

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА

ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ

Кафедра теорії та технології програмування



Кашпур О.Ф.

« 22 » серпня 2020 року

**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ**

для студентів

галузь знань **12 «Інформаційні технології»**
(шифр і назва)
спеціальність **122 «Комп'ютерні науки»**
(шифр і назва спеціальності)
освітній рівень **бакалавр**
(молодший бакалавр, бакалавр, магістр)
освітня програма **«Інформатика»**
(назва освітньої програми)
вид дисципліни **обов'язкова**

Форма навчання	денна
Навчальний рік	2020/2021
Семестр	3
Кількість кредитів ECTS	4
Мова викладання, навчання та оцінювання	українська
Форма заключного контролю	іспит

Викладачі: к.ф.-м.н., доц. Омельчук Л.Л. (лекції, лабораторні заняття),
к.п.н., асистент Русіна Н.Г. (лабораторні заняття)

Пролонговано: на 20__/20__ н.р. _____ (_____) «__» 20__ р.
(підпис, ПІБ, дата)

на 20__/20__ н.р. _____ (_____) «__» 20__ р.
(підпис, ПІБ, дата)

Розробник: Омельчук Людмила Леонідівна, к.ф.-м.н., доцент кафедри «Теорії та технології програмування»

ЗАТВЕРДЖЕНО

Зав. кафедри «Теорії та технології програмування»

Нікітченко М.С.

(підпис)

(прізвище та ініціали)

Протокол № 1 від «28» 08 2020 р.

Схвалено гарантом освітньо-професійної програми «Інформатика»

Омельчук Л.Л. «28» 08 2020 року

Схвалено науково-методичною комісією факультету комп'ютерних наук та кібернетики

Протокол від «28» 08 2020 року № 1

Голова науково-методичної комісії

(підпис)

Омельчук Л.Л.

(прізвище та ініціали)

1. Мета дисципліни – засвоєння базових знань з основ об'єктно-орієнтованого програмування, включаючи основні поняття, парадигми та принципи об'єктно-орієнтованого програмування. Оволодіння базовими навичками проектування програмних систем, роботи з найбільш вживаними шаблонами проектування, набуття навичок об'єктно-орієнтованого програмування та оволодіння мовою програмування C#.

2. Попередні вимоги до опанування або вибору навчальної дисципліни (за наявності):

1. *Знати:* основні поняття програмування та принципи розробки програм; базові класичні алгоритми та різновиди структур даних
2. *Вміти:* проектувати, розробляти та тестувати програми на базовому рівні.
3. *Володіти елементарними навичками:* програмування мовами C, C++.

3. Анотація навчальної дисципліни:

Навчальна дисципліна “Об'єктно-орієнтоване програмування” є складовою освітньо-професійної програми підготовки фахівців за першим (бакалаврським) рівнем вищої освіти галузі знань 12 „Інформаційні технології” зі спеціальності 122 „Комп'ютерні науки”, освітньо-професійної програми „Інформатика”.

Дана дисципліна є обов'язковою навчальною дисципліною за *програмою “Інформатика”*.

Викладається у 3 семестрі 2 курсу в обсязі – 120 год.

(4 кредити ECTS) зокрема: лекції – 28 год., лабораторні – 26 год., консультації – 2 год., самостійна робота – 64 год. У курсі передбачено 2 частини та 2 контрольні роботи. Завершується дисципліна – іспитом в 3 семестрі.

В результаті вивчення навчальної дисципліни студент повинен:

знати основні поняття об'єктно-орієнтованого програмування, основні етапи життєвого циклу ПС, шаблони, антишаблони та принципи об'єктно-орієнтованого проектування програмного забезпечення.

вміти застосовувати на практиці інструментальні програмні засоби проектування та розробки програмного забезпечення, обґрунтовувати власний погляд на задачу, спілкуватися з колегами з питань проектування та розробки програм, складати письмові звіти,

Для допуску до дисципліни „Об'єктно-орієнтоване програмування” освітньо-професійної програми «Інформатика» студент повинен опанувати компетентності та результати навчання, які надає дисципліна „Програмування” програми «Інформатика». Дисципліна „Об'єктно-орієнтоване програмування” є базовою для засвоєння дисципліни «Інструментальні середовища та технології програмування», дисциплін спеціалізації та дисциплін вільного вибору студента програміського спрямування програми «Інформатика».

4. Завдання (навчальні цілі):

набуття знань, умінь та навичок (компетентностей) на рівні новітніх досягнень у програмуванні, відповідно освітньої кваліфікації «Бакалавр з комп'ютерних наук». Зокрема, розвивати:

- здатність застосовувати знання у практичних ситуаціях;
- здатність оцінювати та забезпечувати якість виконуваних робіт;
- 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;
- здатність до алгоритмічного мислення.

5. Результати навчання за дисципліною:

Результат навчання (1. знати; 2. вміти; 3. комунікація; 4. автономність та відповідальність)		Форми (та/або методи і технології) викладання і навчання	Методи оцінювання та пороговий критерій оцінювання (за необхідності)	Відсоток у підсумковій оцінці з дисципліни
Код	Результат навчання			
РН1.1	Знати основні поняття об'єктно-орієнтованого програмування.	Лекція, лабораторне заняття	Контрольна робота, домашні завдання, іспит	16%
РН1.2	Знати основні етапи життєвого циклу ПС.	Лекція, лабораторне заняття	Контрольна робота, домашні завдання, Іспит	15%
РН1.3	Знати шаблони, антишаблони та принципи об'єктно-орієнтованого проектування програмного забезпечення.	Лекція, лабораторне заняття	Контрольна робота, домашні завдання, Іспит	20%
РН2.1	Вміти застосовувати на практиці інструментальні програмні засоби проектування та розробки програмного забезпечення.	Лабораторне заняття, самостійна робота	Захист лабораторної роботи, іспит	23%
РН3.1	Обґрунтовувати власний погляд на задачу, спілкуватися з колегами з питань проектування та розробки програм, складати письмові звіти	Лабораторне заняття	Захист ЛР	10%
РН4.1	Організувати свою самостійну роботу для досягнення результату	Самостійна робота	Захист лабораторної роботи	8%
РН4.2	Відповідально ставитися до виконуваних робіт, нести відповідальність за їх якість	Лабораторне заняття	Захист лабораторної роботи	8%

6. Співвідношення результатів навчання дисципліни із програмними результатами навчання

Результати навчання дисципліни	РН 1.1	РН 1.2	РН 1.3	РН 2.1	РН 3.1	РН 4.1	РН 4.2
Програмні результати навчання							
(з опису освітньої програми)							
ПР8. Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук.	+			+			+
ПР14. Застосовувати знання методології та CASE-засобів проектування складних систем, методів структурного аналізу систем, об'єктно-орієнтованої методології проектування в процесі побудови і практичного застосування функціональних моделей організаційно-економічних і виробничо-технічних систем.		+	+		+	+	

7. Схема формування оцінки.

7.1 Форми оцінювання студентів:

- семестрове оцінювання:

1. Контрольна робота (тест) 1: РН 1.1., РН 1.2 — 11 балів/6 балів.
2. Контрольна робота (тест) 2: РН1.3 - 8 балів/4 бали.
3. Домашні завдання 1-7: РН 1.1., РН 1.2, РН1.3- 11 балів/6 балів.
4. Захист лабораторної роботи 1 (проекту): РН 2.1, РН3.1, РН4.1, РН4.2 – 10 балів/6 балів.
5. Захист лабораторної роботи 2 (проекту): РН 2.1, РН3.1, РН4.1, РН4.2 – 10 балів/6 балів.
6. Захист лабораторної роботи 3 (проекту): РН 2.1, РН3.1, РН4.1, РН4.2 – 10 балів/6 балів.

Одну з форм контролю (усі домашні завдання, або одну з лабораторних робіт) студент має право замінити на здачу сертифікату з можливістю отримання усіх балів, які виділені для обраної форми контролю. А саме: за контрольні роботи – 10 балів, за лабораторну №1 – 11 балів, за лабораторну роботу №2 – 8 балів, за лабораторну роботу №3 – 11 балів. Для реалізації цього права студент повинен до 1.10.2020 написати та прикріпити в classroom заяву (в якій зазначено форму контролю, яку замінюємо та інформація про обраний курс) зі своїм персональним підписом. Курс повинен бути попередньо узгоджений з викладачем. Дата отримання сертифікату повинна бути не від 01.09.2020 до 01.12.2020 включно.

- підсумкове оцінювання (у формі іспиту):

- максимальна кількість балів які можуть бути отримані студентом: 40 балів / 24 бали;
 - результати навчання які будуть оцінюватись: РН1.1, РН1.2, РН1.3, РН2.1;
 - форма проведення і види завдань: письмова робота.
- Види завдань: 8 тестових та 6 письмових завдань.

Критерії оцінювання на іспиті

Завдання	Тема завдання	Максимальний відсоток від 40 балів	Всього відсотків
Завдання 1	Письмове запитання з S.O.L.I.D.		28%
Завдання 2, 3, 4	Письмові запитання з ООП та С#	По 7%	
Завдання 5, 6, 9, 10, 12	Тестове завдання з С# та LINQ	По 5%	35%
Завдання 7, 8	Тестові завдання з ООП, шаблонів, антишаблонів		
Завдання 11	Написати XPath	10%	10%
Завдання 13	Діаграми UML	15%	15%
Завдання 14	Написати LINQ	12%	12%
			100%

Умови домашніх завдань

1. Use-case-діаграма та діаграма класів для заданої предметної області.
2. Діаграма класів та діаграма послідовності використання для заданої предметної області.
3. Завдання з посібника [9] ст. 44-45.

4. Завдання з посібника [9] ст. 73-76.
5. Завдання з посібника [9] ст. 77-80.
6. Завдання з посібника [9] ст. 47-55.
7. Завдання з посібника [9] ст. 68-72, 81-83.

Умови лабораторних робіт:

Лабораторна робота 1: Розробка програми для роботи з електронними таблицями (аналіз та обчислення виразів).

Лабораторна робота 2: Робота з XML-документами.

Лабораторна робота 3: Розробка програм обробки файлів та редагування текстів.

Деталізовані умови розміщено за посиланнями:

<https://classroom.google.com/c/MTQ4MzY1MjU0Mjgxcjcj?cjc=svftkku>

Запитання для підготовки до іспиту

1. Основні принципи моделювання програмних систем.
2. Життєвий цикл ПС. Моделі життєвого циклу ПС. Ітеративні моделі життєвого циклу. Керування ризиками.
3. Життєвий цикл ПС. Моделі життєвого циклу ПС. Модель Гантера «Фази-Функції».
4. Діаграми класів та шаблони проектування. Структура шаблонів. Класифікація шаблонів: породжуючі, структурні, поведінки.
5. Різновиди шаблонів проектування.
6. Шаблон „абстрактна фабрика”.
7. Шаблон „прототип”.
8. Шаблон „одиначний об’єкт”.
9. Шаблон „міст”.
10. Шаблон „ланцюг обов’язків”.
11. Шаблон „декоратор”.
12. Шаблон „ітератор”.
13. Антишаблони.
14. Огляд технології .NET Framework.
15. Середовище CLR.
16. Бібліотека класів Framework Class Library (FCL).
17. Загальна структура C#-програми.
18. Типи значень (вбудовані типи значень, порядковий тип (enum), структури (struct)). Типи посилань (вбудовані типи посилань (object, string), масиви, делегати, інтерфейси, класи).
19. Ітератори.
20. Робота з рядками.
21. Перетворення типів в C#.
22. Оператори обробки виключень в C#.
23. Неявні типи, анонімні типи (тип var) та тип dynamic.
24. Робота з файлами в директорії.
25. Створення підкаталогів.
26. Робота з файлами (читання, запис, створення, видалення).

27. Основні компоненти XML-документа.
28. Моделі DOM та SAX для обробки XML.
29. Застосування XmlDocument. Застосування XmlReader.
30. LINQ to XML.
31. Порівняння LINQ to XML та DOM.
32. Мова запитів XPath.
33. Трансформація XML.
34. Трансформація XML з використанням XSLT.
35. Шаблон XSL.
36. Лямбда-вирази. Призначення та синтаксис.
37. LINQ (Language Integrated Query – мова інтегрованих запитів). Призначення та синтаксис.
38. LINQ to Objects
39. LINQ to DataSet та SQL
40. LINQ to Entities

Студент не допускається до іспиту, якщо під час семестру набрав менше ніж 24 балів. Студент допускається до іспиту за умови виконання 70% передбачених планом лабораторних робіт.

7.2 Організація оцінювання:

Терміни проведення форм оцінювання:

1. Контрольна робота 1 (тест): до 7 тижня семестру.
2. Контрольна робота 2 (тест): до 14 тижня семестру.
3. Домашні завдання 1: до 3 тижня семестру.
4. Домашні завдання 2: до 4 тижня семестру.
5. Домашні завдання 3: до 5 тижня семестру.
6. Домашні завдання 4: до 6 тижня семестру.
7. Домашні завдання 5: до 9 тижня семестру.
8. Домашні завдання 6: до 11 тижня семестру.
9. Домашні завдання 7: до 12 тижня семестру.
10. Захист лабораторної роботи 1 (проекту): до 7 тижня семестру.
11. Захист лабораторної роботи 2 (проекту): до 10 тижня семестру.
12. Захист лабораторної роботи 3 (проекту): до 13 тижня семестру.

Студент має право на одне перескладання кожної контрольної роботи із можливістю отримання максимально 80% початково визначених за цю контрольну роботу балів. Термін перескладання визначається викладачем.

У разі неякісного виконання лабораторної роботи, викладач має право не зарахувати лабораторну роботу, або знизити за неї бали.

Студент має право здавати лабораторні роботи після закінчення визначеного для них терміну, але з втратою одного балу за кожен тиждень, який пройшов з моменту закінчення терміну її здачі.

7.3 Шкала відповідності оцінок

Відмінно / Excellent	90-100
Добре / Good	75-89
Задовільно / Satisfactory	60-74
Незадовільно / Fail	0-59

8. Структура навчальної дисципліни. Тематичний план лекцій і лабораторних занять

№ лекції	Назва лекції	Кількість годин		
		Лекції	Лаборат. заняття	Сам. р-та
Частина 1. Об'єктно-орієнтована методологія.				
1.	Тема 1. Основні поняття ООП. Порівняння ООП з парадигмою процедурного програмування. Втілення ООП в мовах програмування. Самостійна робота: логічне програмування. Методи і алгоритми обчислень. Структури даних і механізми управління. Виконання лабораторної роботи 1.	2	2	4
2.	Тема 2. Основні поняття об'єктно-орієнтованого програмування. Основні етапи життєвого циклу програмних систем. Самостійна робота: функціональне програмування, опрацювання лекційного матеріалу. Виконання домашнього завдання 1, або виконання лабораторної роботи 1.	2	2	4
3.	Тема 3. Основні принципи моделювання програмних систем. Огляд мови UML. Самостійна робота: застосування шаблонів проектування та UML для об'єктно-орієнтованого моделювання ПС. Виконання домашнього завдання 2, виконання лабораторної роботи 1.	2	2	5
4.	Тема 4. .NET Framework. Порівняльний аналіз мов C++ та C#. Самостійна робота: опрацювання лекційного матеріалу. Виконання домашнього завдання 3, виконання лабораторної роботи 1.	2	2	5
5.	Тема 5. Порівняльний аналіз мов C++ та C#. Класи, перетворення типів, оператори та вирази, оператори обробки виключень, неявні, анонімні та динамічні типи даних. Самостійна робота: опрацювання лекційного матеріалу. Виконання домашнього завдання 4, виконання лабораторної роботи 1.	2	2	5
6.	Тема 6. Огляд технології .NET Framework. Середовище CLR. Бібліотека класів Framework Class Library (FCL). Самостійна робота: Основи синтаксичного аналізу та обчислення виразів. Використання ANTLR для виконання лабораторної роботи №1. Виконання домашнього завдання 5. Виконання лабораторної роботи 2.	2	2	5
7.	Тема 7. Вимоги і рекомендації з написання коду. Рефакторинг. Мова запитів XPath. Трансформація XML. Трансформація XML з використанням XSLT. Самостійна робота: Основні компоненти XML документа. Застосування XML документа. Моделі DOM	2	1	6

	<i>та SAX для обробки XML. Виконання домашнього завдання 5. Виконання лабораторної роботи 1.</i>			
	<i>Контрольна робота 1</i>		1	
	Всього по частині 1	14	14	34
Частина 2. Функціональні можливості C#. Шаблони проектування. Огляд технології .NET Core				
8.	Тема 8. Функціональні можливості C#. <i>LINQ to Object, LINQ to XML. Лямбда-вирази. Застосування лямбда-виразів.</i> <i>Самостійна робота: опрацювання лекційного матеріалу. Виконання лабораторної роботи 2. Виконання домашнього завдання 5.</i>	2	2	5
9.	Тема 9. Шаблони проектування. <i>Призначення. Класифікація.</i> <i>Самостійна робота: опрацювання лекційного матеріалу. Виконання лабораторної роботи 2. Виконання домашнього завдання 6.</i>	2	2	5
10.	Тема 10. <i>Породжуючі патерни (шаблони) проектування.</i> <i>Самостійна робота: опрацювання лекційного матеріалу. Виконання лабораторної роботи 3. Виконання домашнього завдання 6.</i>	2	2	5
11.	Тема 11. <i>Структурні патерни (шаблони) проектування.</i> <i>Самостійна робота: опрацювання лекційного матеріалу. Виконання лабораторної роботи 3. Виконання домашнього завдання 7.</i>	2	2	5
12.	Тема 12. <i>Патерни поведінки. Антишаблони..</i> <i>Самостійна робота: опрацювання лекційного матеріалу. Виконання лабораторної роботи 3. Виконання домашнього завдання 7, підготовка доповідей за патернами проектування.</i>	2	2	5
13.	Тема 13. <i>Принципи проектування S.O.L.I.D. та інші.</i> <i>Самостійна робота: опрацювання лекційного матеріалу. Виконання лабораторної роботи 3.</i>	4	1	5
	<i>Контрольна робота 2</i>		1	
	Контроль за підсумками лабораторних робіт 2 та 3			
	Всього по частині 2	14	12	30
	Консультація		2	
	ВСЬОГО	28	30	60

Загальний обсяг 120 год., в тому числі:

Лекцій – **28 год.**

Лабораторні заняття - **26 год.**

Консультації – **2 год.**

Самостійна робота - **64 год.**

9. Рекомендовані джерела:

Основна

1. <http://csc.knu.ua/uk/person/omelchuk>
2. В.В. Зубенко, Л.Л. Омельчук. Програмування : навчальний посібник (гриф МОН України) / - К. : ВПЦ "Київський університет", 2011. - 623 с.
3. <http://msdn.microsoft.com/ru-RU/>
5. Лаврищева Е.М. Современные методы программирования: возможности и инструменты // Проблемы програмування. – 2006.– № 2-3. – С.60-74.
6. Буч Г. Объектно-ориентированное программирование. – Киев, Диалектика, 1993
7. Буч Г., Якобсон А., Рамбо Дж. UML. Классика CS. 2-е изд. / Пер. с англ.; Под общей редакцией проф. С. Орлова — СПб.: Питер, 2006. — 736 с. ISBN 5-469-00599-2
8. Крэг Ларман. Применение UML 2.0 и шаблонов проектирования = Applying UML and Patterns : An Introduction to Object-Oriented Analysis and Design and Iterative Development. — 3-е изд. — М.: Вильямс, 2006. — 736 с. — ISBN 0-13-148906-2
9. Бойко Б.І. Об'єктно-орієнтоване програмування. Лабораторний практикум: навчальний посібник / Б.І. Бойко, Л.Л. Омельчук, Н.Г. Русіна. - Київ. - 90 с. http://csc.knu.ua/media/filer_public/4a/35/4a3533cd-4ec7-45f3-85d2-4edaafdf1b82/oop_2016.pdf

Додаткова:

10. Леффингуелл Д, Уидриг Д. Принципы работы с требованиями к программному обеспечению. Унифицированный подход. . – М.: Изд. дом “Вильямс”, 20024. Д. Дайтеман и др. Программирование в среде Delphi – – Київ, 1995.
11. Ставровский А.Б, Карнаух Т.О. Перші кроки програмування. -К.:Диалектика.-2005, с.389.
12. Кнут Д. Искусство программирования.Т.1,2,3. Изд 3-е, испр. -М.СБП.К.:Вильямс, 2001.
13. Вирт Н. Систематическое программирование. Введение.- М.:Мир,1987.с.184.
14. Вирт Н. Алгоритмы и структуры данных. - М.:Мир,1989.с.263.
15. Джеймс Р. Грофф, Пол Н. Вайнберг, Эндрю Дж. Оппель. SQL: полный справочник = SQL: The Complete Reference. — 3-е изд. — М.: Вильямс, 0. — 960 с. — ISBN 978-5-8459-1654-9
15. Алекс Кригель, Борис Трухнов. SQL. Библия пользователя. Язык запросов SQL = SQL Bible. — 2-е изд. — М.: Диалектика, 2009. — 752 с. — ISBN 978-5-8459-1546-7
16. К. Дж. Дейт. Введение в системы баз данных / Пер. с англ. — 8-е изд. — М.: Вильямс, 2005. — 1328 с.

10. Додаткові ресурси:

<https://classroom.google.com/c/MTQ4MzY1MjU0Mjgxcjcjcvsvftkku>