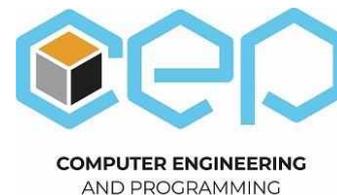




## Силабус освітнього компонента Програма навчальної дисципліни



# Об'єктно-орієнтоване програмування ч1

**Шифр та назва спеціальності**  
123 – Комп'ютерна інженерія

**Інститут**  
ННІ комп'ютерних наук та інформаційних технологій

**Освітня програма**  
Сучасне програмування, мобільні пристрої та комп'ютерні ігри (інноваційний кампус/  
Прикладна комп'ютерна інженерія)

**Кафедра**  
Комп'ютерна інженерія та програмування  
(326)

**Рівень освіти**  
Бакалавр

**Тип дисципліни**  
Профільна  
підготовка

**Семестр**  
3

**Мова викладання**  
Українська

## Викладачі, розробники



Молчанов Георгій Ігорович,  
[heorhii.molchanov@khi.edu.ua](mailto:heorhii.molchanov@khi.edu.ua);  
старший викладач кафедри комп'ютерної інженерії та програмування  
Основні курси: «Об'єктно-орієнтоване програмування ч1», «Об'єктно-орієнтоване програмування ч2», «Стандартна платформа Java».  
[Детальніше про викладача на сайті кафедри](#)

## Загальна інформація

### Анотація

«Об'єктно-орієнтоване програмування» – один з курсів профільної підготовки, що продовжує фундаментальну підготовку бакалаврів за освітньою програмою – «Сучасне програмування, мобільні пристрої та комп'ютерні ігри (інноваційний кампус)». Вона формує фахівця за освітньою кваліфікацією бакалавра з комп'ютерної інженерії. Предметом дисципліни є вивчення методів об'єктно-орієнтованого підходу до побудови програмних продуктів. Вивчаються основні питання інженерії програмного забезпечення у контексті ООП, розглядаються різні способи вирішення типових задач об'єктно-орієнтованого проектування на основі шаблонів (патернів) проектування. Дисципліна забезпечує теоретичну та практичну підготовку для впровадження методів ООП при вирішенні програмних задач, необхідних для підвищення майстерності майбутнього спеціаліста у розробці програмного забезпечення з використанням сучасних технологій.

### Мета та цілі дисципліни

Метою викладання дисципліни є: вивчення базових концепцій об'єктно-орієнтованого програмування (ООП); одержання студентами теоретичних знань про реалізацію загальних методів ООП; одержання теоретичних знань про методи вирішення типових задач об'єктно-орієнтованого проектування на основі шаблонів проектування; знайомство студентів зі спеціалізованими середовищами розробки об'єктно-орієнтованих програм; одержання студентами практичних навичок рішення прикладних задач методами ООП.

## Формат занять

Лекції, лабораторні роботи, самостійна робота, консультації. Підсумковий контроль – залік.

## Компетентності

ФК 3. Здатність створювати системне та прикладне програмне забезпечення комп'ютерних систем та мереж.

ФК 15. Здатність аргументувати вибір методів розв'язування спеціалізованих задач, критично оцінювати отримані результати та захищати прийняті рішення..

## Результати навчання

ПРН 6. Вміти застосовувати знання для ідентифікації, формулювання і розв'язування технічних задач спеціальності, використовуючи методи, що є найбільш придатними для досягнення поставлених цілей.

ПРН 13. Вміти ідентифікувати, класифікувати та описувати роботу комп'ютерних систем та їх компонентів.

## Обсяг дисципліни

Загальний обсяг дисципліни 120 год. (4 кредити ECTS): лекції – 32 год., лабораторні роботи – 16 год., самостійна робота – 72 год.

## Передумови вивчення дисципліни (пререквізити)

Для вивчення курсу студенти потребують базових знань з дисциплін: «Програмування», достатніх для:

- вивчення методів розробки програмних комплексів з використанням засобів ООП;
- вирішення задач програмування з використанням різних мов, що підтримують ООП.
- оволодіння спеціалізованими середовищами розробки об'єктно-орієнтованих програм;
- застосування різних шаблонів проектування при рішенні типових задач з використанням ООП;
- ознайомлення з напрямками розвитку технологій проектування сучасних програмних комплексів та сучасними методами розробки складних систем з використанням ООП;
- отримання навичок об'єктно-орієнтованого аналізу, проектування, розробки, розгортання, налаштування та експлуатації програмних засобів.

Крім того курс є базовим для вивчення наступних дисциплін згідно навчального плану: «Проектування мобільних застосунків», «Проектування серверних застосунків», «Паралельні та розподілені системи».

## Особливості дисципліни, методи та технології навчання

Презентація, лекція-бесіда, лекція-візуалізація, навчальна дискусія, мозкова атака, кейс-метод, демонстрування, самостійна робота, метод порівняння, метод узагальнення, метод конкретизації, метод виокремлення основного, обговорення, робота над помилками.

Вивчення курсу потребує використання програмного забезпечення Java Development Kit (JDK), Eclipse IDE for Java Developers, крім загально вживаних програм і операційних систем.

Мультимедійна дошка, проектор.

## Програма навчальної дисципліни

### Теми лекційних занять

#### Тема 1. Основи розробки програм мовою Java.

Вступ. Ключові слова. Пробільні символи. Ідентифікатори. Типи даних. Автоматичне перетворення примітивних типів. Системи числення запису числових констант. Тип арифметичного виразу. Дійсні константи. Escape-послідовності. Головна функція програми. Логічні операції по короткій схемі. Унарні операції. Операції ділення і визначення залишку від ділення. Арифметичні операції побітового зсуву. Логічні операції. Пріоритет операцій. Основні конструкції мови Java. Оператори break і continue. Позначки в Java. Масиви. Оголошення масивів. Ітерація в багатовимірних масивах. Робота з масивами.

#### Тема 2. Структура програм мовою Java.

Пакети. Структура програми. Одиниці компіляції. Приклади пакетів. Елементи пакету. Оголошення пакета. Іменування пакета. Використання пакетів. Повне кваліфіковане ім'я. Види класів. Екземпляр класу. Елементи класу. Конструктори. Методи. Поля. Блоки ініціалізації.

Спадкування. Інкапсуляція. Поліморфізм. Рівні доступу до елементів класу. Рівні доступу до класів. Конструктор за замовчуванням. Ключове слово this. Ключове слово super. Виклик конструктора попередника з конструктора нащадка.

### **Тема 3. Робота з рядками.**

Клас String. Створення об'єктів класу String. Літерали. Конструктори. Доступ до елементів рядка. Перевантаження операції "+" для рядків. Отримання рядкового представлення значень примітивних типів. Додавання об'єктних змінних, що посилаються на null, до рядка. Рядкові літерали. Пул рядків. Порівняння рядків посимвольно. Метод substring. Лексикографічне порівняння рядків. Метод concat. Метод replace. Конструктор копіювання. Клас StringBuffer. Клас StringBuilder.

### **Тема 4. Перевантаження. Перевизначення. Абстрактні класи та інтерфейси.**

Перевантаження методів. Перекриття методів. Приховування статичних методів. Обмеження при перекритті. Створення об'єкту. Значення полів за замовчуванням. Ініціалізація полів. Ключове слово final. Локальні константи. Абстрактні класи. Властивості абстрактного класу. Призначення абстрактних класів. Абстрактний метод. Інтерфейси. Використання інтерфейсів. Елементи інтерфейсу. Поля інтерфейсу. Методи інтерфейсу. Множинне спадкування інтерфейсів. Реалізація інтерфейсу. Реалізація інтерфейсів і розширення класу. Оператор instanceof для інтерфейсів. Часткова реалізація інтерфейсу.

### **Тема 5. Перетворення типів між класами. Зв'язування. Вкладені класи.**

Перетворення типів між класами. Оператор instanceof. Зв'язування. Вкладені класи. Статичні вкладені класи. Внутрішні класи. Створення об'єктів внутрішнього класу. Доступ до об'єкту зовнішнього класу. Локальні класи. Анонімні класи. Shadowing. Серіалізація внутрішніх класів.

### **Тема 6. Серіалізація об'єктів.**

Серіалізація Java-об'єктів. Десеріалізація. Java Serialization API. Два способи серіалізації. Реалізація інтерфейсу java.io.Serializable. Реалізація інтерфейсу java.io.Externalizable. Стандартна серіалізація. Класи ObjectInputStream та ObjectOutputStream. Унікальний ідентифікатор версії класу, що серіалізується.

### **Тема 7. Використання JAR. Javadoc. Класи-оболонки.**

Використання JAR. Виконання Java програми. Використання Javadoc. Коментарі документації. Класи оболонки. Ієрархія класів оболонок. Autoboxing. Unboxing. Перетворення рядка в число. Метод valueOf (String). Перетворення типів.

### **Тема 8. Основи введення/виведення. JNI.**

Basic I/O. Види потоків введення/виведення. Парні потоки. Парні класи в ієрархіях байтових потоків. Парні класи в ієрархіях символьних потоків. Поле out класу System. Класи-надбудови. Клас DataInputStream. Клас BufferedOutputStream. Клас ByteArrayInputStream. Клас FileOutputStream. Клас PushbackInputStream. Клас RandomAccessFile. Клас OutputStreamWriter. Кодування за замовчуванням. Перекодування виведення. Клас InputStreamReader. Буферизація. Поле in класу System.

Клас File. Порожній абстрактний шлях. Метод getPath. Перетворення абстрактного шляху. Метод getAbsolutePath. Метод listFiles. Інтерфейс FileFilter. Метод getParent. Метод getCanonicalPath. Java Native Interface. Інструменти і компоненти. Виклик C/C++ коду з Java програми.

### **Теми практичних занять**

Практичні заняття в рамках дисципліни не передбачені..

### **Теми лабораторних робіт**

#### **Тема 1. Структура програми мовою Java.**

Ознайомлення з JDK платформи Java SE та середовищем розробки Eclipse IDE. Головна функція програми.

#### **Тема 2. Типи даних, літерали, операції і оператори.**

Ключові слова. Ідентифікатори. Типи даних. Автоматичне перетворення примітивних типів. Основні операції.

#### **Тема 3. Алгоритмічна декомпозиція.**

Розробка програми мовою Java в середовищі Eclipse для вирішення прикладної задачі.

#### **Тема 4. Прості алгоритми обробки даних.**

Розробка простих консольних програм для платформи Java SE.

#### **Тема 5. Утилітарні класи.**

Розробка власних утилітарних класів.

## Тема 6. Обробка масивів і рядків.

Набуття навичок вирішення прикладних задач з використанням масивів і рядків.

## Тема 7. Інтерактивні консольні програми для платформи Java SE.

Реалізація діалогового режиму роботи з користувачем в консольних програмах мовою Java.

## Тема 8. Об'єктно-орієнтована декомпозиція.

Основні принципи об'єктно-орієнтованого проектування. Використання об'єктно-орієнтованого підходу для реалізації класів представлення domain-об'єктів. Управління масивом domain-об'єктів.

### Самостійна робота

Опрацювання лекційного матеріалу.

Підготовка до лабораторних робіт та модульних контролів.

Самостійне вивчення тем та питань, які не викладаються на лекційних заняттях: загальні принципи розробки програмного забезпечення: YAGNI (You Aren't Gonna Need It), KISS (Keep It Simple, Stupid), DRY/DIE (Don't Repeat Yourself / Duplication Is Evil). Чистий та брудний код. Ознаки поганого дизайну та методи його покращення (Code Smells). Антипатерни (Anti-patterns). Процес рефакторингу.

## Література та навчальні матеріали

### ОСНОВНА ЛІТЕРАТУРА

1. The Java Programming Language. Ken Arnold, James Gosling, David Holmes  
ISBN: 9780132761680
2. Core Java Volume I - Fundamentals. Cay Horstmann  
ISBN: 9780135166307
3. Java Examples in a Nutshell. David Flanagan  
ISBN: 9780596006204
4. Core Java Volume II - Advanced Features. Cay Horstmann  
ISBN: 9780135166314
5. Thinking in Java. Bruce Eckel  
ISBN: 9780131872486
6. Effective Java. Joshua Bloch  
ISBN: 9780134685991
7. Design Patterns: Elements of Reusable Object-Oriented Software. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides  
ISBN: 9780201633610, 9780201634983

### ДОДАТКОВА ЛІТЕРАТУРА

8. Applied Java Patterns. Stephen Stelting, Olav Maassen  
ISBN: 9780130935380
9. Bug Patterns In Java. Eric Allen  
ISBN: 9781590590614
10. Bitter Java. Bruce A. Tate  
ISBN: 9781930110434

### ІНФОРМАЦІЙНІ РЕСУРСИ В ІНТЕРНЕТІ

1. Java Language Specification (Java SE 21 Edition)  
<https://docs.oracle.com/javase/specs/jls/se21/jls21.pdf>
2. JDK 21 Documentation  
<https://docs.oracle.com/en/java/javase/21/>
3. The Java Tutorials Bundle  
<https://www.oracle.com/java/technologies/javase/java-tutorial-downloads.html>
4. Java SE Client Technologies: JavaFX  
<https://docs.oracle.com/javase/8/javase-clienttechnologies.htm>
5. Introduction to Java programming - Tutorial  
<https://www.vogella.com/tutorials/JavaIntroduction/article.html>

6. Using the Eclipse IDE for Java programming - Tutorial  
<https://www.vogella.com/tutorials/Eclipse/article.html>
7. Eclipse Shortcuts - Tutorial  
<https://www.vogella.com/tutorials/EclipseShortcuts/article.html>
8. Java Debugging with Eclipse - Tutorial  
<https://www.vogella.com/tutorials/EclipseDebugging/article.html>
9. Introduction to Checkstyle for checking Java code quality - Tutorial  
<https://www.vogella.com/tutorials/Checkstyle/article.html>
10. Java Tutorials  
<https://docs.oracle.com/javase/tutorial/index.html>
11. How to Write Doc Comments  
<http://www.oracle.com/technetwork/java/javase/documentation/index-137868.html>
12. Google Java Style Guide  
<https://google.github.io/styleguide/javaguide.html>

## Система оцінювання

### Критерії оцінювання успішності студента та розподіл балів

Оцінювання проводиться за 100-бальною шкалою. Бали нараховуються за наступним співвідношенням:  
модульні контролі - 40 балів; практичні заняття - 30 балів; залік - 30 балів.

### Шкала оцінювання

Сума балів	Національна оцінка	ECTS
90–100	Відмінно	A
82–89	Добре	B
75–81	Добре	C
64–74	Задовільно	D
60–63	Задовільно	E
35–59	Незадовільно (потрібне додаткове вивчення)	FX
1–34	Незадовільно (потрібне повторне вивчення)	F

## Норми академічної етики і політика курсу

Студент повинен дотримуватися «Кодексу етики академічних взаємовідносин та доброчесності НТУ «ХПІ»: виявляти дисциплінованість, вихованість, доброзичливість, чесність, відповідальність. Конфліктні ситуації повинні відкрито обговорюватися в навчальних групах з викладачем, а при неможливості вирішення конфлікту – доводитися до відома співробітників дирекції інституту. Нормативно-правове забезпечення впровадження принципів академічної доброчесності НТУ «ХПІ» розміщено на сайті: <http://blogs.kpi.kharkov.ua/v2/nv/akademichna-dobrochesnist/>

## Погодження

Силабус погоджено

Дата погодження, підпис  
22.04.2024



Завідувач кафедри  
Олександр ЗАКОВОРОТНИЙ

Дата погодження, підпис  
22.04.2024



Гарант ОП  
Олександр ЗАКОВОРОТНИЙ

