

Мінімальна конфігурація:

- тип процесора. _____;
- обсяг оперативного запам'ятовуючого пристрої _____ Мб і більше;
- обсяг жорсткого диску _____ Мб.
- Рекомендована конфігурація:
- тип процесора. _____;
- обсяг оперативного запам'ятовуючого пристрої _____ Мб;
- обсяг жорсткого диску _____ Мб.
-

4. Вимоги до програмної сумісності.

Програма повинна працювати під управлінням сімейства операційних систем Win 32 (Windows __/__/__/__/__ і т. п.).

5. Вимоги до програмної документації

1. Програмні модулі повинні бути документовані. Тексти програм повинні містити всі необхідні коментарі.
2. Програмний додаток повинен включати довідкову інформацію про роботу програми, описи методів сортування і підказки учням.
3. До складу супроводжуючої документації повинні входити:
 1. Пояснювальна записка на _____ аркушах, що містить опис розробки.
 2. Керівництво користувача.

Контрольні питання

1. Наведіть етапи розробки програмного забезпечення.
2. Що включає в себе постановка задачі та передпроектні дослідження?
3. Перерахуйте функціональні та експлуатаційні вимоги до програмного продукту.
4. Перерахуйте правила розробки технічного завдання.
5. Назвіть основні розділи технічного завдання.

ЛАБОРАТОРНА РОБОТА № 2.

Структурний підхід до програмування.

Стадія «Ескізний проект»

Мета роботи: навчитися створювати формальні моделі і на їх основі визначати специфікації розроблюваного програмного забезпечення.

Теоретична частина.

Розробка специфікацій

Розробка програмного забезпечення починається з аналізу вимог до нього. У результаті аналізу отримують специфікації розроблюваного програмного забезпечення, будують загальну модель його взаємодії з користувачем або іншими програмами і конкретизують його основні функції.

При структурному підході до програмування на етапі аналізу і визначення специфікацій розробляють три типи моделей: моделі функцій, моделі даних і моделі потоків даних. Оскільки різні моделі описують проектоване програмне забезпечення з різних сторін, рекомендується використовувати відразу кілька моделей, що розробляються у вигляді діаграм, і пояснювати їх текстовими описами, словниками і т. п.

Структурний аналіз передбачає використання наступних видів моделей:

- діаграм потоків даних (DFD - Data Flow Diagrams), описують взаємодію джерел і споживачів інформації через процеси, які повинні бути реалізовані в системі;
- діаграм «сутність-зв'язок» (ERD - Entity-Relationship Diagrams), що описують бази даних розроблюваної системи;
- діаграм переходів станів (STD - State Transition Diagrams), що характеризують поведінку системи в часі;
- функціональних діаграм (методика SADT);
- специфікацій процесів;
- словника термінів.

Специфікації процесів

Специфікації процесів зазвичай представляють у вигляді короткого текстового опису, схем алгоритмів, псевдокод, Flow-форм або діаграм Насс - Шнейдермана.

Словник термінів

Словник термінів являє собою короткий опис основних понять, що використовуються при складанні специфікацій. Він повинен включати

визначення основних понять предметної області, опис структур елементів даних, їх типів і форматів, а також всіх скорочень і умовних позначень.

Діаграми переходів станів

За допомогою *діаграм переходів станів* можна моделювати подальше функціонування системи на основі її попереднього і поточного функціонування. Моделювана система в будь-який заданий момент часу знаходиться точно в одному з кінцевого безлічі станів. З плином часу вона може змінити свій стан, при цьому переходи між станами повинні бути точно визначені.

Функціональні діаграми

Функціональні діаграми відображають взаємозв'язку функцій розроблюваного програмного забезпечення.

Вони створюються на ранніх етапах проектування систем, для того щоб допомогти проектувальнику виявити основні функції і складові частини проектованої системи і, по можливості, виявити і усунути істотні помилки. Для створення функціональних діаграм пропонується використовувати методологію SADT.

Діаграми потоків даних

Для опису потоків інформації в системі застосовуються *діаграми потоків даних* (DFD - Data flow diagrams). DFD дозволяє описати необхідну поведінку системи у вигляді сукупності процесів, що взаємодіють за допомогою зв'язують їх потоків даних. DFD показує, як кожен з процесів перетворює свої вхідні потоки даних у вихідні потоки даних і як процеси взаємодіють між собою.

Діаграми «сутність-зв'язок»

Діаграма сутність-зв'язок - інструмент розробки моделей даних, що забезпечує стандартний спосіб визначення даних і відносин між ними. Вона включає *суті* і взаємозв'язку, що відображають основні бізнес-правила предметної області. Така діаграма не надто деталізована, в неї включаються основні *сутності* і *зв'язку* між ними, які задовольняють вимогам, пропонованим до ІС.

Порядок виконання роботи

1. На основі технічного завдання з лабораторної роботи № 1 виконати аналіз функціональних та експлуатаційних вимог до програмного продукту.
2. Визначити основні технічні рішення (вибір мови програмування, структура програмного продукту, склад функцій ПП, режими функціонування) і занести результат ти в документ, званий «Ескізним проектом» (див. Додаток 4).
3. Визначити діаграми потоків даних для розв'язуваної задачі.
4. Визначити діаграми «сутність-зв'язок», якщо програмний продукт містить базу даних.
5. Визначити функціональні діаграми.

6. Визначити діаграми переходів станів.
7. Визначити специфікації процесів.
8. Додати словник термінів.

Контрольні питання

1. Назвіть етапи розробки програмного забезпечення.
2. Що таке життєвий цикл програмного забезпечення?
3. У чому полягає постановка задачі та передпроектні дослідження?
4. Назвіть функціональні та експлуатаційні вимоги до програмного продукту.
5. Перерахуйте компоненти ескізного проекту.
6. Охарактеризуйте специфікації і моделі.

ЛАБОРАТОРНА РОБОТА № 3. Структурний підхід до програмування. Стадія «Технічний проект»

Мета роботи: вивчити питання проектування програмного забезпечення.

Теоретична частина. Складові технічного проекту

ПРОЕКТ ТЕХНІЧНИЙ - образ наміченого до створення об'єкта, представлений у вигляді його опису, схем, креслень, розрахунків, обґрунтувань, числових показників.

Технічний проект

Мета технічного проекту - визначення основних методів, використовуваних при створенні інформаційної системи, і остаточне визначення її кошторисної вартості.

Технічне проектування підсистем здійснюється відповідно до затвердженого технічного завдання.

Технічний проект програмної системи докладно описує:

- виконувані функції і варіанти їх використання;
- відповідні їм документи;
- структури оброблюваних баз даних;
- взаємозв'язку даних;
- алгоритми їх обробки.

Технічний проект повинен включати дані про обсяги й інтенсивність потоків оброблюваної інформації, кількості користувачів програмної системи, характеристиках обладнання та програмного забезпечення, що взаємодіє з проєктованим програмним продуктом.

При розробці технічного проекту оформляються:

- відомість технічного проекту. Загальна інформація по проекту;
- пояснювальна записка до технічного проекту. Вступна інформація, що дозволяє її споживачеві швидко освоїти дані по конкретному проекту;
- опис систем класифікації та кодування;
- перелік вхідних даних (документів). Перелік інформації, яка використовується як вхідний потік і служить джерелом нагромадження;
- перелік вихідних даних (документів). Перелік інформації, яка використовується для аналізу накопичених даних;
- опис використовуваного програмного забезпечення. Перелік програмного забезпечення і СУБД, які планується використовувати для створення інформаційної системи;

- опис використовуваних технічних засобів. Перелік апаратних засобів, на яких планується робота проектного програмного продукту;
- проектна оцінка надійності системи. Експертна оцінка надійності з виявленням найбільш благополучних ділянок програмної системи та її вузьких місць;
- відомість обладнання та матеріалів. Перелік обладнання і матеріалів, які будуть потрібні в ході реалізації проекту.

Структурна схема

Структурної називають схему, що відображає склад і взаємодія з управління частинами розроблюваного програмного забезпечення. Структурна схема визначається архітектурою розробляється ПО.

Функціональна схема

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

Розробка алгоритмів

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

Структурні карти

Методика структурних карт використовується на етапі проектування ПЗ для того, щоб продемонструвати, яким чином 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. Основна увага в методі сконцентровано на відповідності вхідних і вихідних потоків даних.

Порядок виконання роботи

1. На основі технічного завдання з лабораторної роботи № 1 і специфікацій з лабораторної роботи № 2 розробити уточнення алгоритми програм, складових заданий програмний модуль. Використовувати метод покрокової деталізації.
2. На основі уточнених і доопрацьованих алгоритмів розробити структурну схему програмного продукту.
3. Розробити функціональну схему програмного продукту
4. Уявити структурну схему у вигляді структурних карт Константайна.
5. Уявити структурну схему у вигляді структурних карт Джексона.

6. Оформити результати, використовуючи MS Office або MS Visio у вигляді технічного проекту.
7. Здати і захистити роботу.

Захист звіту з лабораторної роботи

Звіт з лабораторної роботи повинен складатися з:

1. Структурної схеми програмного продукту.
2. Функціональної схеми.
3. Алгоритму програми.
4. Структурної карти Константайна.
5. Структурної карти Джексона.
6. Закінченого технічного проекту програмного модуля. Захист звіту з лабораторної роботи полягає в пред'явленні викладачеві отриманих результатів (на екрані монітора), демонстрації отриманих навичок і відповідях на питання викладача.

ЗАТВЕРДЖУЮ

Керівник (замовника ІС)

Особистий підпис Розшифровка підпису _

Друк Дата «» 20__ р

ЗАТВЕРДЖУЮ

Керівник (розробника ІС)

Особистий підпис Розшифровка підпису

Друк Дата «» 20__ р

Ескізний проект на створення інформаційної системи

Система управління базою даних

(найменування виду ІС)

XX

(найменування об'єкта інформатизації)

СУБД «Бібліотека»

(скорочене найменування І С)

На 8 аркушах

Вінниця 20__ р

Зміст

- 1. Відомість ескізного проекту**
- 2. Пояснювальна записка до ескізного проекту**
 - 1. загальні положення**
 - 2. Основні технічні рішення**
 - 3. Рішення за структурою системи**
 - 4. Рішення по режимах функціонування, роботи системи**
 - 5. Рішення за чисельністю, кваліфікації і функцій персоналу**
 - 6. Склад функцій комплексів задач, що реалізуються системою**
 - 7. Рішення по складу програмних засобів, мовам діяльності, алгоритмам процедур і операцій і методів їх реалізації**
- 3. Джерела розробки**

Відомість ескізного проекту

На попередніх стадіях розробки СУБД «Пенсійний Фонд» були складені та затверджені наступні документи:

Технічне завдання на створення інформаційної системи СУБД «Фонд», розроблене на підставі ГОСТ 34.602-89 на написання ТЗ на автоматизовані системи управління від 01.01 ._____ р

Пояснювальна записка до ескізного проекту загальні положення

Даний документ є ескізним проектом на створення Системи Управління Базою Даних для Фонду (СУБД «Бібліотека»).

Перелік організацій, що беруть участь у розробці системи, терміни і стадії розробки, а також її цілі й призначення вказані в технічному завданні на створення інформаційної системи.

Основні технічні рішення

Рішення за структурою системи

СУБД «Бібліотека» представлятиме собою персональну систему управління локальною базою даних, що працює на одному комп'ютері.

Система буде управляти реляційною базою даних, що представляє собою набір пов'язаних між собою таблиць у форматі Paradox, доступ до яких здійснюється за допомогою ключів або індексів. Відомості в одній таблиці можуть відбивати відомості з іншої, і при зміні відомостей в першій таблиці ці зміни негайно відображаються у другій. Таким чином буде досягнута несуперечність даних.

Загальна структура бази даних:

- Анкети організації, які зареєстровані в даний ном Ф:
 - Тип підприємства (Російська організація, Фізична особа, Іноземна організація, Відокремлений підрозділ).
 - Вид підприємства (Адвокати, Бюджетне, Єдиний податок 6%, Єдиний податок 15%, Сільгосппродукція, Служби зайнятості, Фермерське господарство, Інше).
 - Регист ротаційний номер роботодавця в Ф.
 - Свідоцтво: серія, номер.
 - Дата видачі свідоцтва (число_місяць_год).
 - ІПН.
 - КПП.
- Найменування.
- Юридична адреса:
- Поштовий індекс.
- Регіон.
- Район.
- Місто.
- Населений пункт.

- Вулиця.
- Дом.
- Корпус.
- Квартира.
- Адреса постійно діючого органу (при відмінності від юридичної).
- Анкети співробітників цих організацій:
- Прізвище.
- Ім'я.
- По батькові.
- Пол (Ч / Ж).
- Дата народження (Дата).
- Страховий номер.
- Місце народження (Країна, Регіон, Район, Місто, Населений пункт).

Громадянство.

- Адреса реєстрації (Країна, Поштовий індекс, Регіон, Район, Місто, Населений пункт, Вулиця, Будинок, Корпус, Квартира).
- Адреса місця проживання фактичний (Країна, Поштовий індекс, Регіон, Район, Місто, Населений пункт, Вулиця, Будинок, Корпус, Квартира).
- Телефон домашній.
- Телефон службовий.
- Документ (посвідчені. Особистість).
- Дата видачі (дата).
- Ким виданий .
- Дата заповнення (Дата).
- ІПН.
- Відомості про стаж співробітників цих організацій:
- Страховий номер.
- Прізвище.
- Ім'я.
- По батькові.
- Дата народження.

Територіальні умови проживання на

- Таблиця періодів роботи з наступною структурою:
 - Початок періоду (дата).
 - Кінець періоду (дата).
 - Вид діяльності (робота, служба соцстрах, догляд-діти, безробітного, реабіліт, догляд - інвд, профзаб, перегляд).
 - Найменування організації.
 - Посада.

- Територіальні умови.

Рішення по режимах функціонування, роботи системи

СУБД «Бібліотека» функціонуватиме в режимі одного, а також буде здатна:

- переглядати записи бази даних (у тому числі і за допомогою фільтрів);
- додавати нові записи;
- видаляти записи;
- при вході в систему буде запитуватися пароль.

Рішення за чисельністю, кваліфікації і функцій персоналу АС

Зазначені рішення повинні задовольняти вимогам, наведеним у технічному завданні на розробку системи.

Склад функцій комплексів задач, що реалізуються системою

Автоматизована система повинна виконувати наступні функції:

- зробити запис про пенсійному посвідченні;
- видалити інформацію про пенсійному посвідченні;
- видати довідку про всіх пенсійних посвідченнях;
- зареєструвати нове підприємство в Ф;
- видалити підприємство з бази даних;
- видати довідку про всі підприємства, зареєстрованих у Ф;
- підрахувати пенсію для працівників підприємств на підставі стажу;
- видати довідку про пенсійних заощадження працівника.

Рішення по складу програмних засобів, мовам діяльності, алгоритмам процедур і операцій і методів їх реалізації

Для реалізації АС буде використовуватися середу програмування Boland Delphi 7.0 і мова програмування Object Pascal.

Для підрахунку пенсії буде використовуватися наступний алгоритм.

Спочатку визначається стажевий коефіцієнт пенсіонера. Він покладається рівним 0,55 за загальний трудовий стаж до поточної дати не менше 25 років чоловікам і 20 років жінкам. За кожний повний рік стажу понад зазначений стажевий коефіцієнт збільшується на 0,01, але не більше ніж на 0,20.

Потім визначається відношення заробітку пенсіонера до середньомісячної заробітної плати в країні. Цей заробіток може бути взятий за цей відліковий період або за будь-які 60 місяців роботи підряд, або той, з якого було обчислено пенсію на момент реформи. Середньомісячна зарплата в країні береться за той же самий період.

Джерела розробки

Даний документ розроблявся на підставі ГОСТ 34.698-90 на написання ТЗ на автоматизовані системи управління від 01.01.1992 р

СКЛАЛИ

Посада виконавця
Прізвище ім'я по батькові
Підпис
Дата «" 2007 р
Посада виконавця
Прізвище ім'я по батькові
Підпис
Дата «" 2007 р
Посада виконавця
Прізвище ім'я по батькові
Підпис
Дата «" 2007 р

Контрольні питання

1. Назвіть етапи розробки програмного забезпечення.
2. У чому полягає проектування програмного забезпечення?
3. Перерахуйте компоненти технічного проекту.
4. Охарактеризуйте структурний підхід до програмування.
5. З чого складаються структурна і функціональна схеми?
6. Охарактеризуйте метод покрокової деталізації при складанні алгоритмів програм.
7. Наведіть поняття псевдокоду.
8. У чому полягає методика Константайна?
9. У ніж полягає методика Джексона?

Додатки

Додаток 1

Варіанти завдань

Лабораторні роботи № 1-5 виконуються для одного і того ж варіанту.

1. Розробити програмний модуль «Облік успішності студентів». Програмний модуль призначений для оперативного обліку успішності студентів в сесію деканом, заступниками декана і співробітниками деканату. Відомості про успішність студентів повинні зберігатися протягом всього терміну їх навчання і використовуватися при складанні довідок про прослухані курси й додатків до диплому.
2. Розробити програмний модуль «Особові справи студентів». Програмний модуль призначений для отримання відомостей про студентів співробітниками деканату, профкому та відділу кадрів. Відомості повинні зберігатися протягом усього терміну навчання студентів і використовуватися при складанні довідок і звітів.
3. Розробити програмний модуль «Рішення комбінаторно-оптимізаційних задач». Модуль повинен містити алгоритми пошуку циклу мінімальної довжини (задача комівояжера), пошуку найкоротшого шляху і пошуку мінімального зв'язує дерева.
4. Розробити додаток \ Vindows «Органайзер». Додаток призначений для запису, зберігання та пошуку адрес і телефонів фізичних осіб та організацій, а також розкладу, зустрічей тощо. Додаток призначений для будь-яких користувачів комп'ютера.
5. Розробити додаток Windows «Калькулятор». Додаток призначений для будь-яких користувачів і повинно містити всі арифметичні операції (здотриманням пріоритетів) і бажано (але НЕ обов'язково) кілька математичних функцій.
6. Розробити програмний модуль «Кафедра», який містить відомості про співробітників кафедри (ПІБ, посада, науковий ступінь, дисципліни, навантаження, громадська робота, сумісництво та ін.). Модуль призначений для використання співробітниками відділу кадрів і деканату.
7. Розробити програмний модуль «Лабораторія», що містить відомості про співробітників лабораторії (ПІБ, стать, вік, сімейний стан, наявність дітей, посада, науковий ступінь). Модуль призначений для використання співробітниками профкому і відділу кадрів.
8. Розробити програмний модуль «Автосервіс». При записи на обслуговування заповнюється заявка, в якій зазначаються ПІБ власника, марка автомобіля, вид роботи, дата прийому замовлення і вартість ремонту. Після виконання робіт роздруковується квитанція.

9. Розробити програмний модуль «Облік порушень правил дорожнього руху». Для кожної автомашини (і її власника) в базі зберігається список порушень. Для кожного порушення фіксується дата, час, вид порушення і розмір штрафу. При оплаті всіх штрафів машина видаляється з бази.

10. Розробити програмний модуль «Картотека агентства нерухомості», призначений для використання працівниками агентства. У базі містяться відомості про квартирах (кількість кімнат, поверх, метраж та ін.). При надходженні заявки на обмін (купівлю, продаж) проводиться пошук відповідного варіанту. Якщо такого немає, клієнт заноситься в клієнтську базу і оповіщається, коли варіант з'являється.

11. Розробити програмний модуль «Картотека абонентів АТС». Картотека містить відомості про телефонах та їх власників. Фіксує заборгованості з оплати (абонентської і погодинної). Вважається, що погодинна оплата місцевих телефонних розмов вже введена.

12. Розробити програмний модуль «Авіакаса», який містить відомості про наявність вільних місць на авіамаршрути. У базі повинні міститися відомості про номер рейсу, екіпажі, типі літака, дату і час вильоту, а також вартості авіаквитків (різного класу). При надходженні заявки на квитки програма проводить пошук відповідного рейсу.

13. Розробити програмний модуль «Книжковий магазин», який містить відомості про книги (автор, назва, видавництво, рік видання, ціна). Покупець оформляє заявку на потрібні йому книги, якщо таких немає, він заноситься в базу і оповіщається, коли потрібні книги надходять в магазин.

14. Розробити програмний модуль «Автостоянка». У програмі міститься інформація про марку автомобіля, його власника, дату і час в'їзду, вартості стоянки, знижки, заборгованості з оплати та ін.

15. Розробити програмний модуль «Кадрове агентство», який містить відомості про вакансії і резюме. Програмний модуль призначений як для пошуку співробітника, що відповідає вимогам керівників фірми, так і для пошуку підходящої роботи.

Примітка. При розробці програми не обмежуватися функціями, наведеними у варіанті, додати кілька своїх функцій. Обов'язкове використання структурного і модульного підходів до програмування. Бажано використання об'єктного підходу.

Література

1. *Іванова Г. С.* Технологія програмування. М.: Вид - во МГТУ ім. Баумана, 2 002.
2. *Вельбицький І. В.* Технологія програмування. Київ, 1984.
3. *Грейді Буч, Джеймс Рамбо, Айвар Джекобсон.* UML керівництво користувача. М.: ДМ К, 2 000.
4. *Дін Леффінгуелл, Дон Уїдріг.* Принципи роботи з вимогами до програмного забезпечення. М.: Вільямс, +2002.
5. *Девід С. Платт.* Знайомство з Microsoft .NET. М.: Російська редакція, 2001.
6. *Лунаев В. В.* Проектування програмних засобів. М.: Вища школа, 1990.
7. *Майерс Г.* Мистецтво тестування програм. М.: Фінанси і статистика, 1982.
8. *Брукс Ф.* Міфічний людино - місяць, або Як створюються програмні системи. СПб.: Символ-Плюс, 1999..
9. *Роберт Дж. Орберг.* СОМ + технологія. Основи і програмування М.: Вільямс, 2000. 478 с.
10. *Є. Yourdon.* Who Has the Right Stuff? // Software Development. 1997. № 9.
11. *Gause, Weinberg.* Exploring Requirements: Quality Before Design, 1989
12. *Аджиев В.* // Відкриті системи. 1998. № 1.
13. *Батенко Л. Я.* // Менеджмент і менеджер. 2003. № 3.
14. *Маріо Аниселла* // Computerworld. 2000. № 16 - 17.
15. *Davis.* Fifteen Principles of Software Engineering // IEEE Software. 1994. Vol. 11. № 6. P. 94 - 101.
16. *Boehm A.* Spiral Model of Software Development and Enhancement // Computer. 1988. Vol. 21. № 5. P. 61 - 72.
17. *Good-Enough Software* // Application Development Strategies. 1996. № 1.
18. *Алістер Коуберн, Лорі Вільямс.* Парне програмування: переваги і недоліки.
19. *Cusumano M., Selby R.* How Microsoft Builds Software // Communications of the ACM. 1997. Vol. 40. № 6. P. 53 - 61.
20. *Cusumano M., Selby R.* Microsoft's Weaknesses in Software Development // American Programmer. 1997. Vol. 10. № 10.
21. *Knepper S., Ichbiah D.* The Making Microsoft. Prima Publishing, 1993

22. www.maxkir.com.
23. <http://www.martinfowler.com/articles/designDead.html>.
24. <http://www.softportal.com/articles/itemJ:xt.php?id=78> Кент Бек.
25. <http://ergl.rU/archive/cs/tp/01.htm#P4> Лекції ВМиК. Технологія програмування.
26. *Ожегов С. І.* Словник російської мови. М.: Радянська Енциклопедія, 1975.
27. Радянський енциклопедичний словник. М.: Радянська Енциклопедія, 1979.
28. Політехнічний словник / Г л. ред. акад. А. Ю. Ишлинский. 2. е изд. М.: Радянська Енциклопедія, 1980.
29. *Терпи* Кватрані. Візуальне моделювання з допомогою Rational Rose 2002 і UML. М.: Вільямс, 2003.
30. <http://www.zsksoft.ru/zsync-doc/usage/team-development.htm>.
31. *Voas J.* The Software Quality Certification Triangle. Crosstalk, Nov. 1998. P 12-14.
32. <http://www.spc-consulting.ru/standart/cmm.htm>.
33. <http://www.interface.ru/home.asp?artId=3987>.
34. <http://www.intuit.ru/department/se/testing/>.
35. *McCabe T. J. f Butler Ch. W* Design complexity measurement and testing Communications of the ACM. 32, 12 (Dec. 1989). P. 1415-1425.
36. *Макконнелл С.* Досконалий код. СПб.: Питер, 2006.
37. *Жоголев Е. А.* Введення в технологію програмування (конспект лекцій). М.: ДІАЛОГ - МГУ, 1994..
38. *Уоли Б.* Програмування на Бейсике. М.: Радио і зв'язок, 1988.
39. *Калянов Г. Н.* Консалтинг при автоматизації підприємств (підходи, методи, засоби). М.: Сінтег, 1 997.
40. *Страуструп Б.* Мова програмування C++. Київ: ДіаСофт, 1993.
41. Моделі та структури даних / В. Д. Далека, А. С. Дерев'янка, О. Г. Кравець, Л. Є. Тімановського. Харків: ХДПУ, 2000.
42. *Х'юз Дж., Мічтом Дж.* Структурний підхід до програмування. М.: Світ, 1980. С. 29 - 71.
43. *Жоголев Е. А.* Технологічні основи модульного програмування // Програмування. 1980. № 2. С. 44 - 49.
44. *Holt R. C.* Structure of Computer Programs: A Survey // Proceedings of the IEEE. 1975. 63 (6). P. 879 - 893.
45. *Майерс Г.* Надійність програмного забезпечення. М.: Світ, 1980. С. 92-113.
46. *Зелковец М. Шоу А., Геннон Дж.* Принципи розробки програмного забезпечення. М.: Світ, 1982. С. 65 - 71.
47. *Дав У., Дейкстра Е., Хоор К.* Структурний програмування. М.: Світ, 1975. С. 7-19.

48. *Леоненко А. В.* Самовчитель UML. СПб.: BHV, 2 006.
49. *Крачтен Ф.* Rational ü nified Process: введення: третє видання. Видавництво Addison-Wesley Professional, +2003.
50. *Басс Л., Клементс П., Рік Кацман Р.* Практична архітектура програмного забезпечення: друге видання. Addison Wesley, 2003.
51. Object Management Group Inc. Специфікація уніфікованої мови моделювання OMG версія 1.5. Документ номер 3. 03-01. Березня 2003.
52. *Мак-Говерн Джеймс та ін.* Практичне керівництво по архітектурі корпорацій. Prentice Hall, +2004.
53. *Вендров А. М.* CASE - технології. Сучасні методи і кошти проектування інформаційних систем. М.: Фінанси і статистика, 1998..
54. *Пушник А. Ю.* Введення в системи управління базами даних. Джерело матеріалу - Пушник А. Ю. Введення в системи управління базами даних. Частина 2. Нормальні форми відносин і транзакції: навч. п особин. Изд-е Башкирського ун-ту, 1999.
55. Структурний аналіз при розробці програмного забезпечення систем реального часу / В. А. Матяш, А. В. Ніка залізничний рів, В. А. Путілов, А.Е. Федоров, В. В. Фильчаков. Апатити: КФ ПетрГУ, 1997..
56. <http://se.math.spbu.ru/>.
57. *Зачароване В.* Розробка ПО: оцінка результату // Комп'ютерне огляд. 2006. 21 верес.
58. *Albrecht AJ* Measuring Application Development Productivity? Proceedings of the Joint SHARE / GUIDE / IBM Application Develop-, ment Symposium? 1979. October. P 83 - 92.
59. *Боем Б.* Інженерне проектування програмного забезпечення. М.: Радио і зв'язок, 1985.
60. Visual C ++ 6. Керівництво розробника.
61. *Брауде Е. Д.* Технологія розробки програмного забезпечення. СПб.: Питер, 2004.
62. *Зиндер Е. 3.* Бізнес - реінжиніринг і технології системного проєктіваніє. М.: Центр інформаційних технологій, +1996.
63. *Каієр С., Фолк Д., Нгуєн Є. К.* Тестування програмного забезпечення: пров. з англ. Київ: ДіаСофт, 2000.