

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

«ЗАТВЕРДЖУЮ»

Проректор  
з науково-педагогічної роботи

Антон ПАНТЕЛЕЙМОНОВ

« 30 » 06 2021 р.



## РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

### Об'єктно-орієнтоване програмування

рівень вищої освіти перший (бакалаврський) рівень

галузь знань 15 Автоматизація та приладобудування

спеціальність 151 Автоматизація та комп'ютерно-інтегровані технології

освітня програма Автоматизація та комп'ютерно-інтегровані технології

вид дисципліни \_\_\_\_\_

факультет \_\_\_\_\_

обов'язкова

2021 / 2022 навчальний рік

Програму рекомендовано до затвердження Вченою радою факультету комп'ютерних наук  
«30» червня 2021 року, протокол № 15

РОЗРОБНИК ПРОГРАМИ:

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

Програму схвалено на засіданні кафедри штучного інтелекту та програмного забезпечення  
Протокол від «04» червня 2021 року № 11

Завідувач кафедри штучного інтелекту та програмного забезпечення

  
Володимир КУКЛІН

Програму погоджено з гарантом освітньої програми «Автоматизація та комп'ютерно-інтегровані технології»

Гарант освітньої програми «Автоматизація та комп'ютерно-інтегровані технології»

  
Дмитро ЛАБЕНКО

Програму погоджено науково-методичною комісією факультету комп'ютерних наук  
Протокол від «25» червня 2021 року № 9

Голова науково-методичної комісії факультету комп'ютерних наук

  
Анатолій БЕРДНІКОВ

## ВСТУП

Програма навчальної дисципліни «Об'єктно-орієнтоване програмування» складена відповідно до освітньо-професійної програми підготовки першого (бакалаврського) рівня вищої освіти за спеціальністю: 151 Автоматизація та комп'ютерно-інтегровані технології.

### 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;
- набуди практичних навичок створення та відлагодження програмного кода 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, що характеризуються комплексністю та невизначеністю умов, під час професійної діяльності у галузі автоматизації або у процесі навчання, що передбачає застосування теорій та методів галузі.

*Загальні компетентності (К).*

- здатність застосовувати знання у практичних ситуаціях (К01).

*Фахові компетентності спеціальності (К)*

- здатність вільно користуватись сучасними комп'ютерними та інформаційними технологіями для вирішення професійних завдань, програмувати та використовувати прикладні та спеціалізовані комп'ютерно-інтегровані середовища для вирішення задач автоматизації (ФК 9).

1.3. Кількість кредитів – 4

1.4. Загальна кількість годин – 120

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	
56 год.	_____ год.
В т.ч. індивідуальні завдання	
5 год. (контр. робота)	_____

#### 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).*

- вміти застосовувати сучасні інформаційні технології та мати навички розробляти алгоритми та комп'ютерні програми з використанням мов високого рівня та технологій об'єктно-орієнтованого програмування, створювати бази даних та використовувати інтернет-ресурси. (ПР 03).
- вміти проектувати багаторівневі системи керування і збору даних для формування бази параметрів процесу та їх візуалізації за допомогою засобів людино-машинного інтерфейсу, використовуючи новітні комп'ютерно-інтегровані технології. (ПР09)
- вміти використовувати різноманітне спеціалізоване програмне забезпечення для розв'язування типових інженерних задач у галузі автоматизації, зокрема, математичного моделювання, автоматизованого проектування, керування базами даних, методів комп'ютерної графіки. (ПР 12).

## 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 виводу конструкторів, розрешення 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 реалізації (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.

**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. Знайомство з мовою Java. Інкапсуляція.</b>						
Тема 1. Огляд курсу.	7	2		2		3
Тема 2. Основні поняття мови Java.	7	2		2		3
Тема 3. Основні поняття мови Java.	8	2		2	1	3
Тема 4. Структурні типи даних.	7	2		2		3
Тема 5. Об'єктна модель Java.	9	2		2	1	4
<b>Разом за розділом 1</b>	<b>38</b>	<b>10</b>		<b>10</b>	<b>2</b>	<b>16</b>
<b>Розділ 2. Спадкування та поліморфізм</b>						
Тема 6. Повторне використання класів.	8	2		2		4
Тема 7. Об'єктна модель в Java.	8	2		2		4
Тема 8. Об'єктна модель в Java.	9	2		2	1	4
<b>Разом за розділом 2</b>	<b>25</b>	<b>6</b>		<b>6</b>	<b>1</b>	<b>12</b>
<b>Розділ 3. Обробка виключень, внутрішні класи і огляд стандартних пакетів.</b>						
Тема 9. Внутрішні класи.	8	2		2		4
Тема 10. Обробка виключень.	8	2		2		4
Тема 11. Пакет java.lang.	8	2		2		4
Тема 12. Сервісні класи пакету java.util.	8	2		2		4
Тема 13. Пакети і утиліти.	9	2		2	1	4
<b>Разом за розділом 3</b>	<b>41</b>	<b>10</b>		<b>10</b>	<b>1</b>	<b>20</b>
<b>Розділ 4. Бібліотека колекцій</b>						
Тема 14. Структура колекцій. Послідовні колекції.	8	2		2		4
Тема 15. Структура колекцій. Асоціативні Колекції.	13	4		4	1	4
<b>Разом за розділом 4</b>	<b>21</b>	<b>6</b>		<b>6</b>	<b>1</b>	<b>8</b>
<b>Усього годин</b>	<b>125</b>	<b>32</b>		<b>32</b>	<b>5</b>	<b>56</b>

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

№ з/п	Назва теми	Кількість годин
<b>II 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. Обработка исключений, внутренние классы и обзор 7 стандартных пакетов.</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>

#### 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	4
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	4
3	Розробка корисного класу вектор. Самостійне опрацювання лекційного матеріалу. Хорстманна[1], глава 4, Шилдт[3] (глава 7). Контроль: опитування під час практичного заняття.	8
<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	4
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	4
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.	4
<b>Розділ 3. Обработка исключений, внутренние классы и обзор стандартных пакетов.</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	6

№ з/п	Види, зміст самостійної роботи	Кількість годин
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	7
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	7
<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	4
11	Асоціативні колекції. Самостійне опрацювання лекційного матеріалу. Контроль: опитування під час практичного заняття, опитування під час прийому лабораторної роботи № 9.	4
	<b>Разом</b>	<b>56</b>

## 6. Індивідуальні завдання

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

	Лабораторні завдання
<b>2 семестр</b>	<b>6</b>

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



### 9. Схема нарахування балів

Поточний контроль, самостійна робота, індивідуальні завдання											Залікова робота	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

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

<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
Разом балів	<b>100</b>

#### Виконання практичних завдань

Протягом 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>

#### Виконання лабораторних робіт

Протягом 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 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 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

## 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.С., 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/>