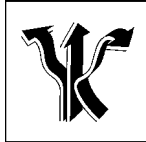


МІЖРЕГІОНАЛЬНА
АКАДЕМІЯ УПРАВЛІННЯ ПЕРСОНАЛОМ



МАУП

НАВЧАЛЬНА ПРОГРАМА
дисципліни
“СУЧАСНІ КОМП’ЮТЕРНІ ТЕХНОЛОГІЇ
ОБ’ЄКТНО-ОРІЄНТОВАНОГО
ПРОГРАМУВАННЯ НА C++”
(для бакалаврів)

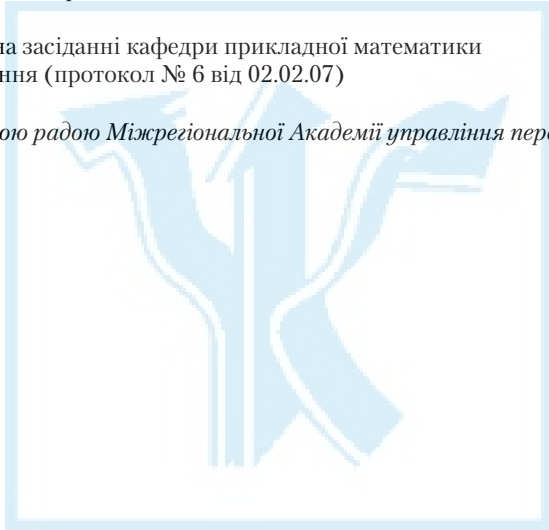
МАУП

Київ 2007

Підготовлено професором кафедри інформатики та інформаційних технологій *Г. А. Шпортюком*

Затверджено на засіданні кафедри прикладної математики та програмування (протокол № 6 від 02.02.07)

Схвалено Вченою радою Міжрегіональної Академії управління персоналом



Шпортюк Г. А. Навчальна програма дисципліни “Сучасні комп’ютерні технології об’єктно-орієнтованого програмування на С++” (для бакалаврів). – К.: МАУП, 2007. – 14 с.

Навчальна програма містить пояснювальну записку, тематичний план, зміст дисципліни “Сучасні комп’ютерні технології об’єктно-орієнтованого програмування на С++”, теми лабораторних занять, питання для самоконтролю, а також список літератури.

© Міжрегіональна Академія управління персоналом (МАУП), 2007

ПОЯСНЮВАЛЬНА ЗАПИСКА

Програма курсу “Сучасні комп’ютерні технології об’єктно-орієнтованого програмування на С++” містить теорію та лабораторні заняття, орієнтовані на опанування студентами основних концепцій об’єктно-орієнтованого програмування – інкапсуляції, успадкування і поліморфізму, а також знаннями, уміннями і навичками алгоритмізації і створення комп’ютерних програм для розв’язання задач прикладної математики та інформатики з використанням сучасної технології об’єктно-орієнтованого програмування на С++, що ґрунтується на програмних поняттях класу та об’єкта, які адекватно відображають реальні фізичні об’єкти, поєднуючи дані і дії над цими даними в єдине ціле, і дають змогу створювати ефективні програмні продукти.

Орієнтовна кількість годин на дисципліну – 162 (з них 24 – лекційні заняття, 24 – лабораторні, 114 – самостійна робота студентів).

Курс необхідний для створення на основі сучасної технології об’єктно-орієнтованого програмування програмних продуктів для прикладних задач та для вивчення подальших курсів у підготовці бакалаврів прикладної математики в управлінні. Це:

- технологія програмування та створення програмних продуктів;
- системне програмування;
- бази даних та інформаційні системи;
- комп’ютерна вірусологія;
- системне програмування і операційні системи;
- статистичне моделювання;
- практичні аспекти побудови баз даних.

ТЕМАТИЧНИЙ ПЛАН
дисципліни
“СУЧАСНІ КОМП’ЮТЕРНІ ТЕХНОЛОГІЇ
ОБ’ЄКТНО-ОРІЄНТОВАНОГО ПРОГРАМУВАННЯ НА C++”

| № пор. | Назва змістового модуля і теми |
|------------------|--|
| | Змістовий модуль I. Засоби процедурного програмування і класи C++ |
| 1 | Основи програмування на C++ |
| 2 | Класи, об’єкти, абстрактні типи даних |
| 3 | Масиви, рядки символів, покажчики і посилання. Оператори динамічного розподілу пам’яті |
| | Змістовий модуль II. Об’єктно-орієнтовані технології програмування |
| 4 | Базові і похідні класи. Успадкування |
| 5 | Перевантаження функцій. Аргументи по замовчуванню. Конструктори копіювання |
| 6 | Перевантаження операторів |
| 7 | Віртуальні функції і поліморфізм |
| 8 | Шаблони класів і виняткові ситуації |
| Разом годин: 162 | |

ЗМІСТ
дисципліни
“СУЧАСНІ КОМП’ЮТЕРНІ ТЕХНОЛОГІЇ
ОБ’ЄКТНО-ОРІЄНТОВАНОГО ПРОГРАМУВАННЯ НА C++”

Змістовий модуль I. Засоби процедурного програмування і класи C++

Тема 1. Основи програмування на C++

Структура програми. Функції. Оператори. Директиви компілятора. Заголовкові файли. Типи даних. Змінні. Блоки і області видимості змінних. Потоки введення-виведення. Маніпулятори. Операції відношення. Арифметичні і логічні операції. Цикли і розгалуження. Вибір типу циклу. Умовні операції. Типи умовних операторів. Вкла-

дені розгалуження. Реалізація багатоваріантних розгалужень. Булеві змінні і операції над ними. Цілі величини в ролі булевих. Пріоритети операцій C++. Структури. Синтаксис визначення структури. Визначення структурних змінних. Доступ до полів структури. Приклади застосування структур. Вкладені структури. Доступ до полів вкладених структур. Ініціалізація структур. Структури і класи C++.

Література [1–4; 8; 9]

Тема 2. Класи, об'єкти, абстрактні типи даних

Визначення класу. Оголошення об'єктів класу. Виклик методів класу. Об'єкти програми і реальні фізичні об'єкти. Клас як тип даних. Абстрактні типи даних. Об'єкти як аргументи функцій. Структури і класи. Статичні дані класу. Вбудовані методи класу. Визначення методів поза класом. Константні методи. Константні об'єкти. Конструктори класу.

Література [1–3; 6–10]

Тема 3. Масиви, рядки символів, покажчики і посилання. Оператори динамічного розподілу пам'яті

Масиви одномірні, двомірні, багатомірні. Операції над масивами. Масиви структур. Масиви як члени класів. Масиви об'єктів. Доступ до об'єктів у масивах. Рядки і рядкові змінні. Копіювання рядків. Масиви рядків. Рядки як члени класів. Стандартний клас string мови C++. Визначення об'єктів класу string. Пошук об'єктів класу string. Модифікація об'єктів класу string. Доступ до символів в об'єктах класу string. Операція одержання адреси об'єктів. Змінні покажчики. Доступ до об'єктів через покажчики. Покажчики і масиви. Покажчики-константи і покажчики-змінні. Покажчики і функції. Покажчики і рядки. Масиви покажчиків на рядки. Покажчики на об'єкти. Масиви покажчиків на об'єкти. Покажчики на покажчики. Посилання. Передавання параметрів за допомогою посилань. Передавання посилання на об'єкти. Повернення посилань. Посилання на довільні типи даних. Операції розподілу пам'яті new і delete. Ініціалізація виділеної пам'яті. Виділення пам'яті під масиви. Виділення пам'яті під об'єкти. Динамічні структури даних.

Література [1–4; 6–8]

Змістовий модуль II. Об'єктно-орієнтовані технології програмування

Тема 4. Базові і похідні класи. Успадкування

Поняття базового і похідного класу. Доступ до базового класу. Специфікація доступу protected. Незмінність базового класу. Конструктор похідного класу. Перевантаження функцій. Ієрархія класів. Абстрактний базовий клас. Загальне і часткове успадкування. Множинне успадкування. Методи класів і множинне успадкування. Конструктори при множинному успадкуванні. Конструктор без аргументів. Конструктор з багатьма аргументами. Успадкування і деструктори. Виклик конструкторів і деструкторів при успадкуванні. Успадкування і захищені члени класу.

Література [1–4; 8–10]

Тема 5. Перевантаження функцій. Аргументи по замовчуванню. Конструктори копіювання

Перевантаження функцій. Перевантаження конструкторів. Створення ініціалізованих і неініціалізованих об'єктів. Конструктор копіювання. Копіювання об'єктів класу. Визначення адреси перевантаженої функції. Аргументи функцій по замовчуванню. Аргументи по замовчуванню і перевантаження. Застосування аргументів по замовчуванню.

Література [1–4; 8–10]

Тема 6. Перевантаження операторів

Перевантаження унарних операцій. Аргументи операції. Створення префіксної і постфіксної форм операторів інкрементації і декрементації. Постфіксні операції. Перевантаження бінарних операцій. Арифметичні операції. Операції порівняння. Операції арифметичного присвоювання. Операція індексації масиву. Перетворення основних типів в основні типи. Перетворення об'єктів в основні типи і навпаки. Перетворення об'єктів класів в об'єкти інших класів. Перевантаження операторів new і delete. Перевантаження операторів new і delete для масивів.

Література [1; 2; 4; 7; 8]

Тема 7. Віртуальні функції і поліморфізм

Поняття віртуальної функції. Доступ до звичайних методів через покажчики. Доступ до віртуальних методів через покажчики. Пізні зв'язування. Абстрактні класи і чисті віртуальні функції. Віртуальні деструктори. Віртуальні базові класи. Віртуальні деструктори. Віртуальні функції і специфікатори доступу. Дружні функції. Дружні класи. Статичні функції. Доступ до статичних функцій. Покажчик `this`. Доступ до компонентних даних через покажчик `this`. Динамічна операція `pro` типу.

Література [1–3; 5–10]

Тема 8. Шаблони класів і виняткові ситуації

Шаблони функцій. Шаблон простої функції. Шаблони функцій з кількома аргументами. Перевантаження шаблонної функції. Шаблони класів. Застосування стандартних типів у шаблонних класах. Застосування аргументів по замовчуванню в шаблонних класах. Виняткові ситуації. Поняття виняткової ситуації. Використання виняткових ситуацій. Збудження виняткових ситуацій. Збудження кількох виняткових ситуацій. Блоки `try`. Застосування блоку `try`. Програмування з винятковими ситуаціями. Оголошення виняткових ситуацій. Обробка виняткових ситуацій. Необроблені виняткові ситуації. Виняткові ситуації і локальні об'єкти. Виняткові ситуації і конструктори. Виняткові ситуації і динамічний розподіл пам'яті.

Література [1–4; 8–10]

ТЕМИ ЛАБОРАТОРНИХ ЗАНЯТЬ

1. Структура програми на C++. Основи потоків введення-виведення. Введення і виведення різних типів даних. Маніпулятори ширини поля виведення, точності, системи числення. Заголовкові файли потоків і маніпуляторів.
2. Типи даних і операції в мові C++. Арифметичні, логічні, булеві, операції відношення. Умовні оператори. Розгалуження в програмі. Багатоваріантні розгалуження. Цикли.
3. Структури і операції зі структурами. Приклади застосування структур. Структури і класи в C++. Відмінності між структурами C і C++.

4. Синтаксис класу. Клас як тип даних. Поля і методи класу. Екземпляри класу. Оголошення і ініціалізація об'єктів. Доступ до полів і методів об'єктів.
5. Специфікатори доступу private і public. Методи як засіб взаємодії програми з полями об'єктів. Об'єкти як параметри функцій.
6. Масиви як члени класів. Масиви структур і об'єктів. Доступ до об'єктів у масивах. Масиви рядків. Операції з рядками. Клас string. Операції з об'єктами класу string.
7. Показчики і об'єкти. Масиви показчиків на об'єкти. Операції з масивами показчиків на об'єкти. Показчики на показчики.
8. Операції динамічного розподілу пам'яті. Ініціалізація динамічної пам'яті. Розподіл пам'яті для об'єктів, масивів показчиків на об'єкти.
9. Створення похідних і базових класів. Доступ до полів. Специфікація доступу protected. Конструктори похідного і базового класів.
10. Множинне успадкування. Конструктори при множинному успадкуванні. Ієрархія класів. Перевантаження функцій.
11. Аргументи функцій по замовчуванню. Вбудовані функції. Конструктор копіювання. Копіювання об'єктів класу.
12. Перевантаження унарних операцій. Перевантаження префіксних і постфіксних операцій.
13. Перевантаження бінарних операцій. Арифметичні операції. Операції арифметичного присвоєння. Операція індексації масиву. Перевантаження операцій динамічного розподілу пам'яті.
14. Віртуальні функції. Оголошення віртуальних функцій. Доступ до віртуальних функцій через показчики. Застосування віртуальних функцій.
15. Чисті віртуальні функції. Віртуальні деструктори. Абстрактні класи. Використання абстрактних класів.
16. Дружні функції. Дружні класи. Дружні функції як члени класів і незалежні дружні функції. Доступ до закритих членів класу.
17. Робота з шаблонами функцій. Шаблони простих функцій. Шаблони функцій з багатьма аргументами. Перевантаження шаблонних функцій.
18. Шаблони класів. Застосування стандартних типів у шаблонних класах. Застосування аргументів по замовчуванню в шаблонних класах.

19. Виняткові ситуації. Оголошення виняткових ситуацій. Обробка виняткових ситуацій. Збудження виняткових ситуацій. Збудження кількох виняткових ситуацій.
20. Програмування з використанням виняткових ситуацій. Блоки try. Необроблені виняткові ситуації. Виняткові ситуації і динамічний розподіл пам'яті.

ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ

1. Чи обов'язково вказувати типи параметрів функції у прототипі? А імена параметрів?
2. Що таке директиви компілятора? Яка їх роль у програмі?
3. Для чого в програму включаються заголовкові файли? Чи завжди вони потрібні?
4. Який заголовковий файл включається в програму при використанні потоків введення-виведення C++?
5. Які ви знаєте маніпулятори? Що вони означають? Який заголовковий файл потрібний для їх використання?
6. Які ви знаєте операції відношення? Де вони застосовуються?
7. Які типи даних в мові C++ ви знаєте? Які операції можна виконувати над ними?
8. Що таке булеві змінні? Які значення вони можуть приймати?
9. Які ви знаєте умовні оператори? Де вони застосовуються?
10. Охарактеризуйте типи циклічних операторів? В яких випадках застосовується той чи інший тип оператора?
11. Що таке структура в C++? Чи може структура містити функції?
12. Який спеціфікатор доступу до полів структури приймається по замовчуванню?
13. Як співвідносяться структури і класи в C++? Як ініціалізуються структури? Як здійснюється доступ до полів структури?
14. Охарактеризуйте структуру класу в мові C++. Як співвідносяться класи і об'єкти? Як і де оголошується клас? Як визначаються об'єкти класу?
15. Як здійснюється доступ до полів класу? Методів класу?
16. Чи можуть об'єкти бути аргументами функцій? Якщо можуть, то як передати об'єкти у функцію?
17. Як розміщуються об'єкти у пам'яті? Чи має кожен об'єкт свої дані? А методи?

18. Що означають статичні дані класу? Що означають статичні методи класу?
19. Що означають вбудовані методи класу? Коли вони застосовуються? Чи є вони обов'язковими?
20. Як і коли ініціалізуються об'єкти? Що таке конструктор класу? Скільки конструкторів може мати клас?
21. Що означають константні методи? Коли доцільні такі методи в класі?
22. Що означають константні об'єкти? Чим характерні методи таких об'єктів?
23. Якого типу елементи можуть мати масиви?
24. Як здійснюється доступ до елементів масиву? Чи залежить він від типу елементів масиву?
25. Чи може бути масив членом класу? Якого типу елементи може мати такий масив?
26. Як оголосити масив структур? Як здійснюється доступ до полів структури в масиві структур?
27. Чи можуть бути об'єкти елементами масиву? Як одержати конкретний об'єкт з такого масиву?
28. Що таке рядок в C++?
29. Як оголосити у програмі масив рядків? Навести приклад застосування такого типу даних.
30. Чи є в C++ тип даних рядок? Для чого в C++ введено клас string?
31. Наведіть приклад застосування класу string для операцій з рядками.
32. Чи є покажчик змінною? Якого типу?
33. Що означає покажчик на невизначений тип? Як використовуються такі покажчики?
34. Як одержати дані, на які вказує покажчик?
35. Що означає покажчик на об'єкт?
36. Як викликати методи об'єкта, користуючись покажчиком на об'єкт?
37. Як оголосити масив покажчиків на об'єкти? Що означає посилання на об'єкт?
38. Дайте порівняльну характеристику покажчиків і посилань. Посилання на об'єкти та їх застосування.
39. Як відбувається динамічний розподіл пам'яті в C++? Як ініціалізувати виділену пам'ять заданим значенням? Як виділити пам'ять

- для одно- і двовимірних масивів? Як передати одно- і двовимірний масиви у функцію?
40. Дати означення базового і похідного класу в C++. Скільки може бути базових класів? А похідних? Як здійснюється доступ до членів базового класу?
 41. Що означає специфікація доступу protected?
 42. Яку роль виконує конструктор похідного класу?
 43. Що означає абстрактний базовий клас?
 44. Що означає загальне успадкування? Часткове? Що таке множинне успадкування?
 45. Як виглядають конструктори при множинному успадкуванні?
 46. Як виконуються деструктори при успадкуванні класів?
 47. Як змінюється специфікація доступу до членів класу при успадкуванні?
 48. Яку роль виконують конструктори в успадкованих класах?
 49. Що таке перевантаження функцій в C++?
 50. Як визначити адресу перевантаженої функції?
 51. Як здійснюється ініціалізація об'єктів при їх створенні?
 52. Що таке конструктор копіювання? Як здійснюється копіювання об'єктів класу?
 53. Аргументи функцій по замовчуванню.
 54. Який порядок розміщення аргументів по замовчуванню?
 55. В яких випадках застосовуються аргументи по замовчуванню?
 56. Що таке перевантаження операцій?
 57. З якою метою здійснюють перевантаження операцій? Чи всі операції можуть бути перевантажені?
 58. Скільки аргументів потрібно для визначення перевантаженої унарної операції?
 59. Чим відрізняється дія перевантаженої операції ++ при її використанні у префіксній формі від використання в постфіксній формі?
 60. Якщо перевантажується операція арифметичного присвоєння, то куди передається об'єкт?
 61. Що потрібно використати для перетворення стандартного типу до типу, визначеному користувачем?
 62. Що означає перевантаження операції індексації масиву для об'єктів?
 63. Перевантаження бінарних операцій.

64. Покажіть на прикладі дію однієї з перевантажених операцій арифметичного присвоєння.
65. Задля чого здійснюють перевантаження операцій динамічного розподілу пам'яті?
66. Що таке віртуальна функція? Чи може один і той самий виклик віртуальної функції виконувати методи об'єктів, які належать різним класам?
67. Якими повинні бути класи, щоб один і той самий виклик віртуальної функції міг викликати методи цих класів?
68. Чи може покажчик на базовий клас посилатися на об'єкти похідних класів?
69. Що означає термін "динамічне зв'язування"?
70. Що таке чиста віртуальна функція? Де вона визначається? Який клас називається абстрактним?
71. Якщо дружня функція не є методом класу, то чи має вона доступ до прихованих даних класу?
72. З якою метою створюють дружні функції?
73. Чи може конструктор копіювання бути перевизначений для копіювання тільки частини даних об'єкта?
74. Які класи називаються дружніми?
75. На що вказує покажчик `this`?
76. Чи може абстрактний клас мати абстрактні об'єкти?
77. Чи можуть деструктори бути віртуальними? Якщо так, то в яких випадках вони застосовуються?
78. Що означає шаблон функції? З якою метою створюють шаблони функцій?
79. Опишіть синтаксис шаблону функції.
80. Коли відбувається генерація компілятором коду функції представленої шаблоном?
81. Особливості шаблону функції з кількома аргументами.
82. Коли доцільно створювати шаблони класів?
83. Виняткові ситуації. Об'єкти виняткових ситуацій.
84. Що таке блок `try`? Як він застосовується?
85. Як задати типи збуджуваних виняткових ситуацій?
86. Як обробляються виняткові ситуації? Що відбувається з необробленими винятковими ситуаціями?
87. Чи можуть збуджувати виняткову ситуацію конструктори?
88. В яких випадках переважно виникають виняткові ситуації?

89. Чи може генерувати виняткову ситуацію оператор new? В яких випадках?
90. Чи може програма продовжити своє виконання при виникненні виняткової ситуації?

СПИСОК ЛІТЕРАТУРИ

Основна

1. *Лафоре Р.* Объектно-ориентированное программирование в С++. — СПб.: Питер, 2005.
2. *Уолтер Савитч.* Язык С++. Курс объектно-ориентированного программирования. — М.: Вильямс, 2001.
3. *Литман С. Б., Лажойе Ж.* Язык программирования С++: Вводный курс. — М.: ДМК, 2001.

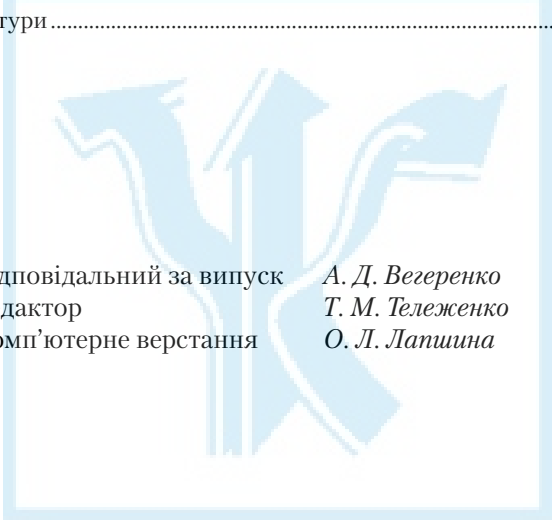
Додаткова

4. *Павловская Т. А.* С С++. Программирование на языке высокого уровня. — СПб.: Питер, 2001.
5. *Уокер Ройс.* Управление проектами по созданию программного обеспечения — М.: Лори, 2002.
6. *Буч Г.* Объектно-ориентированное проектирование с примерами применения. — К.: Диалектика, 1992.
7. *Шлеер С., Меллор С.* Объектно-ориентированный анализ: моделирование мира в состояниях. — К.: Диалектика, 1993.
8. *Шилдт Г.* Теория и практика С++. — СПб.: ВНУ, 1996.
9. *Шилдт Г.* Полный справочник по С++. К., 2003.
10. *Янг М.* Visual С++ 6. — К.: ВНУ, 2000.

МАУП

ЗМІСТ

| | |
|---|----|
| Пояснювальна записка..... | 3 |
| Тематичний план дисципліни “Сучасні комп’ютерні технології об’єктно-орієнтованого програмування на С++” | 4 |
| Зміст дисципліни “Сучасні комп’ютерні технології об’єктно-орієнтованого програмування на С++”..... | 4 |
| Теми лабораторних занять..... | 7 |
| Питання для самоконтролю..... | 9 |
| Список літератури | 13 |



Відповідальний за випуск *А. Д. Вегеренко*
Редактор *Т. М. Тележенко*
Комп’ютерне верстання *О. Л. Лапшина*

МАУП

Зам. № ВКЦ-3015

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