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

МАУП

НАВЧАЛЬНА ПРОГРАМА

дисципліни

“ Мова програмування C# та її реалізація на платформі Microsoft.NET”

(для магістрів)

Київ 2018

Підготовлено проф. кафедри кафедри комп'ютерних інформаційних систем та технологій М.П.Дяченком

Затверджено на засіданні кафедри комп'ютерних інформаційних систем та технологій (Протокол № 1 від 22.08.2018)

Схвалено Вченою радою Міжрегіональної Академії управління персоналом

Дяченко М.П., Навчальна програма дисципліни " Мова програмування C# та її реалізація на платформі Microsoft.NET." (для освітньо-кваліфікаційного рівня 'магістр'). — К.: МАУП, 2018. — 16 с.

Навчальна програма містить пояснювальну записку**,** тематичний план дисципліни, теми контрольних, питання для самоконтролю та список літератури.

ПОЯСНЮВАЛЬНА ЗАПИСКА

Програма курсу “ Мова програмування C# та її реалізація на платформі Microsoft.NET” 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 C#, 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.

Метою курсу є отримання знань про особливості платформи Microsoft.NET та мови програмування для цієї платформи, синтаксис мови C# та створення консольних і графічних додатків.

Завдання курсу полягає у ознайомленні з платформою Microsoft.NET та мовами програмування під цю платформу, з особливостями їх синтаксису на прикладі C# (типами даних, операціями, керуючими структурами, принципами роботи з масивами та рядками, описом класів, структур та об'єднань та ін.)

У результаті вивчення дисципліни:

- студент повинен знати:
  - принципи функціонування платформи Microsoft.NET
  - синтаксис мови C#
  - структуру консольних додатків та додатків з графічним інтерфейсом;
  - методи підвищення ефективності програм мовою C#
- студент повинен вміти:
  - розробляти консольні додатки та додатки з графічним інтерфейсом;
  - використовувати універсальні класи при розробці додатків;

- використовувати синхронний та асинхронний режими роботи з файлами.

Загальний обсяг дисципліни – 120 годин (4 кредити ЄКТС)

З них: 40 год лекції, 20 год практичні та семінарські заняття, 60 год самостійна робота.

Дана дисципліна входить до списку вибіркових навчальних дисциплін.

.

ТЕМАТИЧНИЙ ПЛАН

дисципліни

“ МОВА ПРОГРАМУВАННЯ C# ТА ЇЇ РЕАЛІЗАЦІЯ НА ПЛАТФОРМІ MICROSOFT.NET ”

№ 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 : 120

ЗМІСТ

дисципліни

“ МОВА ПРОГРАМУВАННЯ C# ТА ЇЇ РЕАЛІЗАЦІЯ НА ПЛАТФОРМІ MICROSOFT.NET

Змістовий модуль І. 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. Вкладені 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 нам ’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. Покажчи- ки-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. Пере-

давання 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 ІІ. 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 декре- ментації. 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. Пізнє

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 про типи.

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]

ТЕМАТИКА КОНТРОЛЬНИХ РОБІТ

1. Структура класу в мові C#.
2. Характеристика покажчиків і посилань. Посилання на об’єкти та їх застосування.
3. Аргументи функцій по замовчуванню.
4. Перевантаження бінарних операцій.
5. Особливості шаблону функції з кількома аргументами.
6. Виняткові ситуації. Об’єкти виняткових ситуацій.

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 специфікатор 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. 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. 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?
39. Дати означення базового і похідного класу в C#. Скільки може бути базових класів? А похідних? Як здійснюється доступ до членів базового класу?
40. Що означає специфікація доступу protected?
41. Яку роль виконує конструктор похідного класу?
42. Що означає абстрактний базовий клас?
43. Що означає загальне успадкування? Часткове? Що таке множинне успадкування?
44. Як виглядають конструктори при множинному успадкуванні?
45. Як виконуються деструктори при успадкуванні класів?
46. Як змінюється специфікація доступу до членів класу при успадкуванні?
47. Яку роль виконують конструктори в успадкованих класах?
48. Що таке перевантаження функцій в C#?
49. Як визначити адресу перевантаженої функції?
50. Як здійснюється ініціалізація об'єктів при їх створенні?

51. Що таке конструктор копіювання? Як здійснюється копіювання об’єктів класу?
52. Який порядок розміщення аргументів по замовчуванню?
53. В яких випадках застосовуються аргументи по замовчуванню?
54. Що таке перевантаження операцій?
55. З якою метою здійснюють перевантаження операцій? Чи всі операції можуть бути перевантажені?
56. Скільки аргументів потрібно для визначення перевантаженої унарної операції?
57. Чим відрізняється дія перевантаженої операції ++ при її використанні у префіксній формі від використання в постфіксній формі?
58. Якщо перевантажується операція арифметичного присвоєння, то куди передається об’єкт?
59. Що потрібно використати для перетворення стандартного типу до типу, визначеному користувачем?
60. Що означає перевантаження операції індексації масиву для об’єктів?
61. Задля чого здійснюють перевантаження операцій динамічного розподілу пам’яті?
62. Що таке віртуальна функція? Чи може один і той самий виклик віртуальної функції виконувати методи об’єктів, які належать різним класам?
63. Якими повинні бути класи, щоб один і той самий виклик віртуальної функції міг викликати методи цих класів?
64. Чи може покажчик на базовий клас посилатися на об’єкти похідних класів?
65. Що означає термін “динамічне зв’язування”?
66. Що таке чиста віртуальна функція? Де вона визначається? Який клас називається абстрактним?
67. Якщо дружня функція не є методом класу, то чи має вона доступ до прихованих даних класу?

68. З якою метою створюють дружні функції?
69. Чи може конструктор копіювання бути перевизначений для копіювання тільки частини даних об’єкта?
70. Які класи називаються дружніми?
71. На що вказує покажчик this?
72. Чи може абстрактний клас мати абстрактні об’єкти?
73. Чи можуть деструктори бути віртуальними? Якщо так, то в яких випадках вони застосовуються?
74. Що означає шаблон функції? З якою метою створюють шаблони функцій?
75. Опишіть синтаксис шаблону функції.
76. Коли відбувається генерація компілятором коду функції представленої шаблоном?
77. Коли доцільно створювати шаблони класів?
78. Що таке блок try? Як він застосовується?
79. Як задати типи збуджуваних виняткових ситуацій?
80. Як обробляються виняткові ситуації? Що відбувається з необробленими винятковими ситуаціями?
81. Чи можуть збуджувати виняткову ситуацію конструктори?
82. В яких випадках переважно виникають виняткові ситуації?
83. Чи може генерувати виняткову ситуацію оператор new? В яких випадках?
84. Чи може програма продовжити своє виконання при виникненні виняткової ситуації?

Відповідність підсумкової семестрової рейтингової оцінки в балах оцінці за національною шкалою та шкалою ECTS

Оцінка в балах	Оцінка за національною шкалою	Оцінка за шкалою ECTS	
		Оцінка	Пояснення
90-100	***Відмінно***	**A**	**Відмінно** (відмінне виконання лише з незначною кількістю помилок)
82 – 89	**Добре**	**B**	**Дуже добре** (вище середнього рівня з кількома помилками)
75 – 81		**C**	**Добре** (в загальному вірне виконання з певною кількістю суттєвих помилок)
67 – 74	**Задовільно**	**D**	**Задовільно** (непогано, але зі значною кількістю недоліків)
60 – 66		**E**	**Достатньо** (виконання задовольняє мінімальним критеріям)
35 – 59	**Незадовільно**	**FX**	**Незадовільно** (з можливістю повторного складання)
1 – 34		**F**	Незадовільно **(з обов'язковим повторним курсом)**

Підсумкова семестрова рейтингова оцінка в балах, за національною шкалою та шкалою ECTS заноситься до заліково-екзаменаційної відомості, навчальної картки та залікової книжки студента.

Підсумкова семестрова рейтингова оцінка заноситься до залікової книжки та навчальної картки студента, наприклад, так: ***92/Відм./A, 87/Добре/B, 79/Добре/C, 68/Задов./D, 65/Задов./E*** тощо..

Підсумкова рейтингова оцінка з дисципліни визначається як середньоарифметична оцінка з підсумкових семестрових рейтингових оцінок у балах (з цієї дисципліни – за перший та другий семестри) з наступним її переведенням в оцінки за національною шкалою та шкалою ECTS.

СПИСОК ЛІТЕРАТУРИ

Основна

1. *Лафоре Р* Объектно-ориентированное программирование в C#. - СПб.: Питер, 2007.
2. *Уолтер Савитч.* Язык C#. Курс объектно-ориентированного программирования. — М.: Вильямс, 2010.
3. *Липпман С. Б., Лажойе Ж.* Язык программирования C#: Вводный курс. — М.: ДМК, 2010.

Додаткова

4. *Павловская Т. А.* С C#. Программирование на языке высокого уровня. — СПб.: Питер, 2012.
5. *Уокер Ройс.* Управление проектами по созданию программного обеспечения — М.: Лори, 2012.
6. *Буч Г.* Объектно-ориентированное проектирование с примерами применения. — К.: Диалектика, 2012.
7. *Шлеер С., Меллор С.* Объектно-ориентированный анализ: моделирования мира в состояниях. — К.: Диалектика, 2003.
8. *Шилдт Г.* Теория и практика C#. — СПб.: BHV, 2006.
9. *Шилдт Г.* Полный справочник по C#. К., 2013.
10. *Янг М.* Visual C++. - К.: В НУ, 2010.

ЗМІСТ