

Міністерство освіти і науки України  
Харківський національний університет імені В.Н. Каразіна  
Кафедра штучного інтелекту та програмного забезпечення

“ЗАТВЕРДЖУЮ”

В.о. декана факультету комп'ютерних наук

Євгенія КОЛОВАНОВА

“ 30 ” червня 2023 р.



## РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

### Об'єктно-орієнтоване програмування

рівень вищої освіти	перший (бакалаврський) рівень
галузь знань	12 Інформаційні технології
спеціальність	122 Комп'ютерні науки
освітня програма	Комп'ютерні науки
вид дисципліни	
факультет	обов'язкова

Програму рекомендовано до затвердження Вченою радою факультету комп'ютерних наук  
«29» червня 2023 року, протокол № 14

РОЗРОБНИК ПРОГРАМИ:

доцент кафедри штучного інтелекту та програмного забезпечення **Поклонський Євген  
Васильович**

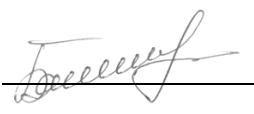
Програму схвалено на засіданні кафедри штучного інтелекту та програмного забезпечення  
Протокол від «09» червня 2023 року, протокол № 10

Завідувач кафедри штучного інтелекту та програмного забезпечення

  
Володимир КУКЛІН

Програму погоджено з гарантом освітньої програми «Комп'ютерні науки»

Гарант освітньої програми «Комп'ютерні науки»

  
Сергій БОГУЧАРСЬКИЙ

Програму погоджено науково-методичною комісією факультету комп'ютерних наук  
Протокол від «21» червня 2023 року, протокол № 12

Голова науково-методичної комісії факультету комп'ютерних наук

  
Лариса ВАСИЛЬОВА

## ВСТУП

Програма навчальної дисципліни «Об'єктно-орієнтоване програмування» складена відповідно до освітньо-професійної програми підготовки першого (бакалаврського) рівня вищої освіти за спеціальністю: 122 Комп'ютерні науки.

### 1. Опис навчальної дисципліни

1.1. Метою вивчення курсу «Об'єктно-орієнтоване програмування» є навчити студентів самостійно розробляти невеликі програмні продукти за допомогою об'єктно-орієнтованого підходу.

1.2. Основними завданнями вивчення дисципліни є:

- навчитися використовувати об'єктно-орієнтований підхід при аналізі явищ та систем;
- досягти рівня вільного володіння переважною більшістю синтаксичних конструкцій мови Java;
- набути практичних навичок створення та відлагодження програмного коду з використанням мови Java та супутніх програмних засобів (компілятор, IDE тощо);
- познайомитися на практиці з найчастіше використовуваними пакетами платформи J2SE та отримати уявлення про наявність інших пакетів;
- познайомитися на практиці із методами аналізу, проектування ПЗ та методологіями розробки програмних систем.

В ході вивчення дисципліни у студента повинні формуватися наступні компетентності.  
*Інтегральна компетентність.*

Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми у галузі комп'ютерних наук або у процесі навчання, що передбачає застосування теорій та методів інформаційних технологій і характеризується комплексністю та невизначеністю умов.

*Загальні компетентності (ЗК).*

- ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.
- ЗК2. Здатність застосовувати знання у практичних ситуаціях.
- ЗК3. Знання та розуміння предметної області та розуміння професійної діяльності.
- ЗК6. Здатність вчитися й оволодівати сучасними знаннями.
- ЗК7. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.
- ЗК8. Здатність генерувати нові ідеї (креативність).
- ЗК9. Здатність працювати в команді.
- ЗК10. Здатність бути критичним і самокритичним.
- ЗК11. Здатність приймати обґрунтовані рішення.
- ЗК12. Здатність оцінювати та забезпечувати якість виконуваних робіт.

*Спеціальні (фахові, предметні) компетентності (СК)*

- СК3. 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.

- СК7. Здатність застосовувати теоретичні та практичні основи методології та технології моделювання для дослідження характеристик і поведінки складних об'єктів і систем, проводити обчислювальні експерименти з обробкою й аналізом результатів.
- СК8. 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.

### 1.3. Кількість кредитів

2 семестр – 5 кредитів

3 семестр – 4 кредити

### 1.4. Загальна кількість годин – 270

1.5. Характеристика навчальної дисципліни	
Нормативна / за вибором	
денна форма навчання	денна форма навчання
Рік підготовки	
1-й,	-й
Семестр	
2-й	-й
Лекції	
32 год.	_____ год.
Практичні, семінарські заняття	
_____ год.	_____ год.
Лабораторні заняття	
32 год.	_____ год.
Самостійна робота	
86 год.	_____ год.
В т.ч. індивідуальні завдання	
5 год. (контр. робота)	_____ год.
Рік підготовки	
2-й,	-й
Семестр	
3-й	-й
Лекції	
32 год.	_____ год.
Практичні, семінарські заняття	
_____ год.	_____ год.
Лабораторні заняття	
32 год.	_____ год.
Самостійна робота	
56 год.	_____ год.
В т.ч. індивідуальні завдання	
15 год.	_____ год.

## 1.6. Заплановані результати навчання

### **знати:**

- сучасні технологій та інструментальні засоби розробки програмних систем, уміння їх застосовувати на всіх етапах життєвого циклу;
- розуміння основних індустріальних мов програмування, інформаційних систем і WEB.

### **вміти:**

- здатність до об'єктно-орієнтованого мислення, знання об'єктно-орієнтованих мов програмування та уміння застосовувати об'єктно-орієнтований підхід під час проектуванні складних програмних систем.

В результаті вивчення дисципліни у студента повинні формуватися наступні програмні результати навчання (ПРН).

- ПРН1. Застосовувати знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу, обробки та синтезу інформації в предметній області комп'ютерних наук
- ПРН9. 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.
- ПРН15. Застосовувати знання методології та CASE-засобів проектування складних систем, методів структурного аналізу систем, об'єктно-орієнтованої методології проектування при розробці і дослідженні функціональних моделей організаційно-економічних і виробничо-технічних систем.

## 2. Тематичний план навчальної дисципліни

### **Розділ 1. Знайомство з мовою Java. Інкапсуляція.**

#### ***Тема 1. Огляд курсу***

Парадигми програмування. Особливості об'єктно-орієнтованого програмування. Етапи розробки проекту. Огляд курсу.

#### ***Тема 2. Основні поняття мови Java.***

Історія ООП та Java. Особливості мови Java. Структура програми. Угоди про використання імен. Види Java-додатків. Засоби розробки Java-додатків.

#### ***Тема 3. Основні поняття мови Java.***

Лексичні основи мови (константи, змінні, ключові слова, операції). Типи даних: примітивні і об'єктні. Перетворення типів. Оператори мови. Типи обчислювальних процесів. Умовний оператор, умовні вирази, оператор вибору варіанту. Циклічні обчислювальні процеси (оператори циклів while, for, do-while). Особливості використання операторів циклу. Оператори break, continue

#### ***Тема 4. Структурні типи даних.***

Робота з одновимірними і багатовимірними масивами. Робота зі строками.

#### ***Тема 5. Об'єктна модель Java.***

Концепція об'єктно-орієнтованого програмування. JVM, JDK, CLASSPATH, компіляція та завантаження класів. Класи і об'єкти (поля, методи та конструктори.).

Інкапсуляція; приклади. Рівні доступу полів. Пакети. Методи і змінні класів (приховування змінних, this, статичні методи і дані). Поєднання методів. Створення і руйнування об'єктів (конструктори, збірка "сміття", finalize ()). Статичні поля.

## **Розділ 2. Спадкування та поліморфізм**

### ***Тема 6. Повторне використання класів.***

Композиція і успадкування. Побудова похідних класів. Ієрархічні структури класів. Структура об'єкту-спадкоємця. Порядок виклику конструкторів, розрішення конфліктів імен. Приклади.

### ***Тема 7. Об'єктна модель в Java.***

Абстрактні класи. Клас Object. Динамічне зв'язування (поліморфізм). Приклади поліморфного поведінку. Посилання типу суперкласу. Пряме та зворотне приведення типів. Раннє та пізнє зв'язування.

### ***Тема 8. Об'єктна модель в Java.***

Проблема множинного наслідування. Інтерфейси. Ієрархія інтерфейсів. Порівняння абстрактних класів і інтерфейсів.

## **Розділ 3. Обробка виключень, внутрішні класи і огляд стандартних пакетів.**

### ***Тема 9. Внутрішні класи.***

Внутрішні класи в методі і контексті. Анонімний внутрішній клас. Зв'язок із зовнішнім класом.

### ***Тема 10. Обробка виключень.***

Загальні відомості про обробку виключень. Класи виключень. Механізм обробки виключень (try, catch, throw, throws; блок finally). Кілька розділів catch. Вкладені оператори try.

### ***Тема 11. Пакет java.lang.***

Оболонки примітивних типів. Клас Runtime. Клас System. Клас Class.

### ***Тема 12. Сервісні класи пакету java.util.***

Розбір строк (StringTokenizer). Класи для роботи з датами. Робота з часом.

### ***Тема 13. Пакети і утиліти.***

Пакети java.math, java.text. Документація і утиліта java.doc. Архіви та утиліта jar.

## **Розділ 4. Бібліотека колекцій**

### ***Тема 14. Структура колекцій (пакет java.util). Послідовні колекції.***

Поняття колекції. Інтерфейс List і його реалізації (ArrayList, LinkedList). Інтерфейс Set и его реализации (HashSet, TreeSet). 1. Доступ до колекції через ітератор. Порівняння послідовних колекцій.

### ***Тема 15. Структура колекцій (пакет java.util). Асоціативні Колекції.***

Ілюстрація використання асоціативних колекцій. Бінарні дерева та хешування. Приклади. Карти відображень. Компаратори. Клас Arrays.

## **Розділ 5. Потоки.**

### **Тема 16. Програмування множинних потоків. Огляд курсу**

Багато поточне програмування. Модель потоків в Java. Головний потік, створення потоку. Створення множинних потоків, пріоритети потоків. Міжпоточні зв'язки.

### **Тема 17. Робота з потоками.**

Синхронізація. Внутрішні блокування. Живучість (Liveness): взаємне блокування (Deadlock), голодування (starvation), активна локіровка (livelock). Захищені блокування (guarded blocks). Незмінні об'єкти (immutable objects).

### **Тема 18. Введення / виведення в Java.**

Потоки введення / виведення. Клас File. Байтові потоки. Символьні потоки. Робота з файлами з довільним доступом. Серіалізація.

## **Розділ 6. AWT, Swing. Введення в проектування**

### **Тема 19. Основи проектування.**

Мова UML та case засоби. Діаграми UML. Варіанти використання і дійові особи.

### **Тема 20. Основи об'єктно-орієнтованого проектування.**

Взаємодія об'єктів. Класи і пакети. Атрибути, операції,

### **Тема 21. Генерація програмного коду.**

Підготовка до генерації програмного коду. Генерація програмного коду Java. Зворотне проектування програмного коду Java.

### **Тема 22. Основи AWT: робота з вікнами.**

Основи віконної графіки. Робота з фреймової вікнами. Малювання простих об'єктів.

### **Тема 23. 8 Обробка подій: модель делегування подій.**

Події та їх джерела. Інтерфейси слухачів. Класи-адаптери.

### **Тема 24. Основи AWT: робота з текстом.**

Робота з кольором. Робота зі шрифтом. Клас FontMetrics. Основи AWT: елементи управління ( Кнопки. Списки. Панелі меню), використання менеджерів компоновання ( BorderLayout, FlowLayout, GridLayout, CardLayout).

### **Тема 25. Аплети.**

Загальні відомості про аплетах. Методи аpletів, вивід графіки. Питання безпеки.

### **Тема 26. Робота з бібліотекою Swing.**

Клас JFrame. Створення стандартних елементів управління Swing (кнопки і їх групи, радіокнопки, прапорці, списки, що випадають і ін.). Бібліотека Swing: каталог компонентів: текстові поля, бордюри, панелі скролінгов, списки що випадають та ін.

### **Тема 27. Робота з бібліотекою Swing.**

Каталог компонентів: меню, спливаючі меню, вікна діалогу, файлові діалоги. Елементи GUI і пов'язані з ними події. Модель проектування MVC. Порівняння побудови програми на основі MVC та без нього на прикладі.

### **Тема 28. Огляд Java FX.**

Огляд Java FX (технологія Java Beans, BDK, інтроспекція).

## 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>2 семестр</b>						
<b>Розділ 1. Знайомство з мовою Java. Інкапсуляція.</b>						
Тема 1. Огляд курсу.	10	2		2		6
Тема 2. Основні поняття мови Java.	10	2		2		6
Тема 3. Основні поняття мови Java.	11	2		2	1	6
Тема 4. Структурні типи даних.	10	2		2		6
Тема 5. Об'єктна модель Java.	11	2		2	1	6
<b>Разом за розділом 1</b>	<b>52</b>	<b>10</b>		<b>10</b>	<b>2</b>	<b>30</b>
<b>Розділ 2. Спадкування та поліморфізм</b>						
Тема 6. Повторне використання класів.	9	2		2		5
Тема 7. Об'єктна модель в Java.	9	2		2		5
Тема 8. Об'єктна модель в Java.	10	2		2	1	5
<b>Разом за розділом 2</b>	<b>28</b>	<b>6</b>		<b>6</b>	<b>1</b>	<b>15</b>
<b>Розділ 3. Обробка виключень, внутрішні класи і огляд стандартних пакетів.</b>						
Тема 9. Внутрішні класи.	9	2		2		5
Тема 10. Обробка виключень.	9	2		2		5
Тема 11. Пакет java.lang.	9	2		2		5
Тема 12. Сервісні класи пакету java.util.	9	2		2		5
Тема 13. Пакети і утиліти.	10	2		2	1	5
<b>Разом за розділом 3</b>	<b>46</b>	<b>10</b>		<b>10</b>	<b>1</b>	<b>25</b>
<b>Розділ 4. Бібліотека колекцій</b>						
Тема 14. Структура колекцій. Послідовні колекції.	9	2		2		5
Тема 15. Структура колекцій. Асоціативні Колекції.	15	4		4	1	6
<b>Разом за розділом 4</b>	<b>23</b>	<b>6</b>		<b>6</b>	<b>1</b>	<b>11</b>
<b>Усього годин</b>	<b>150</b>	<b>32</b>		<b>32</b>	<b>5</b>	<b>81</b>
<b>3 семестр</b>						
<b>Розділ 5. Потіки.</b>						
Тема 16. Програмування множинних потоків. Огляд курсу	8	2		2	2	2
Тема 17. Робота з потоками.	12	4		4	2	2
Тема 18. Введення / виведення в Java.	12	4		4	2	2
<b>Разом за розділом 5</b>	<b>32</b>	<b>10</b>		<b>10</b>	<b>6</b>	<b>6</b>
<b>Розділ 6. AWT, Swing. Введення в проектування</b>						
Тема 19. Основи проектування.	8	2		2	2	2
Тема 20. Основи об'єктно-орієнтованого проектування.	12	4		4	2	2
Тема 21. Генерація програмного коду.	8	2		2	2	2
Тема 22. Основи AWT: робота з вікнами.	8	2		2	2	2
Тема 23. Обробка подій: модель делегування подій.	8	2		2	2	2
Тема 24. Основи AWT: робота з текстом.	8	2		2	2	2
Тема 25. Аплети.	10	2		2	4	2
Тема 26. Робота з бібліотекою Swing.	8	2		2	2	2
Тема 27. Робота з бібліотекою Swing.	10	2		2	2	4
Тема 28. Обзор Java FX.	8	2		2	2	2
<b>Разом за розділом 6</b>	<b>88</b>	<b>22</b>		<b>22</b>	<b>22</b>	<b>22</b>
<b>Усього годин</b>	<b>120</b>	<b>32</b>		<b>32</b>	<b>28</b>	<b>28</b>



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

№ з/п	Назва теми	Кількість годин
<b>2 семестр</b>		
<b>Розділ 1. Інкапсуляція</b>		
1	Введення-виведення, цикли, масиви	4
2	Інкапсуляція. Масиви об'єктів.	6
<b>Розділ 2. Спадкування та поліморфізм</b>		
3	Відкрите спадкування. Абстрактні класи. Поліморфізм	2
4	Інтерфейси.	4
<b>Розділ 3. Обробка виключень, внутрішні класи та огляд 7 стандартних пакетів</b>		
5	Внутрішні класи	2
6	Обробка виключень.	2
7	Пакет java.lang. Пакет java.util.	6
<b>Розділ 4. Бібліотека колекцій</b>		
8	Послідовні колекції	2
9	Асоціативні колекції	4
	<b>Разом</b>	<b>32</b>
<b>3 семестр</b>		
<b>Розділ 5. Потoki.</b>		
1	Багатопотокове програмування. Робота з декількома потоками.	2
2	Потоки вводу. вивода	4
3	Серілізація.	4
<b>Розділ 6. AWT, Swing. Введення в проектування</b>		
4	Розробка програми з використанням AWT	6
5	Розробка програми з використанням Swing	8
6	Розробка аплету	8
	<b>Разом</b>	<b>32</b>

#### 5. Завдання для самостійної роботи

№ з/п	Види, зміст самостійної роботи	Кількість годин
<b>Розділ 1. Інкапсуляція</b>		
1	<b>Введення-виведення, цикли, масиви.</b> Самостійне опрацювання лекційного матеріалу Додатково прочитати у Хорстманна[1] (глава 3) або Шилдта[3] (глави 3-5) про змінні, типи даних, константи та розібрати приклади коду. Розібратись з вводом з клавіатури за допомогою класу Scanner. Контроль: опитування під час практичного заняття, опитування під час здачі лабораторної роботи № 1	6
2	Інкапсуляція. Самостійне опрацювання лекційного матеріалу. Хорстманна[1] (глава 4), Шилдт[3] (глава 6). Контроль: опитування під час практичного заняття, опитування під час здачі лабораторної роботи № 2	10
3	Розробка корисного класу вектор. Самостійне опрацювання лекційного матеріалу. Хорстманна[1], глава 4, Шилдт[3] (глава 7). Контроль: опитування під час практичного заняття.	16
<b>Розділ 2. Спадкування та поліморфізм</b>		
4	Відкрите спадкування. Самостійне опрацювання лекційного матеріалу. Хорстманн[1], глава 5, Шилдт[3] (глава 8). Контроль: опитування під час практичного заняття, опитування під час прийому лабораторної роботи № 3. Підготовка до контрольної роботи	5

5	Порівняння спадкування реалізації та композиції. Самостійне опрацювання лекційного матеріалу. Хорстманн[1], глава 5. Контроль: опитування під час практичного заняття, опитування під час прийому лабораторної роботи № 4	5
6	Поліморфізм. Інтерфейси. Самостійне опрацювання лекційного матеріалу. Хорстманн[1], глава 6 (інтерфейси), Шилдт[3] (глава 9). Контроль: опитування під час практичного заняття, Підготовка до контрольної роботи.	6
<b>Розділ 3. Обробка виключень, внутрішні класи та огляд стандартних пакетів</b>		
7	Внутрішні класи. Самостійне опрацювання лекційного матеріалу. Хорстманн[1], глава 6 (внутр.класи). Контроль: опитування під час практичного заняття, опитування під час прийому лабораторної роботи № 5	10
8	Обробка виключень. Самостійне опрацювання лекційного матеріалу. Хорстманн[1], глава 7, Шилдт[3] (глава 10). Контроль: опитування під час практичного заняття, опитування під час прийому лабораторної роботи № 6	10
9	Пакет java.lang. Пакет java.util. Самостійне опрацювання лекційного матеріалу. Шилдт[3] (глави 16, 18). Контроль: опитування під час практичного заняття, опитування під час прийому лабораторної роботи № 7	11
<b>Розділ 4. Бібліотека колекцій</b>		
10	Послідовні колекції Самостійне опрацювання лекційного матеріалу. Хорстманн[1], глава 9, Шилдт[3] (глави 17). Контроль: опитування під час практичного заняття, опитування під час прийому лабораторної роботи № 8	6
11	Асоціативні колекції. Самостійне опрацювання лекційного матеріалу. Контроль: опитування під час практичного заняття, опитування під час прийому лабораторної роботи № 9.	6
	<b>Разом</b>	<b>86</b>
<b>Розділ 5. Потоки.</b>		
1	Багатопотокове програмування. Робота з декількома потоками. Самостійне опрацювання лекційного матеріалу. Шилдт[3] (глава 11, 27). Контроль: опитування під час практичного заняття, опитування під час прийому лабораторної роботи № 1 . Підготовка до контрольної роботи	2
2	Потоки ввода.вивода. Робота з декількома потоками. Самостійне опрацювання лекційного матеріалу. Шилдт[3] (глава 19). Контроль: опитування під час практичного заняття	2
3	Серілізація. Самостійне опрацювання лекційного матеріалу. Контроль: опитування під час практичного заняття, опитування під час прийому лабораторної роботи № 2	2
<b>Розділ 6. AWT, Swing. Введення в проектування</b>		
4	Розробка програми з використанням AWT. Самостійне опрацювання лекційного матеріалу. Шилдт[3] (глави 23,24,25). Контроль: опитування під час практичного заняття, , опитування під час прийому лабораторних робіт № 3,4.	12
5	Розробка програми з використанням Swing. Самостійне опрацювання лекційного матеріалу. Шилдт[3] (глави 26,30). Контроль: опитування під час практичного заняття, , опитування під час прийому лабораторних робіт № 5,6.	12
6	Розробка аплету. Самостійне опрацювання лекційного матеріалу. Шилдт[3] (глави 22). Контроль: опитування під час практичного заняття, опитування під час прийому лабораторної роботи № 7.	11
	Курсова робота	15
	<b>Разом</b>	<b>56</b>

## 6. Індивідуальні завдання

Індивідуальна робота у 2 семестрі складається з виконання лабораторних робіт з використанням мови Java згідно методичним вказівкам.

Індивідуальна робота у 3 семестрі складається з виконання лабораторних робіт з використанням мови Java згідно методичним вказівкам та курсової роботи.

**Курсова робота.** Як наукове та навчально-дослідне завдання кожен студент має виконати курсову роботу. В ході роботи над проектом студент повинен показати розуміння основ ООП та основ ООП проектування, роботи з файлами. Студент повинен розробити програму, пояснювальну записку до неї, створити презентацію та захистити свою роботу перед комісією та студентами. Кожному студенту дається можливість самостійно вибрати тему для виконання. Студент повинен обговорити тему роботи з викладачем та затвердити її.

В якості теми роботи може бути, наприклад

- 1) термінал для роботи з «квазіБД» (реєстратура, бібліотека і т.п.) (зберігання даних в файлах)
- 2) гра (сапер, змійка, морський бій і т.п.)
- 3) тренажер (запам'ятовування іноземних слів, тестування з якогось предмету і т.п.)

## 7. Методи навчання

На досягнення освітніх цілей спрямовані такі методи навчання студентів:

– практичні (використовують для пізнання дійсності, формування навичок і вмінь, поглиблення знань. Під час їх застосування використовуються такі прийоми: планування виконання завдання, постановка завдання, оперативне стимулювання, контроль і регулювання, аналіз результатів, визначення причин недоліків);

– пояснювальне-ілюстративний (використовують для викладання й засвоєння нового навчального матеріалу, фактів, підходів, оцінок, висновків тощо);

– репродуктивний (для застосування студентами вивченого на основі зразка або правила, алгоритму, що відповідає інструкціям, правилам, в аналогічних до представленого зразка ситуаціях);

Як правило лекційні та практичні заняття проводяться аудиторне. В умовах дії карантину заняття проводяться відповідно до Наказу ректора Харківського національного університету імені В.Н. Каразіна (аудиторне або дистанційно за допомогою платформ Google Meet або Zoom).

## 8. Методи контролю

У другому семестрі передбачений захист лабораторних робіт та індивідуальних завдань перед викладачем з одночасним їх тестуванням. Залік в кінці семестру складається з тесту та балів за лабораторні та індивідуальні завдання.

У третьому семестрі передбачений захист лабораторних робіт перед викладачем, індивідуальної роботи, а також спільний проект, при виконанні якого прогрес контролює викладач. Результати роботи над проектом захищаються. Програмні продукти, розроблені відповідно темам курсових робіт, на CD диску та пояснювальні записки у вигляді паперових звітів зберігаються на кафедрі.

Екзамен в кінці семестру складається з тесту та практичного завдання.

### 9. Схема нарахування балів

Поточний контроль, самостійна робота, індивідуальні завдання 2 семестр											Залік ова робо та	Сума
ЛР 1	ЛР 2	ЛР 3	ЛР 4	ЛР 5	ЛР 6	ЛР 7	ЛР 8	ЛР 9	Контрольна робота, передбачена навчальним планом	Разом		
5	5	5	5	5	5	5	5	5	15	60	40	100

Поточний контроль, самостійна робота, індивідуальні завдання, 3 семестр										Екзамен	Сума
ЛР 1	ЛР 2	ЛР 3	ЛР 4	ЛР 5	ЛР 6	ЛР 7	Індивідуаль не завдання (курсний проект)	Контрольна робота, передбачена навчальним планом	Разом		
5	5	5	5	5	3	2	20	10	60	40	100

#### КРИТЕРІЇ ОЦІНКИ УСПІШНОСТІ РЕЗУЛЬТАТІВ НАВЧАННЯ З КУРСУ «Об'єктно-орієнтоване програмування» 2 семестр

<i>Форми навчальної діяльності</i>	<i>Балів за семестр</i>
1. Виконання практичних завдань	26
2. Виконання лабораторних робіт	24
3. Виконання контрольної роботи	10
4. Залік (письмова робота)	40
<b>Разом балів</b>	<b>100</b>

#### 3 семестр

<i>Форми навчальної діяльності</i>	<i>Бали за семестр</i>
1. Виконання практичних завдань	18
2. Виконання лабораторних робіт	12
3. Виконання контрольної роботи	10
4. Курсова робота	20
5. Екзамен (письмова робота)	40
<b>Разом балів</b>	<b>100</b>

#### Виконання практичних завдань

Протягом 2 семестру студент має виконати 13 практичних завдань за темами:

Теми практичних занять	Вид завдання	Кількість балів
Практичне заняття №1 IDE, примітивні типи даних.	практичне	2
Практичне заняття №2 Керуючі оператори, цикли.	практичне	2
Практичне заняття №3 Масиви, багатовимірні масиви.	практичне	2
Практичне заняття №4 Класи. Створення простого класу.	практичне	2
Практичне заняття №5 Наслідування	практичне	2
Практичне заняття №6 Абстрактні класи.	практичне	2
Практичне заняття №7 Інтерфейси.	практичне	2
Практичне заняття №8 Обробка виключень.	практичне	2

Практичне заняття №9 Внутрішні класи.	практичне	2
Практичне заняття №10 Класи-оболонки.	практичне	2
Практичне заняття №11 Узагальнені типи.	практичне	2
Практичне заняття №12 Колекції.	практичне	2
Практичне заняття №13 Відображення.	практичне	2
<b>Всього</b>		<b>26</b>

Протягом 3 семестру студент має виконати 13 практичних завдань за темами:

Теми практичних занять	Вид завдання	Кількість балів
Практичне заняття №1 Створення множинних потоків, пріоритети потоків. Міжпоточні зв'язки.	практичне	2
Практичне заняття №2 Синхронізація. Взаємні блокування.	практичне	2
Практичне заняття №3 Потоки введення/виведення.	практичне	2
Практичне заняття №4 Основи віконної графіки. Робота з фреймової вікнами.	практичне	2
Практичне заняття №5 Події та їх джерела. Інтерфейси слухачів. Класи-адаптери.	практичне	2
Практичне заняття №6 Основи AWT: елементи управління, використання менеджерів компоунування.	практичне	2
Практичне заняття №7 Бібліотека Swing.	практичне	2
Практичне заняття №8 Каталог компонентів: меню, спливаючі меню, вікна діалогу, файлові діалоги. Елементи GUI і пов'язані з ними події.	практичне	2
Практичне заняття №9 Модель проектування MVC. Порівняння побудови програми на основі MVC та без нього на прикладі.	практичне	2
<b>Всього</b>		<b>18</b>

### Критерії оцінювання практичних завдань

Перед виконанням студент вивчає вимоги завдання, відбувається обговорення, щоб завдання було зрозуміле кореткно. Кожен студент виконує завдання самостійно. Дозволяється сумісне обговорення, але програмний код кожен студент пише власноруч.

Захист практичного завдання складається з представлення результатів: готового програмного додатку та програмних кодів у відповідності до завдання.

### Виконання лабораторних робіт

Протягом 2 семестру студент має виконати 6 лабораторних робіт за темами:

Теми лабораторних робіт	Термін виконання	Кількість балів
Лабораторна робота №1 Двовимірний масив, сортування рядків.	2 тижні	4
Лабораторна робота №2 Клас вектор.	2 тижні	4
Лабораторна робота №3 Наслідування, поліморфізм (військова база).	2 тижні	4
Лабораторна робота №4 Обробка виключень, узагальнення типу (клас вектор).	2 тижні	4
Лабораторна робота №5 Власна реалізація інтерфейсів List, Set (на основі класу вектор). Реалізація інтерфейсу Iterator (внутрішній клас класу вектор).	2 тижні	4
Лабораторна робота №6 Реалізація інтерфейсу Map.	2 тижні	4
<b>Всього</b>		<b>24</b>

Протягом 3 семестру студент має виконати 3 лабораторні роботи за темами:

Теми лабораторних робіт	Термін виконання	Кількість балів
Лабораторна робота №1 Багатопотоковість.	2 тижні	4
Лабораторна робота №2 Введення/виведення даних. Робота з файлами.	2 тижні	4
Лабораторна робота №3 Графічний за стосунок з анімацією.	2 тижні	4
<b>Всього</b>		<b>12</b>

### Критерії оцінювання лабораторних робіт

Студент вивчає вимоги та технічне завдання, виконує роботу самостійно. Якщо програмний код робіт різних студентів має суттєву ступінь схожості, то ці роботи дискваліфікуються та не оцінюються.

Захист лабораторної роботи складається з двох етапів.

Перший етап – представлення результатів: готового програмного додатку та програмних кодів у відповідності до завдання лабораторної роботи.

Критерії оцінювання результатів роботи:

- робота була виконана у відповідності з технічним завданням за вказаний час – студент отримує 100% від максимальної кількості балів;
- робота була виконана у відповідності з технічним завданням с запізненням – студент отримує 50 % від максимальної кількості балів;
- технічне завдання виконано не повністю, а на  $x$  % – студент отримує  $x\%$  від балів, що мали б бути зараховані у відповідності до строків виконання.

Другий етап здачі лабораторної роботи – відповідь на контрольні питання. Кількість питань визначає викладач за результатами представлених результатів, але не менше ніж 2. Кожне контрольне питання оцінюється таким чином:

- повна розгорнута відповідь з прикладами та додатковим завданням, що було опрацьовано на самостійній роботі – кількість балів, що отримані на першому етапі збільшується до 25 % балів;
- повна, але не розгорнута відповідь без додаткового завдання – кількість балів не змінюється;
- неповна відповідь, або відповідь, що містить незначні та некритичні помилки чи суперечності – кількість балів, що було отримано на попередньому етапі зменшується на 25 %;
- відповідь, що містить критичні помилки, або відсутність відповіді – кількість балів, що було отримано на попередньому етапі зменшується на 50 %.

### Критерії оцінювання контрольних робіт

Протягом 2 семестру студент має виконати письмову контрольну роботу. Контрольна роботи виконуються в аудиторії, розрахована на 80 хвилин та оцінюються у 10 балів.

Контрольна робота складаються з практичних та теоретичних завдань. Кожне завдання має фіксовану максимальну кількість балів:

Список завдань	Тип завдання	Кількість балів
Завдання 1.	теоретичне	2
Завдання 2.	теоретичне	2
Завдання 3.	теоретичне	2
Завдання 4.	практичне	2
Завдання 5.	практичне	2
<b>Всього</b>		<b>10</b>

### Залік

Залік у 2 семестрі проводиться в письмовій формі, розрахований на 2 години, оцінюються у 40 балів.

Залікова робота складається з практичних та теоретичних завдань. Кожне завдання має фіксовану максимальну кількість балів:

Список завдань	Тип завдання	Кількість балів
Завдання 1.	теоретичне	2
Завдання 2.	теоретичне	2
Завдання 3.	теоретичне	2
Завдання 4.	теоретичне	2
Завдання 5.	теоретичне	2
Завдання 6.	теоретичне	2
Завдання 7.	теоретичне	2
Завдання 8.	практичне	13
Завдання 9.	практичне	13
<b>Всього</b>		<b>40</b>

### Шкала оцінювання

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

### Критерії оцінювання контрольних робіт 3 семестру

Теоретичні завдання оцінюються таким чином:

- 100% від запланованої кількості балів – у випадку вірної відповіді;
- 80% від запланованої кількості балів – у випадку незначної неточності у відповіді;
- 50% від запланованої кількості балів – у випадку декількох незначних неточностей у відповіді;
- 0 балів – у випадку значної неточності, помилки, або відсутності відповіді.

Практичні завдання оцінюються таким чином:

- 100% від запланованої кількості балів у випадку готового програмного додатку та програмних кодів у відповідності до завдання;
- 0 балів – у випадку відсутності діючого програмного кода.

### Критерії оцінювання курсового проекту 3 семестру

Курсовий проект оцінюється сумарно в 20 балів. Оцінка складається з таких елементів:

Критерій	Бали
Документація	5
Проектування	5
Тестування функціональних вимог	5
Якість презентації	3
Доклад та відповіді на запитання	2
<b>Всього</b>	<b>20</b>

### Екзамен

Екзамен проводиться в письмовій формі, розраховано на 3 години, оцінюються у 40 балів.

Екзаменаційна робота складається з практичних та теоретичних завдань. Кожне завдання має фіксовану максимальну кількість балів:

Список завдань	Тип завдання	Кількість балів
Завдання 1.	теоретичне	10
Завдання 2.	теоретичне	10
Завдання 3.	практичне	20
<b>Всього</b>		<b>40</b>

### Критерії оцінювання екзамену

Теоретичні завдання оцінюються таким чином:

- 100% від запланованої кількості балів – у випадку вірної відповіді;
- 80% від запланованої кількості балів – у випадку незначної неточності у відповіді;
- 50% від запланованої кількості балів – у випадку декількох незначних неточностей у відповіді;
- 0 балів – у випадку значної неточності, помилки, або відсутності відповіді.

Практичні завдання оцінюються таким чином:

- 100% від запланованої кількості балів у – випадку готового програмного додатку та програмних кодів у відповідності до завдання;
- 0 балів – у випадку відсутності діючого програмного коду.

### Шкала оцінювання

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



## 10. Рекомендована література

### Основна література

1. Horstmann C. S. Core Java Volume I – Fundamentals: 11th Edition, Pearson 2018, 928 p.
2. Horstmann C. S. Core Java, Volume II--Advanced Features: 11th Edition, Pearson 2019, 1040p.
3. Schildt H. Java: A Beginner's Guide: 8th Edition, McGraw-Hill Education, 2018, 684 p.
4. Васильєв О. Програмування мовою Java: Навчальна книга Богдан 2020, 696с.
5. Копитко М.Ф., Іванків К.С. Основи програмування мовою Java: Тексти лекцій. – Львів: Видавничий центр ЛНУ ім. Івана Франка, 2002. – 83 с.
6. Брнакевич І.Є., Вагін П.П. Програмування мовою Java: використання фундаментальних класів: Тексти лекцій. – Львів: Видавничий центр ЛНУ імені Івана Франка, 2002. – 75 с.

### Допоміжна література

1. Eckel B. Thinking in Java 4th Edition: Pearson, 2006, 1150 p.
2. Bloch J. Effective Java: 3rd Edition, Addison Wesley, 2017, 412 p.
3. Deitel P., Deitel H. Java How to Program, Early Objects: 11th Edition, Pearson, 2017, 1296 p.
4. Deitel P., Deitel H. Java How To Program, Late Objects: 11th Edition, Pearson, 2017, 1248 p

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

1. Іванов Л. В. Основи програмування Java // URL: [http://iwanoff.inf.ua/java\\_ua](http://iwanoff.inf.ua/java_ua)
2. Освоюємо Java – Вікіпідручник // [http://uk.wikibooks.org/wiki/Освоюємо\\_Java](http://uk.wikibooks.org/wiki/Освоюємо_Java)
3. Java Підручник. Уроки для початківців. W3Schools українською// <https://w3schoolsua.github.io/java/index.html>
4. Java Tutorial // <https://www.w3schools.com/java/>
5. Bruce Eckel. Thinking in Java, 4th Edition  
//[http://sd.blackball.lv/library/Thinking\\_in\\_Java\\_4th\\_edition.pdf](http://sd.blackball.lv/library/Thinking_in_Java_4th_edition.pdf)
6. Java tutorials: [Електроний ресурс]// Oracle and/or its affiliates, 1994-2017. URL: <http://docs.oracle.com/javase/tutorial/reallybigindex.html>
7. Java™ Platform, Standard Edition 8 API Specification: [Електроний ресурс]// Oracle and/or its affiliates, 1993- 2017. URL:<http://docs.oracle.com/javase/8/docs/api/>