



**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ТОРГОВЕЛЬНО-
ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ**
Факультет інформаційних технологій
Кафедра інженерії програмного забезпечення
та кібербезпеки

СИЛАБУС (SYLLABUS)
Дисципліна «Технологія Java/ Java technology»

ІНФОРМАЦІЯ ПРО ВИКЛАДАЧА

Викладач	Цензура Микола Олександрович
Науковий ступінь	Кандидат технічних наук
Вчене звання	Доцент
Посада	Доцент кафедри інженерії програмного забезпечення та кібербезпеки
Адреса кафедри	м.Київ, вул. Кіото 19, каб. Б-524, Б-531
E-mail	progen@ukr.net
Консультації	Відповідно до графіку індивідуальних консультацій на сайті кафедри

ПОЛІТИКА АКАДЕМІЧНОЇ ДОБРОЧЕСНОСТІ

<https://knute.edu.ua/file/NjY4NQ==/bf27ad9293fa2bb6f9b2c3031d4b6e4a.pdf>

Дотримання академічної доброчесності передбачає:

- самостійне виконання навчальних завдань, завдань поточного та підсумкового контролю результатів навчання (для осіб з особливими освітніми потребами ця вимога застосовується з урахуванням їхніх індивідуальних потреб і можливостей);
- посилання на джерела інформації у разі використання не авторських ідей, розробок, тверджень, відомостей і т.п.;
- дотримання норм законодавства про авторське право і суміжні права;
- надання достовірної інформації про результати власної наукової діяльності, використанні методики досліджень і джерела інформації.

Порушенням академічної доброчесності вважається:

- академічний плагіат – оприлюднення (частково або повністю) наукових (творчих) результатів, отриманих іншими особами, як результатів власного дослідження (творчості) та/або відтворення опублікованих текстів (оприлюднених творів мистецтва) інших авторів без зазначення авторства;
- самоплагіат – оприлюднення (частково або повністю) власних раніше опублікованих наукових результатів як нових наукових результатів;
- фабрикація – вигадкування даних чи фактів, що використовуються в наукових дослідженнях;
- фальсифікація – свідомо зміна чи модифікація вже наявних даних, що стосуються наукових досліджень.

За порушення академічної доброчесності здобувачі освіти можуть бути притягнені до академічної відповідальності:

- повторне проходження оцінювання (модульний контроль, іспит, залік тощо);
- повторне проходження відповідного освітнього компонента освітньо-професійної програми;
- відрахування з Університету;
- позбавлення наданих університетом пільг;
- відмова у присудженні відповідного ступеня вищої освіти;

ПОЛІТИКА ЩОДО ВІДВІДУВАННЯ ЗАНЯТЬ

- відвідування занять є обов'язковим;
- Студент, який пропустив практичне заняття, самостійно вивчає матеріал (при виникненні питань може звертатися за консультацією згідно розкладу консультацій викладачів оприлюдненого на сайті кафедри) за наведеними джерелами, виконує завдання і здає його викладачу.
- за об'єктивних причин (наприклад, хвороба, міжнародне стажування та ін.) навчання може відбуватись в он-лайн формі за погодженням із викладачем дисципліни.

ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Назва дисципліни / тип дисципліни	Технологія Java / вибіркова
Навчальний рік	2022-2023, 2023-2024
Факультет	Факультет інформаційних технологій
Курс	3-4
Семестр	6-8
Освітній ступінь	Бакалавр
Галузь знань	12 «Інформаційні технології»
Спеціальність	122 «Комп'ютерні науки»
Загальна характеристика	Кількість годин –180 Кількість кредитів – 6 Види занять: лекції, практичні, самостійна робота. Співвідношення аудиторних годин і годин самостійної роботи - 68/112 Мова викладання – українська Форма викладання – очна
Підсумковий контроль	Екзамен
Програмне забезпечення	Java Development Kit (JDK)
Обладнання	Проектор, комп'ютерна техніка із встановленим програмним забезпеченням та доступом до мережі Інтернет.
Необхідні попередні дисципліни	«Основи програмування», «Об'єктно-орієнтоване програмування», «Алгоритмізація і програмування».
Методика вивчення	Методика вивчення дисципліни полягає у набутті студентами знань теоретичного і практично-прикладного характеру під час лекцій, лабораторних занять, самостійної роботи та вивчення першоджерел і навчально-методичної літератури.
Мета і завдання	Метою вивчення дисципліни «Технологія Java» є формування у здобувачів вищої освіти базової системи знань та вмінь в сфері використання основних концепцій мови Java, сучасних технологій розробки і реалізації програм на базі об'єктно-орієнтованої методології. Завданням дисципліни «Технологія Java» є: теоретична та практична підготовка студентів у напрямку реалізації проектів з розробки програмного забезпечення; вивчення принципів програмної реалізації та застосування базових конструкцій мови програмування Java; створення та робота з масивами; вивчення фундаментальних теоретичних основ професійної діяльності в галузі розробки програмного забезпечення; отримання практичних навичок ведення професійної діяльності в умовах наближених до реальних.

Місце дисципліни в освітньо-професійній програмі	
Фахові компетентності (результати навчання)	<p>СК 3 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.</p> <p>СК 8 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.</p>
Програмні результати навчання	<p>ПР 9 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.</p> <p>ПР 13 Володіти мовами системного програмування та методами розробки програм, що взаємодіють з компонентами комп'ютерних систем, знати мережні технології, архітектури комп'ютерних мереж, мати практичні навички технології адміністрування комп'ютерних мереж та їх програмного забезпечення</p> <p>ПР 14 Володіти мовами системного програмування та методами розробки програм, що взаємодіють з компонентами комп'ютерних систем, знати мережні технології, архітектури комп'ютерних мереж, мати практичні навички технології адміністрування комп'ютерних мереж та їх програмного забезпечення.</p>

ТЕМАТИКА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Тема 1. Огляд структури та можливостей системи програмування Java.

Предмет, завдання і структура дисципліни. Рекомендована література. Термінологія, яка пов'язана з Java: простота, об'єктна орієнтованість, стійкість, багато поточність, архітектурна нейтральність і висока продуктивність, розподілений та динамічний характер. JDK – Java Development Kit засіб розробника. JRE – Java Runtime Environment мінімальна реалізація віртуальної машини Java. Java-платформи: Java SE (J2SE); Standard Edition; Java EE (J2E); Java ME (J2ME) Послідовність дій при створенні додатків за допомогою JDK. Компіляція, інтерпретація, платформна незалежність.

Тема 2. Лексика мови програмування Java.

Лексика і лексеми. Колекція ідентифікаторів, літералів, коментарів, пробілів, операторів та ключових слів. Системи кодування Java ASCII та Unicode. Особливості представлення символів в Unicode. Типи пробільних символів та їх кодування. Призначення та типи коментарів. Ідентифікатори та вимоги до їх структури. Типи літералів та правила їх застосування. Escape-послідовності для відображення спеціальних символів.

Тема 3. Типи даних та оператори.

Типи даних та змінні базового типу: цілочисленні, типи з плаваючою точкою, символьні, об'єктні та булеві значення. Об'ява змінної, динамічна ініціалізація, область видимості і тривалість існування змінних. Позиційні системи числення: десяткові, двійкові, вісімкові та шістнадцяткові. Приклад першої програми, в якій використовується декілька змінних. Операції над примітивними типами. Перетворення та приведення типів даних: автоматичне перетворення типів; перетворення несумісних типів.

Тема 4. Базові оператори.

Основні оператори: арифметичні; логічні; оператори порівняння та побітові (порозрядні). Арифметичні оператори: основні арифметичні оператори; оператор ділення по модулю; складові арифметичні оператори з привласненням; скорочені форми операторів; інкремент та

декремент. Побітові оператори: побітові логічні оператори; зсув вліво; зсув вправо; зсув вправо без урахування знака. Тернарний оператор. Оператор привласнення. Пріоритети операцій.

Тема 5. Керуючи структури.

Оператори розгалуження: if та switch. Блок схеми розгалуженого процесу. Загальний синтаксис оператора if у конструкціях: стандартна форма; спрощена форма; вкладені умовні оператори – конструкція if–else–if. Switch – оператор вибору з варіантом розгалуженням: загальний синтаксис; блок схема оператора; вкладені оператори switch. Поняття циклу в програмуванні. Оператор while та його модифікація do-while: синтаксис, блок схеми виконання операторів та їх особливості. Цикл з параметрами for: синтаксис, блок схема циклу, об'ява керуючих змінних циклу, різні способи виклику оператора циклу. Оператори переходу. Оператор break: завершення послідовності дій в операторі switch; виходу з циклу; як оператор безумовного переходу. Оператор continue - миттєвий перехід до наступної ітерації. Оператор return - явний вихід із методу.

Тема 6. Створення та робота з масивами.

Загальні поняття масиву. Особливості внутрішньої структури масивів в Java. Три етапи створення масивів. Створення одновимірного масиву та його індексація. Визначення розмірності масиву. Привласнення та порівняння масивів. Створення двовимірного масиву та його індексація. Логічне представлення двовимірного масиву. Основні операції з масивами: ініціалізація масивів, створення «рваного» символьного масиву випадкового розміру, упорядкування масиву та пошук елементів масиву. Особливості застосування циклу for для перебору елементів масиву. Змінні перелічувального типу та область їх застосування. Розширення можливостей обробки масивів даних за допомогою класу ARRYS.

Тема 7. Класи та об'єкти.

Визначення класів, структура класу. Оголошення класів: ключове слово class, поля класу, конструктор класу та методи класу. Модифікатори доступу класу. Створення об'єктів: оголошення об'єктної змінної, створення об'єкта за допомогою оператора new та повернення посилання на створений об'єкт. Конструктор класу. Метод класу. Ключові слова void. Застосування ключового слова this. Типізовані та нетипізовані методи. Точковий синтаксис звернення до поля та методу об'єкта. Ініціалізатори та їх відмінність від конструкторів.

Тема 8. Робота з тестом та інші утиліти.

Особливості створення строкових даних у мові Java. Доповнення строкових даних класами StringBuffer та StringBuilder та їх основні методи. Декілька способів створення строкових даних. Пул строкових даних та правила його застосування. Методи обробки текстових рядків типові приклади. Розширення методів обробки примітивних типів даних за допомогою класів-обгортки (wrapper). Синтаксичний розбір строкових даних методами класу StringTokenizer.

Тема 9. Регулярні вирази в Java.

Регулярні вирази – як спосіб описи набору рядків з загальними характеристиками. Аналогія з файловими шаблонами. Тестова програма для аналізу синтаксису регулярних виразів. Особливості індексації символів у текстовому рядку. Три основні класу обробки регулярних виразів з пакету java.util.regex. Побудова перших примітивних регулярних виразів. Основні методи класу Matcher. Керуючі символи (метасимволи) у рядку регулярного виразу. Способи визначення класів символів. Способи завдання логічних операцій. Квантифікатори. Оголошення деяких категорій символів. Приклади побудови типових регулярних виразів. Типові приклади запису квантифікаторів.

Тема 10. Особливості роботи з об'єктами.

Перевантаження методів і конструкторів – створення декілька методів і конструкторів з однаковими ім'ям. Забезпечує поліморфізм, оскільки це один із способів реалізації концепції «один інтерфейс, декілька методів». Створення копій об'єкта. Оголошення загальних (статичних) полів і методів. Застосування ключового поля final для створення констант. Правило побудови найменування констант та особливість їх застосування. Модифікатори доступу до даних та методів класу, області дії модифікаторів. Вкладені класи. Два типи вкладених класів – статичні та нестатичні. Їх особливість та область застосування. Параметри та аргументи класів. Створення аргументів змінної довжини.

Тема 11. Успадкування.

Успадкування – фундаментальний механізм ООП. Створення підкласу. Відкрити та закрити члени класу. Створення конструктора суперкласу та підкласу. Ключове слово `super` та дві його форми. Перевизначені методи та їх відмінність від перевантаження методів. Закриті члени класу (поля та методи), які об'явлені у суперкласі з ключовим словом `private`. Об'єктна змінна суперкласу. Абстрактні класи та методи. Використання ключового слова `final` для блокування перевизначення методу та наслідування класу. Огляд методів спеціального суперкласу `Object`.

Тема 12. Інтерфейси та пакети.

Пакети – контейнери класів. Використовується для забезпечення ізоляції простору імен класу. Створення та посилання на пакети. Створення багато рівного доступу між пакетами. Приклади використання всіх комбінацій модифікаторів управління доступом. Інтерфейс – конструкція в якій можуть описуватись тільки абстрактні методи та статичні константи. Призначення та реалізація інтерфейсів. Ключове слово `interface` – абстрагує інтерфейс класу від його реалізації. Оголошення інтерфейсу. Імплементация інтерфейсу у класі. Особливості реалізації множеного наслідування у класах. Вкладенні інтерфейси та їх особливість. Приклади використання інтерфейсів. Відмінності між інтерфейсами та абстрактними класами.

Тема 13. Обробка виняткових ситуацій.

Виключення – нештатні ситуації. Опис процесів, які реалізуються алгоритмами Java при виникненні помилок. Система вбудованих класів обробки виняткових ситуацій. Підклас `Error`. Підклас `Exception`. Огляд класів виключень - неконтрольовані та контрольовані виключення. Обробка виключень оператором `try – catch`. Створення призначених для користувача класів. Ключове слова `throw`. способи програмування виключень у програмі.

Тема 14. Потоки даних введення та виводу на консоль і файловою системою.

Потоки введення і виведення. Байтові та символні потоки. Класи байтових та символних потоків. Читання консольного введення. Клас `Console` та його основні методи. Типовий приклад застосування класу `Console` для введення базових типів даних. Робота з файлами, які містять байтові та символні дані. Використання системи NIO (New I/O) для операцій з файловою системою. Отримання інформації про шляхи до файлів. Сериалізація – перетворення об'єктів у потоки байтів для передачі.

Перелік навчальних робіт студентів та оцінки їх у балах з дисципліни «Технологія Java»

Види робіт	К-сть балів
Практичне заняття №1. Загальні відомості про систему програмування Eclipse	5
Практичне заняття №2. Програмування розгалужених та циклічних алгоритмів	5
Практичне заняття №3. Обробка даних в однорідних структурах - масивах	5
Практичне заняття №4. Гетери (<code>get</code>) і сетери (<code>set</code>), методи які забезпечують безпеку полів об'єктів	7
Практичне заняття №5. Робота із змінними та незмінними текстовими рядками	7
Практичне заняття №6. Використання регулярних виразів у Java додатках	7
Практичне заняття №7. Клас <code>OBJECT</code> та його методи <code>toString()</code> , <code>equals()</code> та <code>hashCode()</code>	7

Практичне заняття №8. Абстрактні класи. Спадкування, поліморфізм та інтерфейси	7
Модульний контроль	20
Виконання індивідуального завдання (СР)	30
Разом: Аудиторна робота	70
Самостійна робота (СР)	30
Всього:	100

КОНТРОЛЬ ТА КРИТЕРІЇ ОЦІНЮВАННЯ ЗНАТЬ СТУДЕНТІВ

При вивченні дисципліни використовуються наступні форми контролю знань студентів: поточний; модульний; підсумковий.

Поточний контроль передбачає перевірку теоретичних питань, самостійної роботи, практичних робіт та усне опитування по кожній практичній роботі. По даному виду контролю оцінювання знань здійснюється у відповідності до бального розподілу наведеного в попередній таблиці.

Модульний контроль передбачає виконання модульної контрольної роботи. Всі завдання оцінюються в 20 балів. Перше завдання (теоретичне) – 4 бали, друге завдання (практичне) – 8 балів, третє завдання (практичне) – 8 балів.

Формою підсумкового контролю є екзамен. Екзаменаційна оцінка (100 балів) є результатом виконання двох теоретичних питань (2 x 20 балів = 40 балів) та практичного завдання (60 балів).

Результуюча оцінка з дисципліни визначається як середня від балів набраних протягом семестру та отриманих на іспиті.

СПИСОК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ

Основний:

1. Кунгурце А. Б. Основи програмування на мові Java. Середовище Net Beans. Навч. Посібник для студентів вищих навчальних закладів / А. Б. Кунгурцев, Т. В. Ковалюк. – Одеса, 2016. – 183 с.
2. Мельник Р. А. Технологія Java Програмування для Інтернету на основі Java-технології: Навч. посібник для студентів напряму «Комп'ютерні науки». / Р. А. Мельник. – Л. : Львівська політехніка, 2013. – 110 с.
3. Ткаченко О. М. Об'єктно-орієнтоване програмування мовою Java. Навчальний посібник. / О. М. Ткаченко. – Вінниця: ВНТУ, 2016. – 107 с.
4. Троян С.О. Програмування мовою Java: навч. посіб. / С. О. Троян. – Умань: ФОП Жовтий О. О., 2017. – 132 с.
5. Спирінцева О. В. Java-технології та мобільні пристрої. Алгоритми і структури даних: навчальний посібник / О. В. Спирінцева, О. А. Литвинов, В. В. Герасимов. – Д.: Вид-во ДНУ ім. О. Гончара, 2016. 140 с.