

Міністерство освіти і науки України  
Харківський національний університет імені В.Н. Каразіна  
Кафедра моделювання систем і технологій

**«ЗАТВЕРДЖУЮ»**

Проректор з  
науково-педагогічної роботи



А.В.Пантелеймонов

2019 року

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ  
«ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ»

рівень вищої освіти	перший (бакалаврський)
галузь знань	12 Інформаційні технології
напрямок підготовки	122 Комп'ютерні науки
освітня програма	Комп'ютерні науки
спеціалізація	
вид дисципліни	Обов'язкова
факультет	Комп'ютерних наук

2019/2020 навчальний рік

Програму рекомендовано до затвердження вченою радою факультету комп'ютерних наук

Протокол від « 27 » червня 2019 року № 2

РОЗРОБНИК ПРОГРАМИ: кандидат технічних наук, доцент кафедри моделювання систем і технологій Нагорний Костянтин Анатолійович

Програму схвалено на засіданні кафедри моделювання систем і технологій

Протокол від « 30 » травня 2019 року № 15

Завідувач кафедри моделювання систем і технологій

  
М. В. Ткачук

Програму погоджено методичною радою факультету комп'ютерних наук

Протокол від « 20 » червня 2019 року № 9

Голова методичної комісії

  
А. Г. Бердніков

## ВСТУП

Програма навчальної дисципліни «Об'єктно-орієнтоване програмування» складена відповідно до освітньо-професійної програми підготовки першого (бакалаврського) рівня спеціальності 122 – комп'ютерні науки.

### 1. Опис навчальної дисципліни.

#### 1.1. Мета викладання навчальної дисципліни.

Метою вивчення курсу «Об'єктно-орієнтоване програмування» є формування у студентів теоретичних знань та практичних навичок із побудови, керування, модернізації, моніторингу та аналізу продуктивності, діагностики та розв'язання проблем сучасних комп'ютерних мереж. В рамках курсу студенти повинні познайомитися з основами проектування та розгортання мереж (дротові та бездротові), технічними та програмними засобами, як компонентами комп'ютерної мережі.

#### 1.2 Основні завдання вивчення дисципліни.

*Задачами вивчення дисципліни є:* ознайомлення із принципами об'єктно-орієнтованої розробки та первинного аналізу предметної області; побудова базових моделей із використанням сучасних практик проектування програмних систем (ПС).

*Завданнями вивчення навчальної дисципліни є:* вивчення основних підходів та принципів ОО-розробки ПС; специфіки використання базових та просунутих елементів, конструкцій та сучасних можливостей мови програмування Java; ознайомлення із базовими елементами мови проектування UML та побудова моделей відповідних програмних рішень; ознайомлення із деякими шаблонами проектування.

1.3. Кількість кредитів - 4.

1.4. Загальна кількість годин - 120.

1.5. Характеристика навчальної дисципліни	
<u>Нормативна / за вибором</u>	
Денна форма навчання	Денна форма навчання
Рік підготовки	
2-й	-й
Семестр	
3-й	-й
Лекції	
32 год.	год.
Практичні заняття	
год.	год.
Лабораторні заняття	
32 год.	год.
Самостійна робота	
56 год.	год.
Індивідуальні завдання	
год.	

## 1.6. Заплановані результати навчання.

Згідно з вимогами освітньо-професійної програми студенти повинні досягти результатів навчання.

У результаті вивчення даного курсу студент повинен **знати**: сучасні технології та інструментальні засоби розробки програмних систем, вміти їх застосовувати на всіх етапах життєвого циклу; розуміти основні принципи об'єктно-орієнтованої розробки ПС.

У результаті вивчення даного курсу студент повинен **вміти**: впроваджувати базовий аналіз предметної області, використовувати об'єктно-орієнтовані принципи та підходи; використовувати відомі шаблони проектування; будувати моделі програмних рішень, що розробляються.

## 2. Тематичний план навчальної дисципліни.

### Розділ 1. Базові принципи ООП та особливості основних мовних конструкцій.

**Тема 1.1.** «Дисципліна програмування: Ретроспектива та теперішній стан». Ретроспектива становлення основних парадигм та мов програмування: Виникнення дисципліни програмування. 1936 р.–1947 р. Мова вищого рівня. 1953 - 1956 рр. Приклади програм FORTRAN I на картах кодування Початок функціонального програмування. 1958р. Початок об'єктно-орієнтованого програмування. 1967р. Початок структурного програмування. 1968р. Поява об'єктно-орієнтованих мов. 1970 – 1995 рр. Моделі життєвого циклу. 1970 – 1995 рр. Пост об'єктно-орієнтоване програмування 2001 – 2008 рр. Сучасний стан.

**Тема 1.2.** «Базові принципи ООП: Принцип «абстракції». Вступ до UML». Базові принципи ООП-парадигми. Принцип абстракції (Abstraction). Предметна область (problem domain). Основні фази розробки програмних систем (ПС) та місце курсу ООП серед них. UML: типи моделей. Абстракція: Клас. Внутрішня структура класу. Зв'язок між класами в нотації UML. Поняття об'єкту (UML). Приклади java/UML.

**Тема 1.3.** «Базові принципи ООП: Принцип «інкапсуляції». Внутрішня структура класу». Принцип інкапсуляції (Encapsulation). Контроль доступу (Access Control). Види інкапсуляції. Основні типи зв'язків між класами в UML. Інкапсуляція приватного рівня. Інкапсуляція з трансформацією даних. Інкапсуляція через внутрішню абстракцію. Інкапсуляція пакетного рівня. Інкапсуляція при запиті даних. Інкапсуляція примітивних типів даних. Приклади java/UML.

**Тема 1.4.** «Особливості модифікаторів *static* та *final*. Специфіка використання конструкцій на їх основі». Статичний модифікатор (*static*). Особливості базових елементів класу із *static*-модифікатором. Статичні методи-фабрики як альтернатива конструктору для створення об'єктів класу. Фінальний модифікатор (*final*). Особливості використання *final*-модифікатора із конструкція мови java. Константи: переваги та недоліки. Перерахування (*enumeration*): властивості та специфіка використання. Утилітарний клас (*util class*): структура класу та особливості використання. Вкладений статичний клас (*nested static class*): особливості використання. Шаблон проектування “*Singleton*”: структура та призначення шаблону, базові різновиди та особливості використання. Приклади java/UML.

**Тема 1.5.** «Структура методу. Особливості використання конструктивних блоків». Метод. Загальна структура визначення (method definition). Мета-модель методу (method metamodel). Конструктивні елементи сигнатури методу (method's signature). Внутрішня структура «тіла» методу (method's body). Типи внутрішніх блоків метода. Структурний підхід

реалізації метода. Проблема неявного виходу із блоку. Вирішення проблеми неявного виходу із блоку. Логічні секції всередині методу. Трансформація методу у клас. Приклади java/UML.

## **Розділ 2. Колекція принципів SOLID та їх використання в об'єктно-орієнтованій розробці ПС.**

**Тема 2.1.** «Базові принципи об'єктно-орієнтованої розробки - SOLID. Принцип єдиної відповідальності». Ознаки проблемного дизайну на рівні класів ПС. Базові принципи ОО-дизайну. Відповідальність (responsibility): принцип єдиної відповідальності класу (Single Responsibility Principle). Приклад дизайну ПС із порушення SPR та її ре-дизайн із дотриманням SPR. Приклади java/UML.

**Тема 2.2.** «Базові принципи ООП: принцип «успадкування». SOLID: принцип «відкритої-закритості». Базові принципи ООП-парадигми: успадкування. Абстрактний Клас та Інтерфейс. Переваги та недоліки успадкування. Базові принципи ОО-дизайну: OCP - The Open/Closed Principle. Приклад дизайну ПС із порушення OCP та її ре-дизайн із дотриманням OCP. Приклади java/UML.

**Тема 2.3.** «Базові принципи ООП: принцип «поліморфізму». SOLID: принцип заміщення Б. Лісков». Базові принципи ООП-парадигми: поліморфізм. Принцип поліморфізму. Поліморфна поведінка. Поліморфізм на базі анонімного класу (Anonymous class). Базові принципи ОО-дизайну: LSP - The Liskov Substitution Principle. Приклад дизайну ПС із порушення LSP. Приклади java/UML.

## **Розділ 3. Особливості використання просунутих конструкцій в ООП.**

**Тема 3.1.** «Принцип незмінності класу - class immutability. Використання шаблону проектування «Builder» для побудови складних об'єктів». Ознаки мінливості (mutability) класу. Принцип незмінності (immutability): Структурні обмеження класу, Основні переваги незмінності класу. Змінний атрибут в незмінному класі. Захисна копія (defensive copy) змінних атрибутів класу. Змінюваність колекцій (collection mutability). Побудова складних об'єктів: шаблон проектування Builder («Будівельник»). Приклади java/UML.

**Тема 3.2.** «Контракт hashCode - equals. Загальний принцип роботи хеш-базованих колекцій». Основні характеристики примітивних типів Java. Еквівалентність змінних примітивних типів. Особливість змінних об'єктних типів Java. Еквівалентність об'єктів: оператор (equal To) '==' та метод 'equals()'. Перевизначення (Override) методу 'equals()' та його генерація в IDE. Хеш-функція. Колізія хеш-функції. Хеш-код об'єкта та метод 'hashCode()'. Загальний принцип роботи java.util.HashMap. Проблема «мутації» внутрішнього стану об'єкту «ключа» у HashMap. Приклади java/UML.

### 3. Структура навчальної дисципліни

Назви розділів і тем	Кількість годин					
	денна форма					
	усього	у тому числі				
л		п	лаб.	інд.	с. р.	
1	2	3	4	5	6	7
<b>Розділ 1. Базові принципи ООП та особливості основних мовних конструкцій</b>						
Тема 1.1. «Дисципліна програмування: Ретроспектива та теперішній стан»	8	2		2		4
Тема 1.2. «Базові принципи ООП: Принцип «абстракції». Вступ до UML».	11	3		4		4
Тема 1.3. «Базові принципи ООП: Принцип «інкапсуляції»	14	4		4		6
Тема 1.4. «Особливості модифікаторів static та final. Специфіка використання конструкцій на їх основі»	14	4		4		6
Тема 1.5. «Структура методу. Особливості використання конструктивних блоків»	10	2		2		6
<b>Разом за розділом 1</b>	<b>57</b>	<b>15</b>		<b>16</b>		<b>26</b>
<b>Розділ 2. Колекція принципів SOLID та їх використання в об'єктно-орієнтованій розробці ПС</b>						
Тема 2.1. «Базові принципи об'єктно-орієнтованої розробки - SOLID. Принцип єдиної відповідальності»	10	2		2		6
Тема 2.2. «Базові принципи ООП: принцип «успадкування». SOLID: принцип «відкритої-закритості»	12	4		4		4
Тема 2.3. «Базові принципи ООП: принцип «поліморфізму». SOLID: принцип заміщення Б. Лісков	10	4		2		4
<b>Разом за розділом 2</b>	<b>32</b>	<b>10</b>		<b>8</b>		<b>14</b>
<b>Розділ 3. Особливості використання просунутих конструкцій в ООП</b>						
Тема 3.1. «Принцип незмінності класу - class immutability. Використання шаблону проектування «Builder» для побудови складних об'єктів»	12	3		4		5
Тема 3.2. «Контракт hashCode - equals. Загальний принцип роботи хеш-базованих колекцій»	14	4		4		6
Підготовка до контрольної роботи	5					5
<b>Разом за розділом 3</b>	<b>31</b>	<b>7</b>		<b>8</b>		<b>16</b>
<b>Усього годин</b>	<b>120</b>	<b>32</b>		<b>32</b>		<b>56</b>

#### 4. Теми семінарських (практичних, лабораторних) занять

№ з/п	Назва теми	Кількість годин
<b>Розділ 1. Базові принципи ООП та особливості основних мовних конструкцій</b>		
1	Базові поняття та основні елементи інтегрованого середовища розробки ПЗ.	2
2	Використання компоненту відлагодження (debugger) в процесі розробки ПЗ.	4
3	Вивчення основних елементів мови UML та їх використання в процесі розробки ПЗ.	4
4	Застосування принципів абстракції та інкапсуляції в процесі розробки ПЗ. Робота із текстовими файлами.	4
5	Огляд конструктивних блоків методу та специфіка їх використання. Особливості роботи із конструкціями enumeration, utility-class. Використання шаблону проектування «Одинак» (Singleton).	2
Разом за розділом 1		16
<b>Розділ 2. Колекція принципів SOLID та їх використання в об'єктно-орієнтованій розробці ПЗ</b>		
6	Вивчення принципу «єдиної відповідальності» із колекції принципів SOLID. Розділення відповідальності при реалізації конкурентних потоків (concurrency/parallelism).	2
7	Застосування принципів із колекції SOLID в процесі розробки ПЗ. Наслідування та генералізовані класи (generic class).	4
8	Вивчення принципу «заміщення Б.Лісков» із колекції принципів SOLID. Огляд нотації лямбда-виразу Java 8.	2
Разом за розділом 2		8
<b>Розділ 3. Особливості використання просунутих конструкцій в ООП</b>		
9	Вивчення принципу незмінності класу та шаблонів проектування «Будівельник» (Builder) та «Спеціальний випадок» (Null-Object) Java 8.	4
10	Використання контракту hashCode – equals у хеш-базованих колекціях. Огляд Stream-API для роботи із колекціями. Java 8.	4
Разом за розділом 3		8
<b>Усього годин</b>		<b>32</b>

#### 5. Завдання для самостійної роботи

№ з/п	Назва теми	Кількість годин
<b>Розділ 1. Базові принципи ООП та особливості основних мовних конструкцій</b>		
1	Основні елементи інтегрованого середовища розробки (IDE). Режим відлагодження. Автоматизований рефакторинг вихідного коду.	4
2	Виділення основних абстракцій предметної області (PrO). Створення моделі класів UML.	4
3	Інкапсуляція структурних елементів класу, інкапсуляція класу.	6
4	Константи, перерахування, класи-утиліти, види реалізацій шаблону «Одинак» (Singleton)	6
5	Перетворення строк, читання та запис у файл.	6
Разом за розділом 1		26

№ з/п	Назва теми	Кількість годин
<b>Розділ 2. Колекція принципів SOLID та їх використання в об'єктно-орієнтованій розробці ПС</b>		
6	Принцип єдиної відповідальності: визначення завдання (Task) та визначення виконавця (Thread-Executor). Інтерфейси Runnable, Callable, клас Thread.	6
7	Наслідування у генералізованих класах (Generic class).	4
8	Java 8 складові лямбда-виразів: Consumer, Supplier, Function, Predicate.	4
Разом за розділом 2		14
<b>Розділ 3. Особливості використання просунутих конструкцій в ООП</b>		
9	Створення складних об'єктів із шаблоном «Будувельник» (Builder) та обробка виключних ситуацій за допомогою механізму Java 8 – Optional.	5
10	Маніпуляція колекціями із Java 8 Stream API.	6
11	Підготовка до контрольної роботи	5
Разом за розділом 3		16
<b>Усього годин</b>		<b>56</b>

## 6. Індивідуальні завдання

Контрольна робота.

Курсова робота.

## 7. Методи контролю

Протягом навчального семестру проводиться поточний контроль знань, який складається з виконання 3х лабораторних робіт, контрольної роботи та курсової роботи. Загальна сума балів, яку студент може набрати, складає – 60 балів.

Максимальна оцінка за лабораторну роботу – 7 балів.

Максимальна оцінка за контрольну роботу – 10 балів.

Максимальна оцінка за курсову роботу – 30 балів.

Підсумковий контроль – екзамен у письмовій формі.

Допуск до складання екзамену студент отримує, якщо він виконав усі лабораторні роботи, склав контрольну роботу та захистив курсову роботу з сумарною оцінкою не менше 50 балів. В іншому випадку студент не допускається до складання екзамену.

Екзамен складається з трьох питань (два теоретичних та одне практичне).

Максимальна кількість балів за одне теоретичне питання – 15 балів.

Максимальна кількість балів за одне практичне питання – 10 балів.

Максимальна кількість балів за екзамен – 40 балів.



### 8. Схема нарахування балів

Поточний контроль та самостійна робота			Контрольна робота	Курсова робота	Разом	Екзамен	Разом
ЛР 1 (Т 1.3)	ЛР2 (Т 2.2)	ЛР 3 (Т 3.1)	10	30	60	40	100
6	7	7					

#### Критерії поточної оцінки знань студентів (контрольна робота, крок оцінювання 2 бала)

Кількість балів	Критерії оцінки
2	Студент демонструє фрагментарні знання при незначному загальному їх обсязі (менше половини навчального матеріалу).
4	Студент демонструє, значну (більше половини) частину навчального матеріалу може відтворити репродуктивно; може викладача виконати просте навчальне завдання; має елементарні, нестійкі навички необхідні для виконання завдань.
6	Студент знайомий з основними поняттями навчального матеріалу; може самостійно відтворити значну частину навчального матеріалу і робити певні узагальнення; вміє виконати просте навчальне завдання.
8	Студент демонструє вивчений матеріал у стандартних ситуаціях; пояснює основні процеси, що відбуваються під час роботи інформаційної системи та наводить власні приклади на підтвердження деяких тверджень; вміє виконувати навчальні завдання.
10	Студент демонструє стійкі системні знання та продуктивно їх використовує; вміє вільно використовувати нові інформаційні технології для поповнення власних знань та розв'язування задач; має стійкі навички управління інформаційною системою у нестандартних ситуаціях.

#### Критерії поточної оцінки знань студентів (лабораторна робота, крок оцінювання 2 бала)

Кількість балів	Критерії оцінки
1	Студент має фрагментарні знання при незначному загальному їх обсязі (менше половини навчального матеріалу) при відсутності сформованих умінь та навичок.
3	Студент має початковий рівень знань, значну (більше половини) частину навчального матеріалу може відтворити репродуктивно; може з допомогою викладача виконати просте навчальне завдання; має елементарні, нестійкі навички необхідні для виконання завдання.
5	Студент вміє аналізувати навчальний матеріал, в цілому самостійно застосовувати його на практиці; контролювати власну діяльність; самостійно визначити спосіб розв'язування навчальної задачі.

Кількість балів	Критерії оцінки
6	Студент вміє застосовувати вивчений матеріал у стандартних ситуаціях; може пояснити основні процеси, що відбуваються під час роботи інформаційної системи та наводити власні приклади на підтвердження деяких тверджень; вміє виконувати навчальні завдання.
7	Студент має стійкі системні знання та продуктивно їх використовує; вміє вільно використовувати нові інформаційні технології для поповнення власних знань та розв'язування задач; має стійкі навички управління інформаційною системою у нестандартних ситуаціях.

**Критерії поточної оцінки знань студентів  
(курсова робота, крок оцінювання 6 балів)**

Кількість балів	Критерії оцінки
6	Студент розпізнає окремі об'єкти, явища і факти предметної галузі; знає і використовує базові конструкції об'єктно-орієнтованої мови програмування.
12	Студент має фрагментарні знання при незначному загальному їх обсязі (менше половини навчального матеріалу) за відсутності сформованих умінь та навичок.
18	Студент має рівень знань вищий, ніж початковий; може відтворити значну частину програмного коду з використанням базових принципів об'єктно-орієнтованого програмування; має стійкі навички виконання елементарних технологічних застосувань та їх опрацювання.
24	Студент вільно володіє навчальним матеріалом, застосовує знання при розробці програмних додатків; вміє використовувати просунуті конструкції об'єктно-орієнтованої мови програмування; самостійно виконує передбачене практичне завдання; самостійно знаходить і виправляє допущені помилки; може аргументовано обрати необхідний інструментарій для виконання завдання.
30	Студент має стійкі системні знання та продуктивно їх використовує; уміє вільно застосовувати вивчені принципи об'єктно-орієнтованої розробки та просунуті конструкції цільової мови програмування. Вміє сформулювати та реалізувати рішення проблеми, що належить цільовій предметній області.

**Критерії підсумкової оцінки знань студентів  
(одне теоретичне питання екзаменаційного білету, крок оцінювання 2 бали)**

Кількість балів	Критерії оцінки
3	Студент розпізнає окремі об'єкти, явища і факти предметної галузі; знає і виконує базові технологічні застосування.
6	Студент має фрагментарні знання при незначному загальному їх обсязі (менше половини навчального матеріалу) за відсутності сформованих умінь та навичок.
9	Студент має рівень знань вищий, ніж початковий; може з допомогою викладача відтворити значну частину навчального матеріалу з елементами логічних зв'язків; має стійкі навички виконання елементарних технологічних застосувань та їх опрацювання.
12	Студент вільно володіє навчальним матеріалом, застосовує знання на практиці; вміє узагальнювати і систематизувати навчальну інформацію; самостійно виконує передбачені програмою навчальні завдання; самостійно знаходить і виправляє допущені помилки; може аргументовано обрати раціональний спосіб виконання навчального завдання.

Кількість балів	Критерії оцінки
15	Студент має стійкі системні знання та продуктивно їх використовує, стійкі навички керування інформаційною системою в нестандартних ситуаціях; уміє вільно використовувати нові інформаційні технології для поповнення власних знань та розв'язування задач.

**Критерії підсумкової оцінки знань студентів  
(одне практичне питання екзаменаційного білету, крок оцінювання 2 бали)**

Кількість балів	Критерії оцінки
2	Студент розпізнає окремі об'єкти, явища і факти предметної галузі; знає і використовує базові конструкції об'єктно-орієнтованої мови програмування.
4	Студент має фрагментарні знання при незначному загальному їх обсязі (менше половини навчального матеріалу) за відсутності сформованих умінь та навичок.
6	Студент має рівень знань вищий, ніж початковий; може відтворити значну частину програмного коду з використанням базових принципів об'єктно-орієнтованого програмування; має стійкі навички виконання елементарних технологічних застосувань та їх опрацювання.
8	Студент вільно володіє навчальним матеріалом, застосовує знання при розробці програмних додатків; вміє використовувати просунуті конструкції об'єктно-орієнтованої мови програмування; самостійно виконує передбачене практичне завдання; самостійно знаходить і виправляє допущені помилки; може аргументовано обрати необхідний інструментарій для виконання завдання.
10	Студент має стійкі системні знання та продуктивно їх використовує; уміє вільно застосовувати вивчені принципи об'єктно-орієнтованої розробки та просунуті конструкції цільової мови програмування.

**Шкала оцінювання**

Сума балів за всі види навчальної діяльності протягом семестру	Оцінка для чотирирівневої шкали оцінювання
90-100	відмінно
70-89	добре
50-69	задовільно
1-49	незадовільно

## 9. Рекомендована література

### Основна література

1. Sharan, Kishori. Beginning Java 8 Fundamentals. Language Syntax, Arrays, Data Types, Objects, and Regular Expressions. – Apress, 2014. – 789 с.
2. Matt Weisfeld. The Object-Oriented Thought Process (4th Edition) - Addison-Wesely, 2013. – 306 с.
3. Creig Larman. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development. - Addison-Wesely, 2004. – 627с.
4. Бублик В.В. Б90 Об'єктно-орієнтоване програмування: [Підручник] / В.В. Бублик. – К.: ІТкнига, 2015. – 624 с. // [http://itknyga.com.ua/docs/OOP\\_final.pdf](http://itknyga.com.ua/docs/OOP_final.pdf)
5. Т. Бадд. Объектно-ориентированное программирование // [http://khizha.dp.ua/library/Timothy\\_Budd\\_-\\_Introduction\\_to\\_OOP\\_\(ru\).pdf](http://khizha.dp.ua/library/Timothy_Budd_-_Introduction_to_OOP_(ru).pdf)

### Допоміжна література

1. Martin Fowler. UML Distilled. A Brief Guide to the Standard Object Modeling Language (3rd Edition). - Addison-Wesely, 2004. – 175с.
2. Жуковський С.С., Вакалюк Т.А. Об'єктно-орієнтоване програмування мовою С++. Навчально-методичний посібник для студентів напряму 6.040302 Інформатика. – Житомир: Вид-во ЖДУ, 2016. – 100 с.  
//[http://eprints.zu.edu.ua/22367/1/%D0%BF%D0%BE%D1%81%D1%96%D0%B1%D0%BD%D0%B8%D0%BA\\_%D0%A1\\_\\_OOP1\\_%D1%81%D0%B0%D0%B9%D1%82.PDF](http://eprints.zu.edu.ua/22367/1/%D0%BF%D0%BE%D1%81%D1%96%D0%B1%D0%BD%D0%B8%D0%BA_%D0%A1__OOP1_%D1%81%D0%B0%D0%B9%D1%82.PDF)

### Посилання на інформаційні ресурси в Інтернеті, відео-лекції, інше методичне забезпечення

1. Oracle Java Documentation. The Java Tutorials  
// <https://docs.oracle.com/javase/tutorial/java/index.html>
2. Object Oriented Programming – Java OOPs Concepts with Examples  
// <https://www.edureka.co/blog/object-oriented-programming>