ПРИВАТНЕ АКЦІОНЕРНЕ ТОВАРИСТВО «ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД «МІЖРЕГІОНАЛЬНА АКАДЕМІЯ УПРАВЛІННЯ ПЕРСОНАЛОМ»

МАУП

МЕТОДИЧНІ РЕКОМЕНДАЦІЇ

ЩОДО ЗАБЕЗПЕЧЕННЯ САМОСТІЙНОЇ РОБОТИ СТУДЕНТІВ

з дисципліни

«Емпіричні методи програмної інженерії»

(для бакалаврів)

Київ 2019

Підготовлено професором кафедри комп'ютерних інформаційних систем та технологій
А.С. Дудніком
Затверджено на засіданні кафедри комп'ютерних інформаційних систем та технологій
(Протокол № 7 від 16 січня 2019 р.)

Схвалено Вченою радою Міжрегіональної Академії управління персоналом

Дуднік А.С. Методичні рекомендації щодо забезпечення самостійної роботи студентів з дисципліни "Емпіричні методи програмної інженерії" (для освітньо-кваліфікаційного рівня «бакалавр»). — К.: МАУП, 2019— 24 с.

Методичні рекомендації містять пояснювальну записку, тематичний план, зміст дисципліни, питання щодо самостійного вивчення студентами та самоконтролю, теми рефератів, питання для самоконтролю та список рекомендованої літератури.

© ПрАТ «ВНЗ «Міжрегіональна Академія управління персоналом» (МАУП), 2019

ПОЯСНЮВАЛЬНА ЗАПИСКА

Предметом цієї дисципліни є вивчення сучасних інтегрованих середовищ розробки програмних продуктів, сучасних програмних технологій зв'язування окремо розроблених компонентів, стратегій розробки проектів (на прикладі RUP), моделювання інформаційних систем (на прикладі UML), CASE-засобів розробки (на прикладі Rational Rose), стратегій маркетингу програмних продуктів, методів оцінки витрат та економічної доцільності розробки програмних продуктів.

До самостійної роботи відноситься також вивчення та освоєння методичних вказівок до лабораторних робіт і вивчення додаткової літератури, пов'язаної з виконанням цих робіт.

Тільки постійне самостійне навчання дає можливість якомога ближче підійти до вершини знань певної галузі, оволодіти такою сумою знань і вмінь, які б дали змогу заявити про себе як про професіонала. Студент, який хоче якомога краще оволодіти професією, має добре розуміти: на занятті викладач подає основи знань, навчає, як учити, виділяє ті ключові істини дисципліни, які пробуджують у молодої людини потяг до поглиблення й удосконалення усіх знань. Збагачення загальною сумою знань, накопичених людством, розширення загального світогляду, усвідомлення наявної перспективи щодо реалізації певних знань є основним мотивом до сумлінного відношення до навчання. Самостійна навчальна діяльність студента буде лише тоді результативною, коли вона ґрунтуватиметься на внутрішній потребі. Виховання відповідної здатності у студента потребує чіткого узгодження процесу самоосвіти з цілями навчання та виховання.

Згідно з державними стандартами навчальний матеріал навчальної дисципліни, передбачений робочим навчальним планом для засвоєння студентом в процесі самостійної роботи, виноситься на підсумковий контроль поряд з навчальним матеріалом, який опрацьовувався при проведенні навчальних занять. Самостійна робота студента над засвоєнням навчального матеріалу з конкретної дисципліни може виконуватися у бібліотеці вищого навчального

закладу, навчальних кабінетах, комп'ютерних класах (лабораторіях), а також у домашніх умовах. Самостійна робота студента повинна бути спланована, організаційно і методично спрямована як особиста творча праця без прямої взаємодії з викладачем. Навчальний час, відведений для самостійної роботи, регламентується робочим навчальним планом і повинен згідно з Болонською декларацією становити не менше 50% загального обсягу навчального часу студента, відведеного для вивчення конкретної дисципліни. У необхідних випадках ця робота проводиться відповідно до заздалегідь складеного графіка, що гарантує можливість індивідуального доступу студента до потрібних дидактичних засобів. Графік доводиться до відома студентів на початку поточного семестру. При організації самостійної роботи студентів із використанням складного обладнання чи устаткування, складних систем доступу до інформації (наприклад, комп'ютерних баз даних, систем автоматизованого проектування тощо) передбачається можливістю отримання необхідної консультації або допомоги з боку фахівця.

Самостійна навчальна діяльність студента може здійснюватись через:

запам’ятовування певної інформації за рахунок уважного слухання і конспектування лекцій; активної роботи під час практичних занять;

роботу над конспектами лекцій, планами практичних занять;

опрацювання літературних джерел (конспектування самостійно вивченого матеріалу, реферування);

роботу з каталогами звичайних і електронних бібліотек, інформаційно-пошуковими сервісами Internet;

вивчення навчального матеріалу за паперовими та електронними підручниками, навчальними посібниками, практикумами тощо;

опрацювання матеріалу за першоджерелами, науковою і спеціальною літературою;

підготовку доповідей, рефератів, написання курсових робіт; пошукову і науково-дослідну діяльність;

самотестування.

Самостійна робота студента під час лекції. Лекційний матеріал призначається для спрямування студентів у найбільш раціональному напрямі щодо вивчення навчальної дисципліни і акцентуванні уваги на найбільш складних, вузлових питаннях навчальної дисципліни. Належне ведення конспекту під час лекції сприяє збереженню необхідної інформації та дає студенту змогу в подальшому проаналізувати її. За умови подання лекційного матеріалу в усній формі одночасно засвоюється до 20% інформації. Викладання інформатики в комп'ютерних класах або в аудиторіях, обладнаних мультимедійним обладнанням (наприклад, мультимедійним проектором або сенсорним екраном), водночас з демонстрацією студентам прийомів роботи з користувальницьким інтерфейсом програми дозволяє підвищити рівень засвоєння лекційного матеріалу до (50 - 60)%.

Робота над конспектами лекцій, планами практичних занять. При підготовці до практичних занять студент має спиратися на складений ним конспект лекції. При опрацюванні матеріалу лекції слід зіставити законспектований матеріал з планом практичного заняття, що міститься у методичних матеріалах для практичних занять або у навчально-методичному комплексі. Якщо у конспекті бракує матеріалу з окремих питань лекції або недостатньо розкриті деякі питання практичного заняття, або вони винесенні на самостійне опрацювання, студент повинен звернутися до рекомендованих підручників, навчальних посібників і відповідних методичних матеріалів. Підготовку для практичного заняття краще за все здійснювати з використанням ПЕОМ зі встановленим на ньому відповідним програмним забезпеченням.

Вивчення навчального матеріалу за підручниками, навчальними посібниками, методичними вказівками, опрацювання матеріалу за першоджерелами, науковою і спеціальною літературою. Працювати із підручниками, навчальними посібниками, методичними вказівками, практикумами, науковою і спеціальною літературою незалежно від типу їхнього носія (паперового чи електронного) необхідно таким чином, щоб отримати максимум теоретичних знань і навичок. При роботі з цими джерелами студент насамперед повинен

ознайомитись з їх змістом, щоб визначити чи необхідно опрацьовувати це джерело і чи має воно відношення до навчального курсу, що вивчається, і тільки після цього визначити послідовність його опрацювання і відібрати необхідний для вивчення матеріал з цього джерела (глави, розділи тощо). В разі роботи з інтерактивними електронними джерелами слід використовувати можливості навігації за документом, що надаються сучасними програмами, призначеними для читання електронних документів відповідних форматів (*MS Word, Adobe Reader, Adobe Acrobat* та інш.) і, особливо, переваги гіпертекстової технології подачі навчального матеріалу, а саме - за допомогою гіперпосилань знаходити відповіді на поставлені питання. При опрацюванні матеріалу необхідно з'ясувати суть питання, що вивчається, не уникаючи при цьому визначення суті незрозумілих чи незнайомих слів, термінів. Саме інтерактивні гіпертекстові електронні джерела (довідки в складі програмних продуктів, електронні посібники та словники) дозволяють конкретизувати терміни та визначення як найшвидше. При вивченні матеріалу необхідно аналізувати прочитане, порівнюючи з прослуханою та законспектованою лекцією, робити логічні висновки, позначати незрозумілі положення з метою їх подальшого з'ясування на практичному занятті. Бажано відпрацювати зручну для себе певну систему позначень (позначки на полях конспекту, підкреслення маркерами різних кольорів, доповнення конспекту альтернативними формулюваннями та посиланнями на інші джерела тощо) та фіксації опрацьованого матеріалу. Сучасні текстові редактори (в першу чергу *MS Word*) надають можливість створення електронного конспекту з примітками, виносками, коментарями та його роздруківки. Для самостійного поглибленого вивчення навчального матеріалу студенту слід звертатися до наукової та спеціальної літератури, яка може бути і не зазначеною в навчально-методичному комплексі. Використання самостійно отриманих відомостей як у навчанні, так і на практиці є, безперечно, цінним здобутком діяльності студента на шляху формування свого професійного потенціалу.

Робота з бібліотечними фондами та дистанційними джерелами з метою пошуку необхідної інформації. Знання основ системного програмування та операційних систем відносяться до базової підготовки сучасного спеціаліста з інформаційних технологій. З позицій випереджаючої освіти навчання тільки за конспектом лекцій і основною літературою, вказаною у навчальній програмі, є недостатнім. У більшості випадків належна підготовка вимагає вмінь швидко знаходити та опрацювати необхідний матеріал за першоджерелами, науковою і спеціальною літературою та коректно цитувати знайдене. Перелік такої літератури, як правило, наводиться у навчально-методичному комплексі навчальної дисципліни. Тому завдання студента зводиться до самостійного знаходження цих матеріалів шляхом пошуку у паперових або електронних фондах бібліотек, а також у різноманітних файлових архівах, базах даних та базах знань, доступ до яких здійснюється за допомогою відповідних сервісів *Internet* (в основному – *Word Wide Web, FTP* та *UseNet newsgroups*).

Для пошуку документу використовуються різні його ознаки. В першу чергу це - реквізити документу (УДК. Автор(и). Заголовок опису. Основний заголовок: відомості, що відносяться до заголовку/Відомості про відповідальність. - Відомості про видання (в тому числі URL – адреса Web - документу або Ftp - файлу). - Місце видання, дата видання. - Об'єм.). УДК - це універсальна десяткова класифікація будь-яких офіційних видань по всьому світу. Відповідні довідники видаються багатьма мовами і постійно оновлюються. В Україні у 2006 р. Книжковою палатою України імені Івана Федорова видано "Універсальну десяткову класифікацію. Зміни та доповнення." Випуск 4" в паперовому варіанті. Довідкова база УДК постійно нарощується за рахунок електронних видань. Знання УДК дозволяє швидко знайти необхідне джерело за систематичним бібліотечним каталогом. Наприклад, УДК видань з інформаційних технологій починається з 004.

Коли код УДК невідомий, то необхідно звернутися до алфавітного каталогу бібліотеки і за назвою джерела або прізвищем та ініціалами автора знайти відповідний бібліотечний шифр джерела.

Якщо ж студент здійснює наукове дослідження вибраної проблеми, готує наукову доповідь або виступ на конференції і йому не відомі реквізити джерела або саме джерело, то слід зробити пошук у систематичному бібліотечному каталозі. Завдання студента полягає у пошуку необхідної галузі (підгалузі), що охоплює розшукувану інформацію, а потім у межах цієї галузі (підгалузі) — картки з необхідним джерелом і бібліотечним шифром. У подальшому студент повинен оформити бібліотечне замовлення на літературу встановленого зразка, до якого внести шифр знайденого джерела та усі необхідні реквізити. Робота з електронними фондами в цьому варіанті значно ефективніша, оскільки у розвинутих бібліотеках облік літератури ведеться в середовищах систем управління базами даних, за допомогою яких пошук потрібної інформації здійснюється найефективніше.

Сервіси мережі *Internet* надають унікальні можливості знаходження літературних джерел у географічно віддалених фондах та архівах, а також шляхом участі у мережевих конференціях, де можна отримати відповіді та поради щодо питань з розшукуваної інформації. Для доступу до *Internet* - ресурсів необхідно знати їх мережеву адресу. Оскільки *Internet* постійно оновлюється і розвивається, в ньому немає єдиного каталогу, змісту або наочного покажчика ресурсів. Проте в *Internet* існують різні інформаційно-пошукові системи, що допомагають користувачам знайти те, що їм потрібно. Це в першу чергу тематичні каталоги і так звані пошукові машини. Тематичні (наочні) каталоги - це інформаційно-довідкові системи, підготовлені вручну редакторами цих систем на основі інформації, зібраної на серверах *Internet*. Інформація в цих системах розподіляється за тематичними розділами відповідно до певної ієрархії. На верхньому рівні розділів зібрані загальні категорії (наприклад, "Інтернет", "Бізнес", "Мистецтво", "Освіта" тощо), а нижній рівень складають посилання на конкретні Web-сторінки або інші інформаційні ре-

сурси. Для швидкого переходу до потрібного розділу тематичного каталогу можна скористатися вбудованою системою автоматичного пошуку за ключовими словами. Для цього в рядку запиту слід ввести ключове слово (поєднання слів), клацнути ***Пошук***, і система повідомить, чи є відповідний розділ в її каталозі і запропонує в нього перейти, минувши всі проміжні розділи. Рекомендуємо використовувати каталоги: http://www.yahoo.com, http://www.portal.edu.ru , http://www.ipl.org .

Пошукові системи є складними інформаційно-довідковими системами, що автоматично генерують спеціальні бази даних за допомогою спеціальних мережевих програмам-роботів по всій мережі *Internet*, і надаючи у відповідь на запит користувача посиланнями на різні *Internet* -ресурси. Запит здійснюється за певною процедурою (на певній мові), яка може відрізнятися в різних системах, проте в спрощеному вигляді вона зводиться до того, що користувач вводить в спеціальному полі (або в декількох полях) ключові слова, та/або словосполучення, що найточніше відображають суть проблеми.

До загальних положень мов запитів відносяться:

Ключові слова можна вводити у відповідне поле пошукової системи поодинці, послідовно звужуючи пошук, або ж вводити відразу декілька слів, розділяючи їх пробілами або комами. Регістр не має значення.

Режим пошуку "AND" ("І") означає, що будуть знайдені тільки ті дані, де зустрічається кожне з ключових слів.

При використанні режиму "OR" ("АБО") результатом пошуку будуть всі дані, де зустрічається хоч би одне ключове слово.

Використовуйте знаки "+" і "-" перед ключовим словом. Щоб виключити документи, де зустрічається певне слово, поставте перед ним мінус. І навпаки, щоб певне слово обов'язково було присутнє в документі, поставте перед ним плюс. Зверніть увагу на те, що між знаком і словом не повинно бути пропуску.

Якщо Ви хочете виключити яке-небудь слово з пошуку, поставте перед ним знак "-". Наприклад: "+захист -Excell".

За замовчуванням програма шукає всі дані, де зустрічається введене вами слово. Наприклад, при запиті "редактор" будуть знайдені слова "редактор", "текстовий", "графічний", "газети", "головний" і багато інших. Знак оклику перед або після ключового слова означає, що будуть знайдені тільки слова точно відповідні запиту (наприклад, "текстовий! редактор!").

Також корисно запам'ятати і використовувати при пошуку наступні прийоми.

Якщо для пошуку потрібно ввести словосполучення, укладіть його в лапки.

Якщо Ви пишете все слово рядковими буквами, будуть знайдені всі варіанти його написання; якщо Ви вказали хоч би одну букву в шуканому слові прописною, то система шукатиме тільки такі варіанти.

Якщо Ви хочете знайти не текст, а яке-небудь зображення, то можна користуватися словом image. Наприклад, image:sea дасть список сторінок із зображенням моря.

Якщо слово, яке Ви шукаєте, зустрічається в різних контекстах, можна виключити слова, які зустрічаються в непотрібному контексті. Наприклад, вказати аргумент пошуку +Celeron +Price +UA -USA.

Перевіряйте орфографію. Якщо пошук не приніс результатів, можливо, при введенні Ви допустили помилку.

Використовуйте синоніми. Якщо список знайдених сторінок дуже малий або не містить корисних сторінок, спробуйте змінити слово. Наприклад, замість "реферати", можливо, більше підійде "курсові роботи" або "твори".

Якщо один із знайдених документів ближче до шуканої теми, чим інші, клацніть ***Знайти схожі документи***. Це посилання розташовано під короткими описами знайдених документів. Система проаналізує сторінку і знайде документи, схожі на ту, що Ви вказали.

Подібних систем в *Internet* значно більше, чим тематичних каталогів. Серед пошукових систем існують як обширні з тематики метапошукові системи, так і вузькоспеціалізовані. Найбільш відомі з них:

http://www.google.com, http://www.altavista.com, http://www.askjeeves.com, http://www.lycos.com, http://www.sciseek.com, http://www.msn.com, http://meta.ua http://www.rambler.ru, http://www.yandex.ru, http://www.aport.ru, http://www.metabot.ru, http://newsgroups.langenberg.com, uk.wikipedia.org, www.bukinist.agava.ru.

Матеріали щодо методів підвищення ефективності пошуку інформації в *Internet* містяться у статях: http://www.yandex.ru/info/search.html , http://www.searchengines.ru/,

http://www.zodchiy.ru/links/search/, http://www.citforum.ru/internet/search/index.shtml , http://websearch.report.ru/ , http://www.kokoc.com/search-engines/index.shtml , http://www.zhurnal.ru/search-r.shtml.

Самостійна робота має такі складові і форми їх оцінювання:

підготовка та власне аудиторна робота під час практичних і лабораторних занять. Результати її оцінюються під час поточного контролю;

виконання самостійних робіт у формі есе, рефератів з конкретних проблем та складання письмових звітів на електронних або паперових носіях або усних доповідей;

опрацювання програмного матеріалу зі змістового модулю та оцінка її результатів під час проміжного контролю;

виконання письмової контрольної роботи або тестування;

звіт про проходження практики;

звіт про науково-дослідну роботу, результати якої можуть бути використані при написані випускної роботи і за рішенням кафедри опубліковані.

МЕТОДИЧНІ ВКАЗІВКИ ДО ПІДГОТОВКИ, НАПИСАННЯ ТА ЗАХИСТУ РЕФЕРАТУ

Реферат є складовою частиною вивчення дисципліни.

Дані завдання підготовлені відповідно курсу: “Емпіричні методи програмної інженерії ” для бакалаврів.

Мета – допомогти студентам засвоїти теоретичні знання в галузі програмної інженерії, розвинути і удосконалити навички використання даних знать для розробки програмних продуктів. Оформлення й захист рефератів повинні сприяти активному засвоєнню нового матеріалу, виробленню у студентів уміння комплексного використання суміжних дисциплін при вирішенні практичних питань.

Структура реферату

Орієнтовна структура і обсяги реферату наведені нижче.

План (розділи)	**Обсягу сторінках (приблизно)**	**Короткий зміст (що потрібно висвітлити)**
Вступ	До однієї	Мета, загальна характеристика, визначення номеру варіанту завдання
Назва кожного питання відповідно реферату	1 – 2, загальний об'єм роботи у межах 20-30	Викладення суті питання з приведенням прикладів та посилань на літературні джерела
Висновки	До однієї	Прикладне значення
Список літератури	До однієї	
Додатки	До трьох	Якщо є

Загальний обсяг роботи не повинен перевищувати 20-30 сторінок машинописного тексту, надрукованого через 2 інтервали, рукописне викладення тексту не повинно перевищувати 18-24 сторінок шкільного зошита.

Виконання та оформлення реферату

Студент повинен виконати реферат, розкривши історичні та технологічні посилки даної проблеми, відповідаючи на всі питання як теоретичного плану, так і описати технологію розв‘язання практичної задачі, якщо такі передбачені рефератом.

Відповіді на теоретичні питання потребують ретельної роботи з літературою. Крім виписок і конспектування з літературних джерел, наприклад, із Internet, студент повинен зробити висновки. Робота повинна бути виконана самостійно. В тексті реферату потрібно давати посилання на використану літературу. У висновках у цілому з реферату розглядають питання економічної доцільності і практичного застосування сучасних інформаційних технологій та обчислювальної техніки в області операційних систем та системного програмування.

Реферат слід оформляти на стандартних аркушах паперу, зброшурованих у папку. Усі аркуші мають бути пронумеровані. На титульній сторінці необхідно вказати назву вищого навчального закладу, факультет, спеціальність, дисципліну, курс, групу, а також прізвище, ініціали та номер залікової книжки.

На першій сторінці повинні бути представлені розрахунок варіанта контрольної роботи та питання варіанту і проставлені номери сторінок, на яких викладено даний матеріал. На останній сторінці студент підписує роботу і ставить дату. У кінці роботи необхідно подати використану літературу. Зшита папка повинна бути вкладена в поліетиленовий файл та містити дискету з повним текстом, графікою і т.п. набраного варіанту реферату.

Вибір варіанту реферату

Кожний студент отримує окреме завдання для виконання КР згідно з варіантом Z, котрий обчислюється по формулі:

$Z = mod_{20}(NZK + PR – 2000) + 1$,

де NZK - номер залікової книжки (студентського квитка) студента;

PR-поточний рік отримання завдання,

Наприклад, NZK = 398, PR =2002, тоді

$Z = mod_{10} (398+2002 – 2000)+1 = mod_{10} (400) + 1 = 0+1=1$.

Отже тут Z=1.

Зауваження 1. Обчислення варіанту повинно бути у вступі до контрольної роботи.

Зауваження 2. Для довідки : $mod_a b$ дорівнює залишку від ділення b на a.

Увага!

Неправильно оформлена робота повертається без перевірки на доо-формлення. Робота, виконана не за своїм варіантом, підлягає переробці.

ІНДИВІДУАЛЬНО-КОНСУЛЬТАЦІЙНА РОБОТА

Індивідуально-консультативна робота з дисципліни здійснюється у формі консультацій за графіком (одна консультація на два тижні). На консультаціях студентам надаються пояснення з виконання самостійної роботи, підготовки до практичних занять, перевірка та захист завдань, винесених на поточний контроль тощо.

Мета вивчення дисципліни:

1. Чітке розуміння парадигми циклу розробки складних корпоративних і серійних програмних систем, зокрема важливість етапу проектування.

2. Опанування студентами основними сучасними засобами розробки програмних продуктів.

3. Мати ґрунтовну теоретичну базу у сфері сучасної програмноїінжерії.

4. Опанування підходами до формування успішних стратегій їх розповсюдження та маркетингу.

5. Розуміти особливості маркетингу програмних продуктів.

В результаті самостійного вивчення навчальної дисципліни «Емпіричні методи програмної інженерії» студент повинен:

1. Уміти класифікувати програмні системи за рівнем складності, оцінки вартості та трудомісткості.

2. Представляти опис систем в термінах мови моделювання UML.

3. Мати достатньо повне уявлення про особливості групової (командної) роботи над програмними проектами.

4. Набути практичних навичок роботи з будь-якою CASE-системою від проекту до прототипу системи.

5. Уміти оформлювати програмну документацію на всіх етапах циклу розробки та впровадження системи.

Теми для самостійного вивчення дисципліни

	Назва розділу, теми курсу
	Змістовий модуль 1. Загальні питання програмної інженерії
1.	Сучасні програмні системи, сучасна парадигма циклу розробки програмного продукту
2.	Сучасні технології та інструментальні засоби розробки програмних продуктів
	Змістовий модуль 2. Моделювання, проектування та експлуатація сучасних програмних систем
3.	Методології моделювання та проектування програмних систем
4.	Особливості менеджменту розробок і супроводу програмних систем. Маректинг програмних продуктів

Теми рефератів за модулем 1

1. Програмна інженерія – як сучасна галузь знань.
   Література [1, 2,4,6]
2. Етапи розвитку програмної інженерії.
   Література [2,3,14,12]
3. Сучасний підхід до розробки та експлуатації програмних систем.
   Література [4,5,11]
4. Парадигми розробки програмних систем.
   Література [2,4,5,7]
5. Сучасні візуальні засоби розробки програмних продуктів.
   Література [1,3]
6. Сучасні теорії клієнт-серверних застосувань.

Література [6,8,9]

7. Сучасні графічні мови програмування

Література [9,8,10]

8. Технології і засоби побудови сучасних корпоративних систем

Література [1,3,6]

9. Web-орієнтовані програмні засоби.

Література [2,5,3]

10. Десктопні програмні засоби.

Література [5,6,11]

Теми рефератів за модулем 2

1. Сучасні методології моделювання програмних систем.

Література [1-3]

2. Технології і засоби проектування програм.

Література [1-3]

3. Середовища, продукти і засоби, що найчастіше використовуються при розробці коробкових програмних продуктів.

Література [1-3,5,7]

4. CASE-системи. Зв'язок .Net із CASE-системами.

Література [1-3,7,10]

5. Роль і місце баз даних у сучасних програмних комплексах.

Література [1-3,9,11]

6. Порівняння малих "настольних" баз даних із корпоративними серверними базами даних.

Література [1-3,12,13]

7. Огляд MS SQL Server. Механізми транзакцій. Універсальні інтерфейси доступу до баз даних.

Література [2,3,5,7]

8. Технологія MS ADO.

Література [2,3,5,7]

Питання для самоконтролю

1. Якими питаннями займається програмна інженерія?
2. Особливості у розробці великих програмних проектів.
3. Основні етапи повного циклу розробки програмного продукту.
4. Які завдання вирішуються на етапі дослідження проблеми?
5. Що враховується при визначенні економічної доцільності розробки програмного продукту?
6. Що потрібно робити при визначенні структури системи, що проектується?
7. Чи можливо не лише інтуїтивне визначення якості програмної системи?
8. Чи можливо керувати якістю програмного продукту?
9. Що та які спеціалісти роблять під час безпосереднього програмування компонентів?
10. Як і ким здійснюється інтеграція окремих компонентів в єдиний продукт?
11. Що означає термін верифікація системи (проекту)?
12. Які завдання вирішуються під час тестування програмних продуктів?
13. Охарактеризуйте основні джерела помилок в ході розробки системи.
14. Які вам відомі методи тестування систем?
15. Для чого використовують засоби документування результатів тестування?
16. Сутність валідації системи.
17. Яка роль відводиться питанню повторного використання об’єктів (систем) у програмній інженерії?
18. Які є типи програмних продуктів з точки зору їх розміру?
19. Які є типи програмних продуктів з точки зору способу їх розпов-

сюдження?

20. Чи завжди розробка настольного продукту потребує меншої кваліфікації, ніж корпоративного? обґрунтуйте.

21. Що таке корпоративні продукти? опишіть основні риси таких продуктів.

22. Особливості систем, що розробляються за окремим замовленням, порівняно із серійними системами?

23. Що таке web-орієнтовані системи?

24. Які завдання можуть вирішуватися за допомогою web-орієнтованих систем?

25. Особливості впровадження web-орієнтованих систем.

26. Проблеми захисту інформації у web-орієнтованих системах і як вони вирішуються.

27. Як розроблялися програмні продукти до появи об’єктно-орієнтованих методів розробки?

28. Найпоширеніші зараз об’єктно-орієнтовані промислові технології.

29. Опишіть основні ознаки net-технології.

30. Як використовувати net-об’єкти разом з com об’єктами?

31. Особливості інсталяції net-об’єктів.

32. Як здійснюється контроль за версіями net-об’єктів?

33. Навіщо потрібні інтегральні середовища розробки?

34. Найпоширеніші промислові інтегральні середовища розробки.

35. Опишіть основні риси інтегрального середовища, з яким ви працюєте.

36. Дайте порівняння vb.net (c#.net, c++.net тощо) із іншими інтегральними середовищами розробки.

37. Як створити проект і скомпілювати модуль, що виконується, у середовищі vb.net (c#.net, c++.net тощо)?

38. Як виявляти помилки у програмі за допомогою середовища

vb.net (c#.net, c++.net тощо)?

39. Зв'язок інтегральних середовищ розробки із case-системами.

40. Роль і місце баз даних у сучасних програмних системах.

41. Чи існують типи систем, які не потребують баз даних, якщо "так", які саме типи?

42. Які моделі клієнт/сервер ви знаєте?

43. Що таке інтегральна цілісність даних і якими засобами вона забезпечується?

44. Що таке транзакції і навіщо вони потрібні?

45. У чому проблема масштабування програмних систем, зокрема баз даних?

46. Охарактеризуйте універсальний інтерфейс доступу до баз даних ms ado і як він працює?

47. Охарактеризуйте корпоративну систему керування базами даних, з якою ви працюєте чи вивчаєте (зокрема, комп'ютерна платформа, модель даних, інтерфейси доступу, системи-оболонки розробки клієнтських програм, журналізація подій, питання масштабування баз тощо).

48. Що таке моделювання інформаційної системи? його особливості.

49. Навіщо потрібно моделювання інформаційної системи та які проблеми вирішуються шляхом моделювання?

50. Які є сучасні методи моделювання?

51. Які основні риси мови моделювання uml?

52. Як в uml моделюється поведінка системи?

53. Хто ще, окрім розробників програмних систем, може використовувати мову uml і для чого?

54. Яким чином опис мовою uml залежить від типу програмної системи?

55. Які існують зараз та існували раніше технології розробки про-

грамних систем?

56. Навіщо потрібно дотримуватися певної стандартизованої технології при розробці програмних систем?

57. Що таке технологія rup і які її основні риси?

58. Основні етапи розробки згідно з rup.

59. Основні технологічні процеси згідно з rup.

60. Порівняйте підхід “водоспаду” та ітераційний підхід.

61. Порівняйте rup як продукт і як технологію.

62. Які є сучасні case засоби для розробки програмних систем і на яких технологіях вони базуються?

63. Як виконується трансформація проекту у програмну систему?

64. Що таке rational rose і які основні ознаки цього продукту?

65. Як за допомогою rational rose створити макет системи у vb.net (c#.net, c++.net тощо)?

66. Як поєднані між собою rational rose і мова моделювання uml?

67. Які риси програмного продукту як товару?

68. Чим суттєво відрізняється програмний продукт від інших видів товарів?

69. Що таке коробкові програмні продукти?

70. Як визначити економічну доцільність при розробці коробкового програмного продукту?

71. Які середовища найчастіше використовуються при розробці коробкових програмних продуктів?

72. Особливості розповсюдження коробкових програмних продуктів.

73. Яка повинна бути документація до коробкових програмних продуктів?

74. Особливість корпоративних програмних продуктів?

75. як визначити економічну доцільність при розробці корпоративного програмного продукту?

76. Які середовища найчастіше використовуються при розробці корпоративних програмних продуктів?
77. Особливості впровадження корпоративних програмних продуктів.
78. Яка повинна бути документація до корпоративних програмних продуктів?
79. Як організується взаємодія між розробниками та замовниками при створенні корпоративних програмних продуктів?
80. Які ви можете навести рекомендації щодо підбору складу колективу розробників програмної системи?
81. Як оцінюються витрати на розробку окремих компонентів програмної системи?
82. Як оцінюються витрати на інтеграцію програмної системи з іншими системами (зокрема, власного виготовлення та сторонніх компаній)?
83. Як визначаються матеріальні витрати при розробці програмної системи (наведіть кілька підходів)?
84. Як визначається загальна економічна доцільність проекту по розробці програмної системи?
85. Сутність маркетингу програмних продуктів.
86. Особливості інтернет маркетингу програмних продуктів.
87. Як здійснюється підбір кадрів про розробці програмного проекту?
88. Як правильно розподіляти роботу між виконавцями згідно з ком понентною структурою системи?
89. Як контролювати виконання етапів робіт?

СПИСОК ЛІТЕРАТУРИ

Основна

1. Бабенко Л. П., Лавріщева К. М. Емпіричні методи програмної інженерії. — К.:
   Знання, 2001. — 269 с.
2. Калянов Г. Н. CASE-технологии. Консалтинг при автоматизации бизнес-процессов. — М.: Горячая линия-Телеком, 2000. — 320 с.
3. Новоженов Ю. В. Объектно-ориентированные технологии разработки сложных программных систем. — М., 1996.
4. Маклаков С. В. Case-средства разработки информационных систем. — М.: Диалог-Мифи, 2001. — 304 с.
5. Платт Д. С. Знакомство с Microsoft. Net. — М.: Издат.-торг. дом “Русская редакция”, 2001. — 240 с.
6. Троелсен Э. C# и платформа .Net. Библиотека программиста. — СПб.: Питер, 2004. — 796 с.
7. Мамаев Е. В. MS SQL Server 7.0: Проектирование и реализация баз данных. — СПб.: БХВ-Санкт-Петербург, 2000. — 416 с.
8. Крачтен Ф. Введение в Rational Unified Process. — М.: Издат. дом “Вильямс”, 2002. — 240 с.
9. Буч Г., Рамбо Д., Джекобсон А. Язык UML. Руководство пользователя. — М.: ДМК Пресс, 2001. — 432 с.
10. Державний стандарт України. Основні напрямки оцінювання та відбору CASE-інструментів. ДСТУ 3919–1999. 2000 р.

Додаткова

11. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на C++. — СПб.: Невский диалект. — 1999.
12. Фронковяк Д., Гарсиа М., Уолен Э. Руководство администратора MS SQL Server. — М.: Издат.-торг. дом “Русская редакция”,

2000. — 672 с.

13. Microsoft Corporation. Разработка распределенных приложений на Microsoft Visual Basic 6.0. — М.: Издат.-торг. дом “Русская редакция”, 2000. — 400 с.
14. Трофимов С. А. CASE0-технологии. Практическая работа в Rational Rose. — М.: Бином, 2002. — 284 с.
15. Клоков Ю. К., Папушин В. К., Хамитов Р. Р. Методы повышения надежности программного обеспечения // Зарубежная радиоэлектроника. — 1984. — № 6. — С. 3–22.
16. Інтернет-сайт http://www.rational.com
17. Інтернет-сайт http://www.idef.com