

МІЖРЕГІОНАЛЬНА
АКАДЕМІЯ УПРАВЛІННЯ ПЕРСОНАЛОМ



МАУП

**МЕТОДИЧНІ РЕКОМЕНДАЦІЇ
ЩОДО ЗАБЕЗПЕЧЕННЯ САМОСТІЙНОЇ
РОБОТИ СТУДЕНТІВ
з дисципліни
“МОВИ ОБ’ЄКТНО-ОРІЄНТОВАНОГО
ПРОГРАМУВАННЯ”
(для бакалаврів)**

МАУП

Київ

ДП «Видавничий дім «Персонал»

2009

Підготовлено професором кафедри інформатики та інформаційних технологій *С. Г. Карпенком*

Затверджено на засіданні кафедри інформатики та інформаційних технологій (протокол № 19 від 29.02.08)

Схвалено Вченою радою Міжрегіональної Академії управління персоналом

Карпенко С. Г. Методичні рекомендації щодо забезпечення самостійної роботи студентів з дисципліни “Мови об’єктно-орієнтованого програмування” (для бакалаврів). – К.: ДП «Вид. дім «Персонал», 2009. – 24 с.

Методичні рекомендації містять пояснювальну записку, теми самостійної роботи, теми рефератів, питання для самоконтролю та співбесіди, тестові завдання, а також список літератури.

Призначена для методичного забезпечення самостійної роботи студентів денної форми навчання, які вивчають змістові модулі “Основи програмування на мові С++”, “Засоби і методи Об’єктно-орієнтованого програмування в С++” навчальної дисципліни “Мови об’єктно-орієнтованого програмування”.

© Міжрегіональна Академія управління персоналом (МАУП), 2009
© ДП «Видавничий дім «Персонал», 2009

ПОЯСНЮВАЛЬНА ЗАПИСКА

Метою дисципліни “Мови об’єктно-орієнтованого програмування” є ознайомлення студентів з концепцією об’єктно-орієнтованого програмування (ООП) з використанням мов Object Pascal, або C++.

Курс передбачає наявність у студентів певного досвіду програмування мовою Pascal, C.

Завдання курсу: ознайомити студентів з основними поняттями мов ООП як сучасного інструментарію по створенню об’єктно-орієнтованого програмного забезпечення.

На початку курсу розглядається середовище розробки програм (Delphi, Borland C++ або Visual C++), наводиться короткий огляд структурного програмування і розглядаються його відмінності від ООП, вивчаються основні (базові) поняття ООП, особливості вибраної мови ООП, що не мають еквівалента у відповідній базовій мові програмування. Далі основна увага приділяється практичному освоєнню фундаментальних концепцій ООП (інкапсуляції, успадкуванню і поліморфізму) засобами і конструкціями вибраної мови ООП. Наприкінці курсу обговорюються проблеми, що виникають при використанні мов ООП для проектування і розробки великих програмних систем.

Розподіл годин на дисципліну (із загальних 135 годин на дисципліну, 24 виділяється на лекційні заняття, 24 — на практичні та 87 — на самостійну роботу студентів) визначає основне навантаження із засвоєння цієї дисципліни, а саме на самостійне засвоєння матеріалу.

Практична суть освіти полягає в тому, що завдання того, хто навчає, є не стільки передача певних знань студентові, скільки вміння навчити студента, маючи деякий мінімальний обсяг інформації, збільшувати цей обсяг самостійною роботою як на практичних заняттях під керівництвом викладача, так і вдома, в бібліотеці тощо. Лише постійне самостійне навчання дає можливість якомога ближче підійти до глибин знань певної галузі, оволодіти такою сумою знань і вмінь, які б дали змогу заявити про себе як про професіонала.

При організації самостійної роботи студентів передбачається можливість 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, рефератування);
- 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-дослідницьку діяльність;
- самотестування.

Самостійна робота студента під час лекції. Належне ведення конспекту під час лекції сприяє більш уважному засвоєнню нового матеріалу, 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 програмування в комп'ютерних класах з одночасною практичною реалізацією самими студентами теоретичних аспектів програмування у вигляді прикладів і конкретних задач дає можливість підвищити рівень засвоєння лекційного матеріалу до 50–60%.

Робота над конспектами лекцій, планами практичних занять. 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 винесені на самостійне опрацювання, студент повинен звернутися до рекомендованих підручників, навчальних посібників і відповідних методичних матеріалів. Підготовку практичного заняття з програмування слід здійснювати з використанням ПЕОМ зі встановленим на ньому відповідним програмним забезпеченням. За цієї можливості слід використовувати інтерактивні довідкові системи інтегрованих середовищ розробки фірми Borland та Microsoft.

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 їх носія (паперового чи електронного) необхідно таким чином, щоб отримати максимум теоретичних знань і навичок. У разі роботи з інтерактивними електронними джерелами слід використовувати можливості навігації за документом, що надаються сучасними довідковими системами інтегрованих середовищ розробки. Для самостійного поглибленого вивчення навчального матеріалу студенту слід звертатися до наукової та спеціальної літератури, яка може бути і не зазначеною в навчально-методичному комплексі.

Робота з бібліотечними фондами та дистанційними джерелами з метою пошуку необхідної інформації. З позиції випереджучої освіти навчання тільки за контекстом 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 — *World Wide Web*, *FTP* та *UseNet newsgroups*).

Для пошуку документа використовуються різні його ознаки. В першу чергу це — реквізити документа (УДК. Автор(и). Заголовок опису. Основний заголовок: відомості, що належать до заголовка/Відомості про відповідальність. — Відомості про видання (в тому числі URL-адреса Web-документа або Ftp-файла). — Місце видання, дата видання. — Обсяг.). 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 дає можливість швидко

знайти необхідне джерело за систематичним бібліотечним каталогом. Наприклад, УДК видань з інформаційних технологій починається з 004.

Коли код УДК невідомий, 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, 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 всі 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 мережних конференціях, де можна отримати відповіді та поради щодо питань з розшукуваної інформації. Особливо багато інформації в мережі *Internet* з питань програмування. Для доступу до *Internet*-ресурсів необхідно знати їх мережну адресу. Оскільки *Internet* постійно оновлюється і розвивається, в ньому немає єдиного каталога, змісту або наочного покажчика ресурсів. Проте в *Internet* існують різні інформаційно-пошукові системи, що допомагають користувачам знайти те, що їм потрібно. Це насамперед 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 ресурси. Для швидкого переходу до потрібного розділу тематичного каталогу можна скористатися вбудованою системою автоматичного пошуку за ключовими слова-

ми. 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.ua>, <http://www.portal.edu.ru>, <http://www.ipl.org>.

Пошукові системи є складними інформаційно-довідковими системами, що автоматично генеруються на основі даних, які збираються мережними програмами-роботами по всій *Internet*, і дають у відповідь на запит користувача посилання на різні *Internet*-ресурси. Запит здійснюється за певною процедурою (певною мовою), 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 належать:

- Ключові слова можна вводити у відповідне поле пошукової системи поодиночі, послідовно звужуючи пошук, або ж вводити відразу декілька слів, розділяючи їх пробілами або комами. Реєстр 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 усі дані, де зустрічається хоча б 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: “+таблиці -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 такі 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 хоча б 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 припустились помилки.
- 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, ніж 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 статтях: <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>.

Велику популярність мають різні тематичні форуми, на яких користувач може ставити конкретне питання, на яке отримає відповідь (і, може, не одну) від інших членів форуму.

Самостійна робота має такі складові і форми її 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 модуля 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 написанні випускної роботи і за рішенням кафедри опубліковані.

Програма вивчення дисципліни “Мови об’єктно-орієнтованого програмування” складається з таких змістових модулів: “Основи програмування на мові С++”, “Засоби і методи об’єктно-орієнтованого програмування в С++”.

Мета змістового модуля “Основи програмування на мові С++” ознайомлення студентів з конструкціями мови програмування С++, структурою програм, типами даних, структурами даних, типами операцій, управляючими конструкціями, функціями, з модульним принципом побудови програм.

У результаті самостійного вивчення змістового модуля “Основи програмування на мові С++” студенти повинні **знати** основи програмування на мові С++, синтаксичні конструкції, що створюють програму, об’являють типи та структури даних, що об’являють та визначають функції. Крім того, студенти повинні **знати** методи ініціалізації різних типів та структур даних, основні принципи перевантаження функцій.

Мета змістового модуля “Засоби і методи об’єктно-орієнтованого програмування в С++” ознайомлення студентів з методологією об’єктно-орієнтованого програмування (ООП) та створення на їх основі об’єктно-орієнтованого програмного забезпечення з використанням системи програмування Borland С++ Builder.

У результаті самостійного вивчення змістового модуля “Засоби і методи об’єктно-орієнтованого програмування в С++” студенти повинні **знати** основні концепції об’єктно-орієнтованого програмування, поняття об’єктів, базових і похідних класів, абстрактних типів даних; переваження унарних і бінарних операцій; використання інкапсуляції, успадковування, віртуальних функцій, виняткових ситуацій.

ТЕМИ САМОСТІЙНОЇ РОБОТИ

Тема 1. Базові поняття програмування мови ООП

1. Літерали, їх типи. Сталі та змінні, ініціалізація змінних, присвоєння, зведення типів при присвоєнні.
2. Масиви. Показчики. Показчики і масиви. Показчики-константи і показчики-змінні. Показчики і функції. Показчики і рядки. Масиви показчиків на рядки. Показчики на показчики. Посилання. Передавання параметрів за допомогою посилань. Повернення посилань. Посилання на довільні типи даних. Операції розподілу пам’яті new і delete. Ініціалізація виділеної пам’яті. Виділення пам’яті під масиви. Динамічні структури даних.
3. Вирази, оператори (операції), зведення типів у виразах.
4. Інструкції: розгалуження, вибір, цикли з лічильником, за умовою повторення (закінчення), передача управління: (break, continue).
5. Структури. Синтаксис визначення структури. Визначення структурних змінних. Доступ до полів структури. Приклади застосування структур. Вкладені структури. Доступ до полів вкладених структур. Ініціалізація структур.

Теми рефератів

1. Програмне середовище розробника С++ Builder. Інтегроване середовище розробки програм. Елементи інтерфейсу, налагодження параметрів.
2. Програмне середовище розробника С++ Builder. Проект програми. Багатофайлові проекти. Програмні модулі.

Література [1–3]

Література [1–3]

3. Програмне середовище розробника C++ Builder. Налаштування програм, режими покрокового виконання, перегляд локальних змінних.

Література [1–3]

4. Структури у мові C++.

Література [1–3]

Питання для самоконтролю та співбесіди

1. Чи обов'язково вказувати типи параметрів функції в прототипі? Чи обов'язково вказувати імена параметрів?
2. Для чого в програму включаються заголовкові файли? Чи завжди вони потрібні?
3. Який заголовковий файл включається в програму для введення-виведення даних в C++?
4. Які ви знаєте операції відношення? Де вони застосовуються?
5. Які типи даних в мові C++ ви знаєте? Які операції можна виконувати над ними?
6. Що таке компілятор? Особливості компілятора середовища візуального програмування.
7. Що таке директиви препроцесора? Яке їх призначення?
8. Що називають програмним модулем?
9. Опишіть способи ініціалізації змінних різних типів.
10. Як виділити та звільнити область пам'яті для вказівника (pointer).
11. Особливості синтаксису реалізації лінійних структур, розгалуження, вибору та циклів.
12. Як оголосити масив? Опишіть основні операції обробки масивів.
13. Опишіть призначення функцій та особливості їх оголошення.
14. Охарактеризуйте особливості передачі параметрів функції значенням, масивів; замовчувані значення параметрів.
15. Які ви знаєте умовні оператори? Де вони застосовуються?
16. Охарактеризуйте типи циклічних операторів? В яких випадках застосовується той чи інший тип оператора?
17. Що таке структура в C++? Чи може структура містити функції?
18. Який специфікатор доступу до полів структури приймається за замовчуванням?

Тестові завдання

1. Визначіть, що таке заголовкові файли:

- а) файли із записами заголовків функцій;
- б) файли з інтерфейсною частиною модулів програми;
- в) файли із назвами змінних програми;
- г) файли з описом програми.

2. Визначіть, що означає оголошення змінної в програмі:

- а) що ця змінна є ексклюзивною у програмі;
- б) що назва цієї змінної не може бути використана для інших змінних;
- в) виділення пам'яті для зберігання значення цієї змінної;
- г) що з цього місця програми зміна є частиною програми.

3. Зазначте службове слово, яке визначає переліковий тип даних:

- а) enum;
- б) number;
- в) union;
- г) struct.

4. Зазначте тип даних, що визначає службове слово struct:

- а) масив даних з різною структурою;
- б) тип функцій, що можуть мати різну структуру параметрів;
- в) тип даних, що можуть змінювати свою структуру;
- г) складений об'єкт, до якого можуть входити елементи різних типів.

5. Зазначте, що таке покажчик:

- а) змінна, значення якої є адреса комірки пам'яті, в якій зберігається значення деякої змінної;
- б) число, що визначає степінь числа в експоненціальному вигляді;
- в) змінна, в якій зберігається адреса комірки пам'яті степені дійсного числа;
- г) курсор, що показує на напрям виконання програми.

6. Зазначте елементи, які може містити структура:

- а) тільки поля;
- б) тільки функції;
- в) поля і функції;
- г) тільки змінні.

7. Зазначте типи доступу, що можливі для елементів структури:

- а) будь-які;
- б) тільки закриті;
- в) тільки відкриті;
- г) тільки захищені.

8. З наданих варіантів ініціалізації масиву даних зазначте вірний:

- а) `int a[4]=2,4,16,8;`
- б) `int a[4]=(2,4,16,8);`
- в) `int a[4]=[2,4,16,8];`
- г) `int a[4]={2,4,16,8}.`

9. Зазначте оператора, який вірно виділяє пам'ять для покажчика на масив із 4 змінних цілого типу:

- а) `p new int[4];`
- б) `p=new int[4];`
- в) `p new= int[4];`
- г) `new p=int[4].`

10. Зазначте, що означає наведений запис функції: `void find(int &var)`:

- а) що в результаті дії функції `find` до змінної `var` будуть додаватися значення;
- б) що до функції `find` буде передано адресу змінної `var`;
- в) що до функції `find` буде передано значення змінної `var`;
- г) що значення змінної `var` в тілі функції `find` не буде змінюватися.

Тема 2. Об'єктне програмування

1. Принцип інкапсуляції даних. Конструктори, деструктори.
2. Класи і об'єкти, атрибути об'єктів — члени класів, функції класу, функції поза класу, область видимості і права доступу.
3. Інтерфейс класу, реалізація класу; визначення і оголошення класу; вбудовані функції класу. Ідентичність об'єктів, копіювання, ініціалізація і присвоєння, ліквідація об'єктів.
4. Поточний об'єкт `this`.
5. Створення і ініціалізація об'єктів, властивості конструкторів, довизначення конструкторів, конструктор за замовчуванням,

обмеження прав доступу до конструктора. Ліквідація об'єктів, властивості деструкторів.

6. Перевантаження (overloading) операторів (операцій-функцій): арифметичні операції (бінарні, унарні, суміщені з присвоєнням), оператори управління пам'яттю, оператор виклику функції, оператор доступу до елемента масиву, оператор присвоєння, конвертори (перетворення типів).

Теми рефератів

1. Типи конструкторів, конструктори за замовчуванням, конструктори копіювання, конструктори присвоєння.
Література [2–4]
2. Конструктори та деструктори. Послідовність виконання конструкторів та деструкторів.
Література [2–4]
3. Перевантаження бінарних операторів, оформлених як функція класу, та як дружні функції.
Література [2–4]
4. Перевантаження операторів виклику функцій, доступу до елемента масиву та перетворення типів.
Література [2–4]

Питання для самоконтролю та співбесіди

1. Що таке структура в C++? Чи може структура містити функції?
2. Який специфікатор доступу до полів структури приймається за замовчуванням?
3. Як співвідносяться структури і класи в C++? Як ініціалізуються структури? Як здійснюється доступ до полів структури?
4. Охарактеризуйте структуру класу в мові C++? Як співвідносяться класи і об'єкти? Як і де оголошується клас? Як визначаються об'єкти класу?
5. Як здійснюється доступ до полів класу? Методів класу?
6. Чи можуть об'єкти бути аргументами функцій? Якщо можуть, то як передати об'єкти у функцію?
7. Як розміщуються об'єкти у пам'яті? Чи має кожен об'єкт свої дані? Чи має кожен об'єкт свої методи?
8. Що означають статичні дані класу? Що означають статичні методи класу?

9. Що означають вбудовані методи класу? Коли вони застосовуються? Чи є вони обов'язковими?
10. Як і коли ініціалізуються об'єкти? Що таке конструктор класу? Скільки конструкторів може мати клас?
11. Що означають константні методи? Коли доцільні такі методи в класі?
12. Що означають константні об'єкти? Чим характерні методи таких об'єктів?
13. Чи може бути масив членом класу? Якого типу елементи може мати такий масив?
14. Як оголосити масив структур? Як здійснюється доступ до полів структури в масиві структур?
15. Чи можуть бути об'єкти елементами масиву? Як одержати конкретний об'єкт з такого масиву?

Тестові завдання

1. Визначіть, для чого призначені конструктори класу:

- а) для конструювання класу;
- б) для ініціалізації об'єктів класу;
- в) для виділення пам'яті об'єктам класу;
- г) для об'яви об'єктів класу.

2. Зазначте, з яких елементів складаються об'єкти класу:

- а) з членів класу;
- б) з функцій класу;
- в) тільки з конструкторів та деструкторів;
- г) з членів і функцій класу.

3. Зазначте, яке службове слово позначає інтерфейсну частину класу:

- а) public;
- б) private;
- в) interface;
- г) implementation.

4. Зазначте, що позначає службове слово this:

- а) поточний об'єкт класу;
- б) наступний об'єкт поточного класу;
- в) покажчик на поточний об'єкт класу;
- г) покажчик на попередній об'єкт поточного класу.

5. Зазначте конструктор, який буде викликаний компілятором, якщо об'єкт А класу Point об'явити таким чином – Point A:

- а) конструктор за замовчуванням;
- б) конструктор копіювання;
- в) конструктор присвоювання;
- г) конструктор переміщення.

6. Зазначте, коли викликається деструктор деякого об'єкта:

- а) після виклику конструктора цього об'єкта;
- б) після завершення програми;
- в) коли програма виходить з області дії об'єкта;
- г) після виклику деструктора цього об'єкта.

7. Зазначте, який тип повертають конструктори об'єктів:

- а) void;
- б) int;
- в) char;
- г) double.

8. Зазначте, що означає перевантаження операторів для об'єктів певного класу:

- а) що перевантажний оператор може бути застосований для об'єктів будь-яких класів;
- б) що перевантажний оператор може бути застосований для функцій цього класу;
- в) що перевантажний оператор може бути застосований для об'єктів цього класу;
- г) що перевантажний оператор може бути застосований для членів цього класу.

9. Зазначте, що таке дружба функція певного класу:

- а) функція, що має однакові параметрами з функціями цього класу;
- б) функція, що може звертатися до членів цього класу;
- в) функція, що використовує конструктор цього класу;
- г) функція, що використовує деструктор цього класу.

10. Визначіть, чим повинні відрізнятися різні конструктори одного класу:

- а) назвами;
- б) сигнатурою;

- в) типом, що повертаються конструкторами;
- г) кількістю параметрів.

Тема 3. Об'єктно-орієнтовне програмування

1. Поняття базового і похідного класу.
2. Доступ до базового класу. Специфікація доступу protected. Незмінність базового класу.
3. Конструктор похідного класу.
4. Перевантаження функцій.
5. Ієрархія класів. Абстрактний базовий клас. Загальне і часткове успадкування.

Теми рефератів

1. Типи доступу до елементів базового класу.
Література [2; 3; 5]
2. Послідовність виклику конструкторів та деструкторів базового та похідного класу.
Література [2; 3; 5]
3. Перевантаження і шаблони функцій.
Література [2; 3; 5]
4. Ієрархія класів та абстрактний базовий клас.
Література [2; 3; 5]

Питання для самоконтролю та співбесіди

1. Дайте визначення базового і похідного класу в C++.
2. Скільки може бути базових класів в одного похідного класу?
3. Скільки похідних класів може наслідувати один базовий клас?
4. Як здійснюється доступ до членів базового класу?
5. Які специфікації доступу до базового класу можливі?
6. Що означає специфікація доступу protected?
7. Які встановлюються специфікатори доступу до елементів базового класу, якщо специфікатор доступу до базового класу відсутній?
8. Яку роль виконує конструктор похідного класу?
9. В якій послідовності викликаються конструктори базового і похідних класів?
10. В якій послідовності викликаються деструктори базового і похідних класів?
11. Чи можуть похідні класи мати елементи і функції класу з тими самими назвами, що й у базовому класі?
12. Що означає абстрактний базовий клас?

Тестові завдання

1. Визначіть, що таке базовий клас:

- а) клас, побудований на основі іншого класу;
- б) клас, на основі якого побудовано інший клас;
- в) клас, на основі якого не побудовано жодного класу;
- г) клас, у якого немає елементів та методів.

2. Визначіть, що таке похідний клас:

- а) клас, побудований на основі іншого класу;
- б) клас, на основі якого побудовано інший клас;
- в) клас, на основі якого не побудовано жодного класу;
- г) клас, у якого крім елементів є методи.

3. Зазначте, які не можуть бути специфікатори доступу при успадкуванні класів:

- а) public;
- б) protected;
- в) private;
- г) open.

4. Визначіть, чи має доступ похідний клас до закритих елементів свого базового класу:

- а) має;
- б) немає;
- в) має при відкритому успадкуванні;
- г) має за відсутності специфікатора успадкування.

5. Визначіть, який доступ має в похідному класі метод базового класу із специфікатором *public*, якщо модифікатор успадкування відсутній:

- а) public;
- б) protected;
- в) private;
- г) недосяжний.

6. Визначіть, який має доступ в похідному класі метод базового класу із специфікатором *public*, якщо модифікатор успадкування *public*:

- а) public;
- б) protected;
- в) private;
- г) недосяжний.

7. Визначіть, який має доступ в похідному класі метод базового класу із специфікатором *public*, якщо модифікатор успадкування *protected*:

- а) *public*;
- б) *protected*;
- в) *private*;
- г) недосяжний.

8. Визначіть, який має доступ в похідному класі метод базового класу із специфікатором *public*, якщо модифікатор успадкування *private*:

- а) *public*;
- б) *protected*;
- в) *private*;
- г) недосяжний.

9. Визначіть, чи можна об'єкт базового класу *a* присвоїти об'єкту похідного класу *b* (тобто $b=a$):

- а) можна.
- б) не можна.
- в) можна лише в деяких випадках.
- г) можна через приведення типів.

10. Якщо похідний клас містить метод з тим самим ім'ям, що й у базовому, то це призведе:

- а) до помилки компіляції;
- б) до заміщення методу базового класу;
- в) до тієї ж реалізації методу базового класу;
- г) до ігнорування методу похідного класу.

Тема 4. Множинне успадкування

1. Множинне успадкування. Методи класів і множинне успадкування. Конструктори при множинному успадкуванні. Конструктор без аргументів. Конструктор з багатьма аргументами.
2. Успадкування і деструктори. Виклик конструкторів і деструкторів при успадкуванні. Успадкування і захищені члени класу.
3. Створення похідних об'єктів, конструктор похідного класу, послідовність виклику конструкторів, ліквідація дочірніх об'єктів,

деструктор дочірнього класу, послідовність виклику деструкторів.

4. Статичне і динамічне зв'язування: поліморфізм, віртуальні функції, заміщення функцій, віртуальні деструктори.

Теми рефератів

1. Конструктори при множинному успаковуванні.
Література [3; 6]
2. Неоднозначність при множинному успаковуванні.
Література [3; 6]
3. Поліморфізм при множинному успаковуванні.
Література [3; 6]
4. Статичне та динамічне зв'язування при множинному успаковуванні.
Література [3; 6]

Питання для самоконтролю та співбесіди

1. Ієрархія об'єктів і ієрархія класів.
2. Що називають базовим класом?
3. Що називають похідним класом?
4. Що називають успаковуванням?
5. Захищена область класу.
6. Область видимості класу.
7. Особливості створення та ліквідації похідних об'єктів.
8. Послідовність виклику конструкторів.
9. Деструктор похідного класу, послідовність виклику деструкторів.
10. Охарактеризуйте статичне і динамічне зв'язування.
11. Що таке поліморфізм?
12. Що таке статичне та динамічне зв'язування.

Тестові завдання

1. Визначіть, що таке множинне успаковування:

- а) успаковування декількох похідних класів від одного базового;
- б) успаковування похідного класу від декількох базових;
- в) успаковування похідного класу із специфікатором `mapu`;
- г) успаковування похідного класу від класу, що є похідним від іншого класу.

2. Визначіть, що таке чисто віртуальний метод:

- а) метод, що визначений у класі, але не має конкретної реалізації;

- б) метод, що визначений у похідному класі;
- в) метод, що очищає змінні об'єктів класу;
- г) метод, що не має параметрів.

3. Визначіть, що таке абстрактний клас:

- а) клас, що не має нащадків;
- б) клас, перед визначенням якого стоїть abstract;
- в) клас, на базі якого побудовані всі інші класи;
- г) клас, що містить хоча б один чисто віртуальний метод.

4. Якщо функція задається як чисто віртуальна, то це передбачає:

- а) що вона не визначається в жодному похідному класі;
- б) що вона не повинна підмінятися в кожному похідному класі;
- в) що вона повинна підмінятися в кожному похідному класі;
- г) що вона не повинна використовуватися в жодному похідному класі.

5. Зазначити специфікатор, який слід додати перед модифікатором доступу до базового класу за множинного успадкування, щоб базовий клас був включений до похідного класу лише один раз:

- а) onetime;
- б) virtual;
- в) const;
- г) abstract.

6. Визначити, чи успадковуються конструктори в похідних класах:

- а) взагалі не викликаються;
- б) успадковуються;
- в) не успадковуються;
- г) ігноруються.

7. Визначити, коли викликається конструктор похідного класу:

- а) взагалі не викликається;
- б) перед конструктором базового класу;
- в) після конструктора базового класу;
- г) викликається замість конструктора базового класу.

8. Зазначте, в якій послідовності викликаються деструктори:

- а) у послідовності, оберненій до послідовності виклику конструкторів;
- б) у такій же послідовності, як і конструктори;
- в) взагалі не викликаються;
- г) викликаються лише деструктори базових класів.

9. Зазначте, в якій послідовності викликаються конструктори базових класів при множинному успадковуванні класу (наприклад, в такому визначенні: `class C: public A, public B`):

- а) взагалі не викликаються;
- б) спочатку А, потім В, потім С;
- в) спочатку С, потім А, потім В;
- г) спочатку В, потім А, потім С.

10. Визначіть, скільки копій класу А буде містити клас D, якщо класи А, В, С, D визначені таким чином: `class B: virtual public A`; `class C: virtual public A`; `class D: public B, public C` }

- а) 0;
- б) 1;
- в) 2;
- г) 3.

СПИСОК ЛІТЕРАТУРИ

Основна

1. Лафоре Р. Объектно-ориентированное программирование в С++. — СПб.: Питер, 2004. — 923 с.
2. Павловская Т. А., Щупак Ю. А. С++. Объектно-ориентированное программирование: Практикум. — СПб.: Питер, 2006. — 265 с.
3. Павловская Т. А. С/С++ Программирование на языке высокого уровня. — СПб.: Питер, 2001.
4. Шилдт Г. Теория и практика С++. — СПб.: ВНУ, 1996.
5. Джон Макгрегор, Давид Сайкс. Тестирование объектно-ориентированного программного обеспечения: Практик. пособие. ТИД "ДС", 2002. — 432с
6. Брукс Ф. Как проектируются и создаются программные комплексы. — М.: Наука, 1979.

Додаткова

1. *Альфред В. Ахо, Джон Э. Хопкрофт, Джеффри Д. Ульман.* Структуры данных и алгоритмы. — 2000, Вильямс. — 384 с.
2. *Ахо А., Хопкрофт Дж., Ульман Дж.* Построение и анализ вычислительных алгоритмов. — 1979. — М.: Мир. — 540 с.
3. *Вирт Н.* Алгоритмы + структуры данных = программы. — М.: Мир, 1985. — 406 с.
4. *Гудман С., Хидетниеми С.* Введение в разработку и анализ алгоритмов. — М.: Мир, 1981. — 366 с.
5. *Зелковиц М., Шоу А., Геннон Дж.* Принципы разработки программного обеспечения. — М.: Мир, 1982. — 368 с.
6. *Иан Соммервилл.* Инженерия программного обеспечения. — М.: Вильямс. — 624 с.
7. *Кнут Д.* Искусство программирования для ЭВМ. Основные алгоритмы. — М.: Мир, 1976. — 735 с.
8. *Сэм Канер, Джек Фолк, Енг Кек Нгуен.* Тестирование программного обеспечения. — ДиаСофт, 2000. — 544 с.
9. *Уокер Ройс.* Управление проектами по созданию программного обеспечения. — Лори, 2002. — 448 с.
10. *Фокс Дж.* Программное обеспечение и его разработка. — М.: КОМПЬЮТЕР ПРЕСС, 1995. — 280 с.
11. *Янг М.* Visual C++ 6. — К.: BHV, 1999.

ЗМІСТ

Пояснювальна записка	3
Теми самостійної роботи.....	10
Список літератури.....	22

Відповідальний за випуск *А. Д. Везеренко*
Редактор *Т. М. Федосенко*
Комп'ютерне верстання *К. П. Махія*

Зам. № ВКЦ-3911

Підп. до друку 16.04.09. Формат 60×84/₁₆. Папір офсетний.
Друк ротативний трафаретний.

Ум. друк. арк. 1,39. Обл.-вид. арк. 1,19. Наклад 30 пр.

Міжрегіональна Академія управління персоналом (МАУП)
03039 Київ-39, вул. Фрометівська, 2, МАУП

ДП «Видавничий дім «Персонал»
03039 Київ-39, просп. Червонозоряний, 119, літ. XX

*Свідоцтво про внесення до Державного реєстру
суб'єктів видавничої справи ДК № 3262 від 26.08.2008 р.*