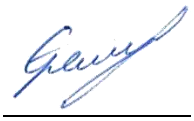
Міністерство освіти і науки України
Національний аерокосмічний університет
«Харківський авіаційний інститут»

Кафедра інформаційно-комунікаційних технологій
ім. О. О. Зеленського (№ 504)

ЗАТВЕРДЖУЮ

Гарант освітньої програми



(підпис) Олег ЄРЕМЕЄВ (ім'я та прізвище)

31 серпня 2026 р.

СИЛАБУС ОБОВ'ЯЗКОВОЇ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Об'єктно-орієнтоване програмування та ефективні практики

(назва навчальної дисципліни)

Галузь знань: F «Інформаційні технології»
(шифр і найменування галузі знань)

Спеціальність: F6 «Інформаційні системи і технології»
(код і найменування спеціальності)

Освітня програма: «Штучний інтелект та інформаційні системи»
(найменування освітньої програми)

Рівень вищої освіти: *перший (*бакалаврський*)*

Силабус введено в дію з 01.09.2026

Харків – 2026 р.

Розробник: Рубель О. С., доцент, к.т.н.
(прізвище та ініціали, посада, науковий ступінь і вчене звання) (підпис)

Силабус навчальної дисципліни розглянуто на засіданні кафедри інформаційно-комунікаційних технологій ім. О. О. Зеленського (№ 504)
(назва кафедри)

Протокол № 1 від «31» серпня 2026 р.

Завідувач кафедри д.т.н., професор (науковий ступінь і вчене звання) (підпис) Володимир ЛУКІН (ім'я та прізвище)

Погоджено з представником здобувачів освіти:

здобувачка гр. 528 (підпис) Ангеліна ПОГРЕБНЯК (ім'я та прізвище)

1. Загальна інформація про викладача



ПІБ: Рубель Олексій Сергійович

Посада: доцент

Науковий ступінь: к.т.н.

Вчене звання: -

Перелік дисциплін, які викладає:
Об'єктно-орієнтоване програмування та ефективні практики, Програмування на Python

Напрями наукових досліджень:
цифрова обробка зображень, машинне навчання

Контактна інформація: o.rubel@khai.edu

2. Опис навчальної дисципліни

Форма здобуття освіти	Денна
Семестр	З/1*
Мова викладання	Українська
Тип дисципліни	Обов'язкова
Обсяг дисципліни: кредити ЄКТС/ кількість годин	кредити ЄКТС 4,5/ годин 135 (64 аудиторних, з яких: лекції – 32, практичні – 32; СРЗ – 71)
Види навчальної діяльності	Лекції, практичні заняття, самостійна робота
Види контролю	Поточний контроль, модульний контроль, семестровий контроль – залік
Пререквізити	«Основи програмування», «Алгоритми і структури даних»

*Для здобувачів, які навчаються за скороченим терміном.

3. Мета та завдання навчальної дисципліни, переліки компетентностей та очікуваних результатів навчання

Мета – набуття студентами знань і вмінь, використання їх у своїй практичній роботі, пов'язаній з розробкою програмного забезпечення за допомогою мови програмування Java, використанням усіх елементів об’єктно-орієнтованого програмування у ході розробки програмного забезпечення інформаційно-комунікаційних систем і сервісів.

Завдання – вивчення сучасних засобів розробки програмного забезпечення, мови програмування Java, об’єктно-орієнтованого програмування.

Компетентності, які набуваються:

Інтегральна компетентність:

Здатність розв’язувати складні спеціалізовані задачі та практичні проблеми в області інформаційних систем та технологій, або в процесі навчання, що характеризуються комплексністю та невизначеністю умов, які потребують застосування теорії та методів інформаційних технологій.

Загальні компетентності

Після закінчення цієї програми здобувач освіти буде здатен:

- Здатність до абстрактного мислення, аналізу та синтезу;
- Здатність застосовувати знання у практичних ситуаціях;
- Здатність вчитися і оволодівати сучасними знаннями;
- Здатність до пошуку, оброблення та узагальнення інформації з різних джерел;
- Здатність ухвалювати рішення та діяти, дотримуючись принципу неприпустимості корупції та будь-яких інших проявів недоброчесності.

Спеціальні компетентності

Після закінчення цієї програми здобувач освіти буде здатен:

- Здатність проектувати, розробляти та використовувати засоби реалізації інформаційних систем, технологій та інфокомунікацій (методичні, інформаційні, алгоритмічні, технічні, програмні та інші);
- Здатність управляти та користуватися сучасними інформаційно-комунікаційними системами та технологіями (у тому числі такими, що базуються на використанні Інтернет).

Програмні результати навчання:

- Використовувати базові знання інформатики й сучасних інформаційних систем та технологій, навички програмування, технології безпечної роботи в комп'ютерних мережах, методи створення баз даних та інтернет-ресурсів, технології розроблення алгоритмів і комп’ютерних програм мовами високого рівня із застосуванням об’єктно-орієнтованого програмування для розв’язання задач проектування і використання інформаційних систем та технологій;
- Демонструвати знання сучасного рівня технологій інформаційних систем, практичні навички програмування та використання прикладних і спеціалізованих комп’ютерних систем та середовищ з метою їх запровадження у професійній діяльності;
- Обґрунтовувати вибір технічної структури та розробляти відповідне програмне забезпечення, що входить до складу інформаційних систем та технологій.

4. Зміст навчальної дисципліни

МОДУЛЬ 1

Змістовний модуль 1. Основи мови програмування Java

Тема 1. Java і віртуальна машина Java. Предмет вивчення і задачі дисципліни. Місце дисципліни в навчальному плані. Необхідне програмне забезпечення. Віртуальна машина Java (JVM). Java Runtime Environment (JRE). Java Development Kit (JDK). Java SE. Компоненти JVM. Загальна архітектура JVM. Платформа Java. Як писати і запускати програми в середовищі Java. Компіляція програми. Java компілятор javac. Запуск програм. Інструменти для розробки на Java. IntelliJ IDEA. Установка середовища розробки JetBrains IntelliJ IDEA.

Лекція № 1. Вступ до мови Java. Віртуальна машина Java.

Практична робота «Установка Java Development Kit. Перша програма».

Самостійна робота здобувача освіти: опрацювання матеріалу лекції, виконання практичної роботи.

Тема 2. Основи Java. Створення нових проектів в IntelliJ IDEA і налагоджування середовища розробки. Конфігурації запуску IntelliJ IDEA. Створення першого проекту. Інструменти збірки програм. Архів Java (JAR файли). Огляд Gradle. Створення програм за допомогою Gradle. Вступ до Java. Основні особливості Java. Деякі загальні правила для Java. Основні синтаксичні особливості мови програмування Java. Основні літерали: числа, рядки і символи. Змінні і значення. Написання першої програми. Друк даних – println() і print(). Читання введених користувачем даних за допомогою сканера. Scanner. Читання багаторядкового введення. Форматований вивід даних. Методи System.out.print() і System.out.println(). Метод printf(). Метод String.format().

Лекція № 2. Типи даних і змінні.

Самостійна робота здобувача освіти: опрацювання матеріалу лекції.

Тема 3. Типи даних і змінні. Типи даних і змінні. Примітивні типи даних. Доступ до значення змінної. Висновок типу (type inference). Типи даних і їх розміри. Числа. Цілі числа. Числа з плаваючою комою – float і double. Читання чисел з плаваючою комою. Десятковий роздільник. Символи. Булеві типи. Ключове слово new. Перетворення типів – неявне та явне. Примітивні (primitive) типи та типи посилань (reference). Ключове слово new. Зберігання даних у пам’яті. Два основних простори пам'яті: стек і купа. Тип null. Константи. Кінцеві змінні. Ключове слово final. Коли використовувати кінцеві змінні.

Лекція № 2. Типи даних і змінні.

Самостійна робота здобувача освіти: опрацювання матеріалу лекції.

Тема 4. Операції над примітивними типами. Операції над цілими числами. Форми оператора присвоювання. Читання чисел зі стандартного введення. Інкремент і декремент. Арифметичні операції. Порівняння значень. Оператори відношення. Пріоритет операторів. Символи і escape-послідовності. Булев тип і операції з ним. Логічні оператори (NOT, AND, OR і XOR). Пріоритет логічних операторів.

Лекція № 3.Операції над примітивними типами. Керуючи оператори потоку.

Практична робота «Робота з консоллю».

Самостійна робота здобувача освіти: опрацювання матеріалу лекції, виконання практичної роботи.

Тема 5. Керуючі оператори потоку. Умовні оператори. Випадки використання if, if-else, if-else-if. Вкладені конструкції. Тернарний оператор. Оператор switch, case, default. Використання switch як вираз. Цикли. Цикл for. Вкладені цикли. Цикл while. Цикл do-while. Читання послідовності невідомої довжини. Розгалуження. Оператори переходу – break, continue, return.

Лекція № 3.Операції над примітивними типами. Керуючи оператори потоку.

Самостійна робота здобувача освіти: опрацювання матеріалу лекції.

Тема 6. Масиви і рядки. Масиви в Java. Оголошення і ініціалізація масивів. Створення масиву із заданими елементами. Створення масиву за допомогою ключового слова "new". Довжина масиву. Доступ до елементів. Ітерація по масивах. Обробка масивів за допомогою циклів. Читання масиву зі стандартного введення. Використання вбудованих методів і функцій для роботи з масивами. Тип String. Створення рядків. Довжину та символи рядка. Корисні методи роботи з рядками. Конкатенація рядків. Додавання значень до рядка. Порівняння рядків. Обробка рядків. Метод split. Ітерація по рядку.

Лекція № 4.Масиви і рядки.

Практична робота «Проект з роботою з текстом і операціями вводу/виводу».

Самостійна робота здобувача освіти: опрацювання матеріалу лекції, виконання практичної роботи.

Тема 7. Стиль коду.

Коментарі. Багаторядкові коментарі. Іменування змінних. Правила іменування змінних. Положення про стиль написання коду. Конвенції Java. Кількість пробілів. Довжина рядка.

Лекція № 4.Масиви і рядки.

Самостійна робота здобувача освіти: опрацювання матеріалу лекції, підготовка до модульної контрольної роботи.

Модульний контроль 1 «Основи мови програмування Java».

Змістовний модуль 2. Об'єктно-орієнтоване програмування в Java

Тема 8. Основи об'єктно-орієнтованого програмування. Парадигми програмування. Об'єктно-орієнтоване програмування (ООП). Визначення ООП. Ознайомлення з поняттями класів, об'єктів, атрибутів і методів у Java. Принципи ООП: успадкування, поліморфізм, інкапсуляція, абстракція. Основні відомості про класи. Оголошення класів. Константи класу. Властивості класів. Атрибути (змінні екземпляра і класу). Методи класу. Оголошення методів. Синтаксис методів. Метод new. Модифікатори методів. Метод main. Статичні методи (static), методи класу і екземпляра класу. Рекурсивні функції. Перевантаження методу. Передача масивів методам. Аргументи командного рядка. Методи екземплярів (instance). Порядок створення екземплярів класів. Ініціалізація нових екземплярів. Виклик методу. Конструктор. Перевантаження конструктора. Виклик конструкторів з інших конструкторів. Поняття про об'єкти і їх використання. Методи доступу: гетери і сетери. Зміна атрибутів, пов'язаних з об'єктом через методи. Типові і параметризовані конструктори. Архітектура програмного забезпечення. Застосування UML для моделювання об'єктно-орієнтованих систем. Діаграми UML. UML діаграма компонентів.

Лекція № 5. Основи об'єктно-орієнтованого програмування.

Лекція № 6. Перевантаження конструктора. Методи доступу: гетери і сетери.

Самостійна робота здобувача освіти: опрацювання матеріалу лекцій.

Тема 9. Класи і особливості використання ООП в Java. Різновиди класів в Java. Організація класів у пакети. Імпортування класів. Модулі. Модифікатори доступу: public, protected, private, package-private (default). Документування класів і пакетів javadoc. Перевантаження конструкторів. Ієрархія класів. Особливості успадкування в Java. Доступ до полів і методів суперкласу. Ключове слово "super". Виклик конструктора суперкласу. Посилання на об'єкти підкласу. Доступ до полів і методів через посилання суперкласу. Ключове слово instanceof. Поліморфізм в Java. Поліморфізм в ієрархії класів. Перевизначення методів. Правила перевизначення методів. Заборона перевизначення. Перевизначення та перевантаження методів разом. Приховування статичних методів. Внутрішні класи. Вкладені класи. Анонімний клас. Класи-перерахування enum. Визначення enum. Методи обробки enum. Поля і методи в enum. Абстрактний клас. Інтерфейси. Відмінності між абстрактними класами і інтерфейсами. Оголошення інтерфейсів. Спільне використання абстрактних класів і інтерфейсів. Реалізація інтерфейсів. Реалізація і розширення декількох інтерфейсів. Статичні методи в інтерфейсах. Лямбда-вирази і функціональні інтерфейси. Анотації. Додавання анотацій. Користувальницькі анотації і типи анотацій. UML діаграма класів. Файли класів і байт-код. Компіляція вихідного файлу. Стек викликів і його структура. Стек і виконання методів. Переповнення стека. Трасування стека. Огляд класу StackTraceElement.

Лекція № 7. Ієрархія класів і особливості успадкування.

Практична робота «Реалізація гри «Хрестики-Нулики»».

Лекція № 8. Поліморфізм в Java. Внутрішні і вкладені класи.

Лекція № 9. Абстрактні класи та інтерфейси.

Практична робота «Використанням об'єктно-орієнтованого програмування у створенні складних моделей».

Самостійна робота здобувача освіти: опрацювання матеріалу лекцій, виконання практичних робіт.

Тема 10. Виключення і їх обробка. Помилки під час компіляції і помилки під час виконання. Винятки. Ієрархія винятків. Базовим клас всіх винятків java.lang.Throwable. Створення винятків. Оператор throw. Обробка винятків. Оператор try-catch. Перехоплення кількох винятків. Блок finally. NullPointerException (NPE). Правила уникнення NPE. Створення власних підкласів виключень.

Лекція № 10. Обробка виключень.

Практична робота «Обробка виключень».

Самостійна робота здобувача освіти: опрацювання матеріалу лекції, виконання практичної роботи.

Тема 11. Робота з даними. Клас File. Основні методи. Читання файлів. Запис файлів. Клас FileWriter. Клас PrintWriter. Керування файлами. Створення файлів. Регулярні вирази в Java. Класи java.util.regex.Pattern і java.util.regex.Matcher. Огляд класів Math, LocalDate, LocalTime і LocalDateTime.

Лекція № 11. Робота з даними.

Практична робота «Розробка багатофункціонального калькулятора».

Самостійна робота здобувача освіти: опрацювання матеріалу лекції, виконання практичної роботи.

Тема 12. Колекції і огляд фреймворку Collections. Колекції. Змінні і незмінні колекції. Клас ArrayList і його методи. EnumSet. Інтерфейс Collections. Інтерфейс Set. Інтерфейс List. Інтерфейс Map. Стек. Дек. Розгляд класів String, StringBuffer, StringBuilder.

Лекція № 12. Колекції.

Практична робота «Робота з колекціями».

Самостійна робота здобувача освіти: опрацювання матеріалу лекції, виконання практичної роботи.

Тема 13. Основи шаблонів проектування. Шаблони проектування. Принципи SOLID. Творчі, структурні і поведінкові шаблони проектування. Шаблон Singleton. Шаблон фабричний метод (factory method). Шаблон фасад (facade). Шаблон адаптер. Шаблон декоратор. Шаблон ланцюжок відповідальностей. Шаблони проектування MVC і MVP. Чиста архітектура (Clean architecture) як підхід до організації архітектури програмного забезпечення.

Лекція № 13. Принципи SOLID. Основи шаблонів проектування.

Лекція № 14. Шаблони проектування MVC і MVP. Чиста архітектура.

Самостійна робота здобувача освіти: опрацювання матеріалу лекцій.

Тема 14. Основи багатопоточності в Java. Що таке потоки. Потокова модель в Java. Потоки як об’єкти. Клас потоків java.lang.Thread. Головний потік (main). Вхідні і вихідні потоки. Потоки байтів і символів. Буферизовані потоки. Створення власних потоків. Запуск потоків. Проста багатопотокова програма. Керування потоками: призупинення, відновлення і зупинка потоків. Винятки в потоках. Стани потоку – Thread.State enum клас. Життєвий цикл потоку. Обмін даними між потоками. Пріоритети потоків. Синхронізація потоків. Об’єднання і синхронізація потоків: ключове слово synchronized і бібліотека java.util.concurrent.

Лекція № 15. Основи багатопочтоності в Java.

Практична робота «Робота з потоками».

Самостійна робота здобувача освіти: опрацювання матеріалу лекції, виконання практичної роботи.

Тема 15. Інструменти тестування і налагодження програм. Налагодження (debugging) простих конструкцій. Використання assert. Точки зупинки конструктора і методу. Точки зупинки винятків. Покрове налагодження (stepping). Вступ до логування в Java (Java Logging). Клас Logger. Logback. Логування/'printf' налагодження. Основи модульного тестування. Модульне тестування з JUnit. Написання тестів. Клас Assertions фреймворку JUnit. Запуск тестів. Результати тестування.

Лекція № 16. Тестування з JUnit.

Практична робота «Робота з JUnit».

Самостійна робота здобувача освіти: опрацювання матеріалу лекції, виконання практичної роботи, підготовка до модульної контрольної роботи.

Модульний контроль 2 «Об'єктно-орієнтоване програмування в Java».

5. Індивідуальні завдання

Не передбачені навчальним планом.

6. Методи навчання

Використовуються наступні методи навчання: словесні, наочні та практичні, а саме: проведення лекцій (із застосуванням пояснювально-ілюстративного та проблемного викладання навчального матеріалу), робота на практичних заняттях (із застосуванням як репродуктивного методу, так і проблемного викладання), консультації протягом семестру, самостійна робота за матеріалами, опублікованими кафедрою, використання мережі Internet.

7. Методи контролю

Поточний контроль: виконання практичних завдань, контроль засвоєння навчального матеріалу, запланованого на самостійне опрацювання студентом.

Контроль:

- поточний контроль (оцінювання роботи студентів на практичних заняттях);
- модульний контроль за змістовними модулями;

Семестровий контроль – залік.

8. Критерії оцінювання та розподіл балів, які отримують здобувачі освіти

Таблиця 8.1 – Розподіл балів, які отримують здобувачі освіти

Складові навчальної роботи	Бали за одне заняття (завдання)	Кількість занять (завдань)	Сумарна кількість балів
Змістовний модуль 1			
Модульний контроль (теми 1–7)	0…30	1	**0…30**
Виконання і захист практичних робіт	0..4	3	**0..12**
Змістовний модуль 2			
Виконання і захист практичних робіт	0…4	7	**0…28**
Модульний контроль (теми 8-15)	0…30	1	**0…30**
Усього за семестр			**0..100**

Семестровий контроль (залік) проводиться у разі відмови здобувача освіти від балів підсумкового контролю й за наявності допуску до заліку. Під час складання семестрового заліку здобувач освіти має можливість отримати максимум 100 балів.

Білет для заліку складається з тридцяти тестових теоретичних і практичних питань.

Таблиця 8.2 – Шкали оцінювання: бальна і традиційна

Сума балів	Оцінка за традиційною шкалою	
	Іспит, диференційний залік	Залік
90 – 100	Відмінно	Зараховано
75 – 89	Добре	
60 – 74	Задовільно	
0 – 59	Незадовільно	Не зараховано

Критерії оцінювання роботи здобувача освіти протягом семестру

Задовільно (60-74). Показати мінімум знань та умінь. Вміти викладати отримані знання в усній чи письмовій формі; при цьому неповний обсяг засвоєного навчального матеріалу не повинен перешкоджати засвоєнню наступного програмного матеріалу; допускаються окремі істотні помилки, виправлені за допомогою викладача. Захистити всі практичні завдання і здати тестування. Знати синтаксис мови програмування Java і базові принципи програмування. Мати уявлення про об'єктно-орієнтоване програмування.

Добре (75-89). Твердо знати мінімум, захистити всі практичні завдання, здати тестування і поза аудиторну самостійну роботу. Уміти: окрім наведених базових знань застосовувати шаблони проектування для написання гнучкого і придатного для масштабування програмного забезпечення, знати основні елементи і прийоми об'єктно-орієнтованого програмування.

Відмінно (90-100). Здати всі контрольні точки з оцінкою «відмінно». Досконально знати всі теми та уміти застосовувати їх. Відповідати на теоретичні питання на високому рівні в межах конспекту лекцій, рекомендованих підручників та додаткової літератури, вміти аналізувати надану інформацію та пропонувати нестандартні рішення, вміти їх обґрунтовувати. Досконально знати всі теми та уміти застосовувати їх на практиці.

9. Політика навчального курсу

Відвідування занять. Регуляція пропусків. Відвідування практичних занять є обов'язковим. Допускаються пропуски занять з таких поважних причин, як хвороба, сімейні обставини, участь в олімпіаді, творчому конкурсі тощо за попереднього домовленістю та згодою викладача за умови дозволу деканату (викладачу надаються копії довідок, документів або інших матеріалів, які підтверджують причину пропуску). За умов дистанційного навчання поважними причинами пропусків можуть бути порушення у функціонуванні електромереж і збої в роботі Інтернету. Відпрацювання пропущених занять є обов'язковим незалежно від причини пропущеного заняття. Відпрацювання практичного заняття полягає в самостійному виконанні завдань пропущеної практичної роботи.

Дотримання вимог академічної доброчесності здобувачами освіти під час вивчення навчальної дисципліни. Під час вивчення навчальної дисципліни здобувачі освіти мають дотримуватися загальноприйнятих морально-етичних норм і правил поведінки, вимог академічної доброчесності, передбачених Кодексом академічної доброчесності та забезпечення її дотримання Національного аерокосмічного університету «Харківський авіаційний інститут». Очікується, що роботи здобувачів освіти будуть їх оригінальними дослідженнями або міркуваннями. Відсутність посилань на використані джерела, фабрикування джерел, списування, втручання в роботу інших здобувачів освіти становлять, але не обмежують, приклади можливої академічної недоброчесності. Виявлення ознак академічної недоброчесності в письмовій роботі здобувача освіти є підставою для її незарахування викладачем незалежно від масштабів плагіату чи обману.

Вирішення конфліктів. Порядок і процедури врегулювання конфліктів, пов'язаних із корупційними діями, зіткненням інтересів, різними формами дискримінації, сексуальними домаганнями, міжособистісними стосунками та іншими ситуаціями, що можуть виникнути під час навчання, а також правила етичної поведінки регламентуються Кодексом етики в Національному аерокосмічному університеті «Харківський авіаційний інститут».

10. Методичне забезпечення

1. Скоб Ю.О. Основи програмування мовою JAVA: навч. посіб. до лаб. практикуму / Ю. О. Скоб, М. Л. Угрюмов, В. О. Халтурін ; М-во освіти і науки України, Національний аерокосмічний університет ім. М. Є. Жуковського "Харківський авіаційний інститут". - Харків. - Національний аерокосмічний університет ім. М. Є. Жуковського " Харківський

авіаційний інститут ", 2017. - 112 с.

2. Сторінка дисципліни у системі дистанційного навчання *Ментор* знаходиться за посиланням: https://mentor.khai.edu/enrol/index.php?id=4151

11. Рекомендована література

Базова

1. Sierra K. Head First Java: A Brain-Friendly Guide, 3rd Edition / K. Sierra, B. Bates, Trisha Gee. – O'Reilly Media, 2022. – 752 p.
2. Schildt H. Java: The Complete Reference, Twelfth Edition 12th Edition / H. Schildt. – McGraw Hill, 2021. – 1280 p.

Допоміжна

1. Freeman E., Robson E. Head First Design Patterns: Building Extensible and Maintainable Object-Oriented Software / E. Freeman, E. Robson. – O'Reilly Media, 2021. – 669 p.

12. Інформаційні ресурси

1. Науково-технічна бібліотека ХАІ - Режим доступу: http://library.khai.edu
2. Навчальні матеріли з Java - https://docs.oracle.com/javase/tutorial/tutorialLearningPaths.html
3. Інтерактивний підручник з Java - https://www.learnjavaonline.org/
4. Посібник зі стилю для коду Java - https://google.github.io/styleguide/javaguide.html#s6.1-override-annotation