

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
МИКОЛАЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ В. О. СУХОМЛИНСЬКОГО
Механіко-математичний факультет
Кафедра інформаційних технологій



ЗАТВЕРДЖУЮ

Проректор із науково-педагогічної роботи

О. А. Кузнецова

27 серпня 2020 р.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
ПРОГРАМУВАННЯ

Ступінь бакалавра

Галузь знань 12 Інформаційні технології

спеціальність 122 Комп'ютерні науки

Освітньо-професійна програма Комп'ютерні науки

2020 – 2021 навчальний рік

Розробник: Ходякова Галина Вікторівна, доцент кафедри інформаційних технологій, кандидат педагогічних наук _____ (Ходякова Г.В.)

Робоча програма затверджена на засіданні кафедри інформаційних технологій
Протокол № 1 від «26» серпня 2020 р.

Завідувач кафедри _____ (Зосімов В.В.)
«26» серпня 2020 р.

Програму погоджено з гарантом ОП Комп'ютерні науки
доцент кафедри, к.техн.н. _____ (Булгакова О.С.)

Анотація

Професійний рівень бакалавра з комп'ютерних наук значною мірою залежить від знань та умінь уміло використовувати програмні засоби для створення програмних додатків різного рівня складності. Мова С# - одна з найпотужніших мов, входить до TOP5 рейтингового списку найбільш популярних мов програмування. Функціонал С# дуже широкий: від розробки невеликих desktop-них програм, до величезних веб-сервісів, які обслуговують щодня мільйони користувачів. Фреймворк «.NET» представляє потужну платформу для створення додатків. Після вивчення даної дисципліни здобувачі освіти будуть здатними створювати прикладні програми за допомогою С#, набудуть практичні навички використання принципів об'єктно-орієнтованого програмування, створення графічних додатків в середовищі Windows.

Ключові слова: визначення класів в мові С#, програмування мовою С#, розробка програм, об'єктно-орієнтоване програмування.

Abstract

The professional level of a bachelor of computer science largely depends on the knowledge and skills to use software tools to create software applications of varying levels of complexity. C # is one of the most powerful languages, included in the TOP5 ranking list of the most popular programming languages. The functionality of C # is very wide: from the development of small desktop programs, to huge web services that serve millions of users every day. The .NET framework is a powerful platform for creating applications. After studying this discipline, students will be able to create applications using C #, gain practical skills in using the principles of object-oriented programming, creating graphics applications in Windows.

Keywords: definition of classes in C #, programming in C #, program development, object-oriented programming.

1. Опис навчальної дисципліни

Найменування показників	Галузь знань, освітній ступінь	Характеристика навчальної дисципліни
		<i>денна форма навчання</i>
Кількість кредитів – 5	Галузь знань: 12 Інформаційні технології	Нормативна
Індивідуальне наукове-дослідне завдання – розробка об’єктно-орієнтованої програми «Робота з матрицями»	Спеціальність: 122 Комп’ютерні науки	<i>Семестр</i>
Загальна кількість годин 150		7-й
		<i>Леції</i>
Тижневих годин для денної форми навчання: аудиторних – 4 самостійної роботи студента - 8		20 год.
		<i>Практичні, семінарські</i>
		-
		<i>Лабораторні</i>
		30 год.
		<i>Самостійна робота</i>
		100 год.
Електронна адреса дистанційного курсу на платформі Moodle: <i>http://moodle.mdu.edu.ua/course/view.php?id=1914</i>	Ступінь: бакалавра	Вид контролю: іспит

Мова навчання – українська.

Примітка.

Співвідношення кількості годин аудиторних занять до самостійної та індивідуальної роботи становить: для денної форми навчання – 50 год. – аудиторні заняття, 100 год. – самостійна робота (30% ~ 70%).

2. Мета, завдання навчальної дисципліни та результати навчання

Мета курсу: ознайомлення студентів з сучасною мовою програмування C# та оволодіння основними можливостями цієї мови, навичками хорошого стилю програмування, методами проектування та створення програм згідно сучасних технологій програмування.

Завдання курсу:

Студент повинен знати:

основні елементи мови C# – типи даних, оператори та керуючі інструкції мови; основні принципи ООП, правила визначення класів в мові C#.

Студент повинен вміти:

побудувати алгоритм розв'язку задачі відповідної складності, порівняти різні алгоритми і обрати оптимальний для конкретної задачі, створити консольний додаток, що реалізує обраний алгоритм; налагодити та протестувати програму; використовуючи панель інструментів середовища, створити простий Windows-додаток.

Передумови для вивчення дисципліни: «Алгоритми та структури даних», «Програмування».

Програмні результати навчання:

ПР4. Використовувати методи обчислювального інтелекту, машинного навчання, нейромережевої та нечіткої обробки даних, генетичного та еволюційного програмування для розв'язання задач розпізнавання, прогнозування, класифікації, ідентифікації об'єктів керування тощо.

ПР8. Використовувати методологію системного аналізу об'єктів, процесів і систем для задач аналізу, прогнозування, управління та проектування динамічних процесів в макроекономічних, технічних, технологічних і фінансових об'єктах.

ПР10. Використовувати інструментальні засоби розробки клієнт-серверних застосунків, проектувати концептуальні, логічні та фізичні моделі баз даних, розробляти та оптимізувати запити до них, створювати розподілені бази даних, сховища та вітрини даних, бази знань, у тому числі на хмарних сервісах, із застосуванням мов веб-програмування.

ПР12. Застосовувати методи та алгоритми обчислювального інтелекту та інтелектуального аналізу даних в задачах класифікації, прогнозування, кластерного аналізу, пошуку асоціативних правил з використанням програмних інструментів підтримки багатовимірного аналізу даних на основі технологій DataMining, TextMining, WebMining.

Згідно з вимогами освітньо-професійної програми студент оволодіває такими *компетентностями:*

I. Загальнопредметні:

ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК4. Здатність спілкуватися державною мовою як усно, так і письмово.

II. Фахові:

ФК3. Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв'язності та нерозв'язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем.

ФК8. Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.

Програма навчальної дисципліни

Кредит 1. Базові поняття програмування.

Тема 1. Основні поняття програмного забезпечення. Огляд процесу розробки програмного забезпечення. Етапи виконання комп'ютерної програми. Блок-схема та псевдокод як засоби, що використовуються у процесі розробки програми. Мови програмування. Процедурні та об'єктно-орієнтовані мови програмування. Місце мови C# серед сучасних мов програмування.

Тема 2. Представлення даних та двійкова арифметика. Необхідність типізації даних. Фундаментальні поняття типів даних та змінних. Змінні та константи. Правила іменування змінних. Операції. Використання програмного синтаксису.

Основні типи конструкцій в програмуванні. Використання розгалужень та циклів. Способи визначення логічних помилок. Культура програмування, поняття про ефективність алгоритму.

Тема 3. Процедури та функції як засіб створення модульних програм. Поняття аргументу, поняття параметра. Формальні і фактичні параметри, їх зв'язок за значенням та за адресою. Вхідні та вихідні параметри. Повернення скалярного значення функцією. Рекурсія.

Кредит 2. Загальний огляд платформи Microsoft .NET

Тема 4. Огляд платформи Microsoft .NET. Огляд циклу розробки програмного забезпечення. Переваги платформи .NET.

Тема 5. Компоненти платформи .NET. Формат виконуваних програмних модулів (EXE і DLL), загальномовне виконуюча среда, Common Language Runtime, CLR, набір програмних засобів.

Кредит 3. Основи мови програмування C#

Тема 6. Структура програми мовою C#. Основні елементи програми. Основні операції вводу/виводу. Засоби форматування. Компілювання, виконання та відлагодження.

Тема 7. Огляд загальної системи типів. Використання вбудованих типів-значень даних у мові C#. Створення типів даних користувача в мові C# (злічені типи та структури). Перетворення типів (явне та неявне). Використання пробілів та дужок у виразах для покращення читання програми.

Тема 8. Використання основних програмних конструкцій у мові C#. Розгалуження. Цикли. Поняття виключної ситуації. Обробка основних виключних ситуацій. Генерування виключних ситуацій.

Кредит 4. Поняття про класи та методи. Робота з масивами даних в мові C#

Тема 9. C# як об'єктно-орієнтована мова. Поняття про визначення класу та створення об'єкту. Типи-значення та типи-посилання. Визначення класу, доступ до даних-членів класу. Визначення методів класу.

Тема 10. Використання методів. Використання параметрів. Параметри за замовчуванням. Рекурсія.

Тема 11. Загальний огляд масивів як способу групування даних. Створення та використання масивів у мові C#. Використання конструкції new. Багатомірні масиви. Масиви як параметри методів. Методи, що повертають масиви. Цикл по елементах масива (foreach).

Кредит 5. Використання класів в мові C#

Тема 12. Створення об'єкту даного класу. Конструктор класу. Конструктор за замовчуванням та конструктор з параметрами. Створення параметризованих об'єктів класу.

Тема 13. Деструктор класу. Поняття про систему збору «сміття». Використання службового слова this.

Тема 14. Управління доступом до членів класу, інкапсуляція. Передача об'єктів методам класу. Правила використання модифікаторів ref- та out- для параметрів методів.

3. Структура навчальної дисципліни

Назви кредитів і тем	Кількість годин				
	усього	у тому числі			
		л	П	лаб	інд
Кредит 1. Базові поняття програмування.					
Тема 1. Основні поняття програмного забезпечення.	8	2			6
Тема 2. Представлення даних та двійкова арифметика.	12	2			10
Тема 3. Процедури та функції як засіб створення модульних програм.	10			4	6
Усього:	30				
Кредит 2. Загальний огляд платформи Microsoft .NET					
Тема 4. Огляд платформи Microsoft .NET.	15	2		4	9
Тема 5. Компоненти платформи .NET.	15	2		4	9
Усього:	30				
Кредит 3. Основи мови програмування C#					
Тема 6. Структура програми мовою C#.	10	2		2	6
Тема 7. Огляд загальної системи типів.	10	2		2	6
Тема 8. Використання основних програмних конструкцій у мові C#.	10			2	8
Усього:	30	4		6	20
Кредит 4. Поняття про класи та методи. Робота з масивами даних в мові C#					
Тема 9. Поняття про визначення класу та створення об'єкту.	10	2		2	6
Тема 10. Використання методів.	10	2		2	6
Тема 11. Загальний огляд масивів як способу групування даних.	10	2		4	4
Усього:	30	6		8	16
Кредит 5. Використання класів в мові C#					
Тема 12. Створення об'єкту даного класу.	10	2		2	6
Тема 13. Деструктор класу.	10			2	8
Тема 14. Управління доступом до членів класу, інкапсуляція.	10				10
Усього:	30	2		4	24
Усього годин:	150	20		30	100

4. Теми лекційних занять

№ з/п	Назва теми	Кількість годин
<i>Кредит 1. Базові поняття програмування.</i>		
1	Тема 1. Основні поняття програмного забезпечення.	2
2	Тема 2. Представлення даних та двійкова арифметика.	2
<i>Кредит 2. Загальний огляд платформи Microsoft .NET</i>		
3	Тема 4. Огляд платформи Microsoft .NET.	2
4	Тема 5. Компоненти платформи .NET.	2
<i>Кредит 3. Основи мови програмування C#</i>		
5	Тема 6. Структура програми мовою C#.	2
6	Тема 7. Огляд загальної системи типів.	2
<i>Кредит 4. Поняття про класи та методи. Робота з масивами даних в мові C#</i>		
7	Тема 9. Поняття про визначення класу та створення об'єкту.	2
8	Тема 10. Використання методів.	2
9	Тема 11. Загальний огляд масивів як способу групування даних.	2
<i>Кредит 5. Використання класів в мові C#</i>		
10	Тема 12. Створення об'єкту даного класу.	2
Усього:		20

5. Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
<i>Кредит 1. Базові поняття програмування.</i>		
1	Тема 3. Процедури та функції. Поняття аргументу, поняття параметра. Формальні і фактичні параметри, їх зв'язок за значенням та за адресою. Вхідні та вихідні параметри.	4-2
<i>Кредит 2. Загальний огляд платформи Microsoft .NET</i>		
2	Тема 4. Огляд платформи Microsoft .NET. Огляд циклу розробки програмного забезпечення.	4
3	Тема 5. Компоненти платформи .NET: EXE та DLL, CLR	4
<i>Кредит 3. Основи мови програмування C#</i>		
4	Тема 6. Структура програми мовою C#. Основні операції вводу/виводу. Засоби форматування.	2
5	Тема 7. Огляд загальної системи типів. Створення типів даних користувача в мові C#	2
6	Тема 8. Використання основних програмних конструкцій у мові C#. Розгалуження. Цикли.	2
<i>Кредит 4. Поняття про класи та методи. Робота з масивами даних в мові C#</i>		
7	Тема 9. Поняття про визначення класу та створення об'єкту. Типи-значення та типи-посилання. Визначення класу, доступ до даних-членів класу.	2
8	Тема 10. Використання методів, параметрів. Рекурсія.	2
9	Тема 11. Створення та використання масивів у мові C#.	4
<i>Кредит 5. Використання класів в мові C#</i>		
10	Тема 12. Конструктор класу за замовчуванням та конструктор з параметрами.	2
11	Тема 13. Деструктор класу. Використання слова this.	2
Усього:		30

6. Самостійна робота

№ з/п	Назва теми	Кількість годин
<i>Кредит 1. Базові поняття програмування.</i>		
1	Тема 1. Процедурні та об'єктно-орієнтовані мови програмування. Місце мови C# серед сучасних мов програмування	6
2	Тема 2. Правила іменування змінних. Операції. Використання програмного синтаксису.	10
3	Тема 3. Формальні і фактичні параметри, їх зв'язок за значенням та за адресою. Вхідні та вихідні параметри. Повернення скалярного значення функцією.	6
<i>Кредит 2. Загальний огляд платформи Microsoft .NET</i>		
4	Тема 4. Набуття практичних навичок роботи в середовищі Visual Studio .NET. Переваги платформи .NET.	9
5	Тема 5. Компоненти платформи .NET. набір програмних засобів.	9
<i>Кредит 3. Основи мови програмування C#</i>		
6	Тема 6. Структура програми мовою C#. Засоби форматування. Компілювання, виконання та відлагодження.	6
7	Тема 7. Огляд загальної системи типів. Перетворення типів (явне та неявне).	6
8	Тема 8. Використання основних програмних конструкцій у мові C#. Поняття виключної ситуації. Обробка основних виключних ситуацій. Генерування виключних ситуацій.	8
<i>Кредит 4. Поняття про класи та методи. Робота з масивами даних в мові C#</i>		
9	Тема 9. Визначення класу, доступ до даних-членів класу. Визначення методів класу.	6
10	Тема 10. Параметри за замовчуванням. Рекурсія.	4
11	Тема 11. Загальний огляд масивів як способу групування даних. Багатомірні масиви. Масиви як параметри методів. Методи, що повертають масиви. Цикл по елементах масива (foreach).	16
<i>Кредит 5. Використання класів в мові C#</i>		
12	Тема 12. Створення параметризованих об'єктів класу.	6
13	Тема 13. Деструктор класу. Поняття про систему збору «сміття».	8
14	Тема 14. Управління доступом до членів класу, інкапсуляція. Передача об'єктів методам класу. Правила використання модифікаторів get- та out- для параметрів методів.	10
Усього:		100

7. Індивідуальне науково-дослідне завдання

Індивідуальне науково-дослідне завдання складається з розробки об'єктно-орієнтованої програми «Робота з матрицями».

Основне завдання цього виду діяльності – набуття практичних навичок самостійної розробки професійного програмного забезпечення та використання сучасних інформаційних технологій для розв'язання різноманітних задач у практичній діяльності.

Вимогою до завдання є створення власного класу "Матриця".

Клас повинен передбачати такі методи:

- конструктор, в якому задаються розміри матриці;
- метод для заповнення матриці випадковими числами;
- перевантаження операції «*» для множення матриці (приймає у вигляді параметра на вхід матрицю В) і повертає результат перемноження поточної матриці на матрицю В у вигляді нового об'єкта класу "Матриця";
- метод, що знаходить і повертає визначник матриці;
- метод, що знаходить і повертає транспоновану матрицю;
- метод, що знаходить і повертає обернену матрицю.
-

8. Форми роботи та критерії оцінювання

Рейтинговий контроль знань студентів здійснюється за 100-бальною шкалою:

Шкала оцінювання: національна та ECTS

ОЦІНКА ЄКТС	СУМА БАЛІВ	ОЦІНКА ЗА НАЦІОНАЛЬНОЮ ШКАЛОЮ	
		екзамен	залік
A	90-100	5 (відмінно)	5/відм./зараховано
B	80-89	4 (добре)	4/добре/ зараховано
C	65-79		
D	55-64	3 (задовільно)	3/задов./ зараховано
E	50-54		
FX	35-49	2 (незадовільно)	Не зараховано

Форми поточного та підсумкового контролю. Комплексна діагностика знань, умінь і навичок студентів із дисципліни здійснюється на основі результатів проведення поточного й підсумкового контролю знань (КР). Поточне оцінювання (індивідуальне, групове і фронтальне опитування, самостійна робота, самоконтроль). Завданням поточного контролю є систематична перевірка розуміння та засвоєння програмового матеріалу, виконання практичних, лабораторних робіт, уміння самостійно опрацювати тексти, складання конспекту рекомендованої літератури, написання і захист реферату, здатності публічно чи письмово представляти певний матеріал.

Завданням підсумкового контролю є перевірка глибини засвоєння студентом програмового матеріалу.

Критерії оцінювання відповідей на лабораторних заняттях:

Студенту виставляється відмінно, якщо студент здатний самостійно здійснювати основні види навчальної діяльності. Знання студента є глибокими, міцними, узагальненими; студент вміє застосовувати знання творчо, його навчальна діяльність позначена вмінням самостійно оцінювати різноманітні життєві ситуації, явища, факти, виявляти і відстоювати особисту позицію.

Студенту виставляється дуже добре, якщо студент знає істотні ознаки понять, явищ, закономірностей, зв'язків між ними, а також самостійно застосовує знання в нестандартних ситуаціях, володіє розумовими операціями, вміє робити висновки, виправляти допущені помилки. Відповідь повна, правильна, логічна, обґрунтована.

Студенту виставляється добре, якщо студент знає ознаки понять, явищ, закономірностей, зв'язків між ними на середньому рівні, а також самостійно застосовує знання в стандартних ситуаціях, володіє розумовими операціями, вміє робити висновки, виправляти допущені помилки. Відповідь повна, правильна, логічна, обґрунтована.

Студенту виставляється достатньо, якщо відповідь студента при відтворенні навчального матеріалу елементарна, зумовлюється початковими уявленнями про предмет

вивчення. Студент відтворює основний навчальний матеріал, здатний виконувати завдання за зразком, володіє елементарними вміннями навчальної діяльності.

Студенту виставляється задовільно, якщо відповідь студента при відтворенні навчального матеріалу елементарна, зумовлюється початковими уявленнями про предмет вивчення. Студент відтворює основний навчальний матеріал.

Кількість балів у кінці **семестру** повинна складати від 250 до 500 балів (за 5 кредитів), тобто сума балів за виконання усіх завдань.

Відповідний **розподіл балів, які отримують студенти** за 5 кредитів

Поточне оцінювання та самостійна робота														КР	Накопичувальні бали/ Сума
T1	T2	T3	T4	T5,6	T7	T8	T9	T10	T11	T12	T13	T14	T15		
30	35	35	10	40	30	35	35	10	20	20	30	35	35	100	500/100*

***Примітка.** Коефіцієнт для іспиту – 0,6. Іспит оцінюється в 40 б.

9. Засоби діагностики

Засобами діагностики та методами демонстрування результатів навчання є: завдання до лабораторних занять, завдання для самостійної та індивідуальної роботи, презентації результатів досліджень, тестові завдання, контрольні роботи.

10. Методи навчання

Усний виклад матеріалу: наукова розповідь, спрямована на аналіз фактичного матеріалу; пояснення – вербальний метод навчання, за допомогою якого розкривається сутність певного явища, закону, процесу; проблемне навчання, робота з підручником та додатковими джерелами.

Лекційні заняття призначені для теоретичного осмислення і узагальнення складних розділів курсу, які освітлюються, в основному, на проблемному рівні та у формі діалогічно-проблемних лекцій.

Лабораторні заняття є аудиторними, проводяться по наперед відомих темах у вигляді активних форми проведення занять. Вони призначені для закріплення і глибшого вивчення певних аспектів лекційного матеріалу на практиці.

Самостійна робота є позааудиторною і призначена для самостійного ознайомлення студента з певними розділами курсу за рекомендованими педагогом матеріалами і підготовки до виконання індивідуальних завдань по курсу.

Поточний рейтинг-контроль проводиться викладачем в процесі проведення всіх видів занять. Проміжний рейтинг-контроль призначений для практичної комплексної оцінки освоєння розділів курсу і здійснюється шляхом підготовки студентами відповідей на поставлені питання.

Кінцевий контроль знань з дисципліни проводиться під час складання іспиту.

11. Рекомендована література

Базова

1. Коноваленко І.В. Програмування мовою C# 6.0. Тернопіль: ТНТУ, 2016. 229 с.
2. Основи програмування: конспект лекцій для студентів спеціальності 121 "Інженерія програмного забезпечення" / уклад. М.В.Делявський. Луцьк: Луцький НТУ, 2016. 84 с.

3. Павловская Т.А. Программирование на языке высокого уровня С#: 2-е изд., исправленное. М.: Интуит, 2016. 245 с.
4. Michaelis Mark. Essential C# 7.0. Addison-Wesley Professional, 2018. 1088 p.
5. Васильев А.Н. Программирование на С# для начинающих. Основные сведения. М.: Эксмо, 2018. 586 с.

Допоміжна

1. Васильев А.Н. Программирование на С# для начинающих. Особенности языка. Москва: Эксмо, 2019. 528 с.
2. Мюллер Джон, Семпф Билл, Сфер Чак. С# для чайников: пер. с англ. и ред. канд. техн. наук И.В. Красикова. М.: Диалектика, 2019. 608 с.

12. Інформаційні ресурси

1. Інтерактивний онлайн-підручник з основ С#-
https://ulearn.me/Course/BasicProgramming/Hello_world_70597ba7-f436-4301-816d-17dc34551bdb?autoplay=1
2. Класи (class), конструктори (constructor) і властивості (property) в С # - Вчимо Шарп # 9 -
<https://www.youtube.com/watch?v=0Q1IHASPzms&list=PLIIXgDT0bKw4OmiZ9yGmShKsY0XncViZ8&index=9>
3. Програмування на мові С #: розробка консольних додатків Є.В. Кудріна, М.В. Огнева, М.С. Портенко <http://www.intuit.ru/department/se/csharpdevtools/>
4. Розробка додатків на С # в середовищі Visual Studio В.М. Снетков- <http://www.intuit.ru/department/se/csharpvs2005>
5. Технологія Microsoft ADO .NET
 Ч.А. Каріїв- <http://www.intuit.ru/department/se/msadonet/>