



ПРОГРАМУВАННЯ. ЧАСТИНА 2. ОСНОВИ ВЕБПРОГРАМУВАННЯ ТА РОЗРОБЛЕННЯ МОБІЛЬНИХ ЗАСТОСУНКІВ

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Перший (бакалаврський)</i>
Галузь знань	<i>12 Інформаційні технології</i>
Спеціальність	<i>121 Інженерія програмного забезпечення</i>
Освітня програма	<i>Інженерія програмного забезпечення мультимедійних та інформаційно-пошукових систем</i>
Статус дисципліни	<i>Нормативна</i>
Форма навчання	<i>Очна (денна)</i>
Рік підготовки, семестр	<i>2 рік підготовки, 4 семестр</i>
Обсяг дисципліни	<i>Лекції: 36 год., комп'ютерний практикум: 18 год., самостійна робота: 51 год.</i>
Семестровий контроль/ контрольні заходи	<i>Залік, модульна контрольна робота, календарний контроль</i>
Розклад занять	<i>Згідно розкладу на осінній семестр поточного навчального року (http://roz.kpi.ua/)</i>
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	<i>Керівник курсу: к.т.н., доцент, Іщеряков Сергій Михайлович, ismismif@gmail.com До викладання залучені фахівці IT-компанії «ЕПАМ Системз»</i>
Розміщення курсу	<i>Google classroom: https://classroom.google.com/c/NTg4Nzc1NDQ5ODEy?cjc=uzxm473</i>

Програма навчальної дисципліни

1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

Вивчення дисципліни «Програмування. Частина 2. Основи вебпрограмування та розроблення мобільних застосунків» дозволяє сформувати у здобувачів освіти компетенції, необхідні для розв'язання практичних задач професійної діяльності, пов'язаної із розробленням програмного забезпечення для мережевих та мобільних систем.

Метою вивчення дисципліни «Програмування. Частина 2. Основи вебпрограмування та розроблення мобільних застосунків» є формування у здобувачів освіти здатностей самостійно розробляти програмне забезпечення для створення високопродуктивних алгоритмів та засобів опрацювання великих обсягів даних в мережевих системах, а також мобільних застосунків на основі операційної системи Android.

Предметом дисципліни «Програмування. Частина 2. Основи вебпрограмування та розроблення мобільних застосунків» є технології розроблення засобами мови Java програмних продуктів для мережесистем та мобільних систем.

Вивчення дисципліни «Програмування. Частина 2. Основи веб-програмування та розроблення мобільних застосунків» формує у здобувачів освіти **фахові компетентності (ФК)**, необхідні для розв'язання практичних задач професійної діяльності, пов'язаних з розробленням, вдосконаленням та супроводженням інтелектуальних інформаційних систем оброблення мультимедійних даних:

ФК02 Здатність брати участь у проєктуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.

ФК03 Здатність розробляти архітектуру, модулі та компоненти програмних систем.

ФК08 Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення.

ФК13 Здатність обґрунтовано обирати та освоювати інструментарій з розроблення та супроводження програмного забезпечення.

ФК14 Здатність до алгоритмічного та логічного мислення.

ФК17 Здатність розробляти програмне забезпечення інформаційно-пошукових систем.

ФК19 Здатність розробляти програмне забезпечення мультимедійних та мультимедійних систем.

Програмних результатів навчання (ПРН) за освітньою програмою:

ПРН01 Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки.

ПРН03 Знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення;

ПРН04 Знати і застосовувати професійні стандарти і інші нормативно-правові документи в галузі інженерії програмного забезпечення;

ПРН06 Уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення.

ПРН07 Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.

ПРН08 Знати та вміти розробляти людино-машинний інтерфейс.

ПРН09 Вміти використовувати методи та засоби збору, формулювання та аналізу вимог до програмного забезпечення.

ПРН12 Застосовувати на практиці ефективні підходи щодо проєктування програмного забезпечення.

ПРН13 Знати і застосовувати методи розроблення алгоритмів, конструювання програмного забезпечення та структур даних і знань;

ПРН15 Мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення;

ПРН23 Вміти документувати та презентувати результати розроблення програмного забезпечення;

ПРН38 Вміти застосовувати технології програмування для розроблення програмного забезпечення мультимедійних та інформаційно-пошукових систем.

2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)

Успішному вивченню дисципліни «Програмування. Частина 2. Основи вебпрограмування та розроблення мобільних застосунків» передують вивчення дисциплін «Програмування. Частина 1.

Об'єктно-орієнтоване програмування та шаблони проектування», «Основи комп'ютерних систем і мереж», «Бази даних», «Основи програмування», «Алгоритми та структури даних», «Компоненти програмної інженерії» навчального плану підготовки бакалаврів за спеціальністю 121 Інженерія програмного забезпечення.

Отримані при засвоєнні дисципліни «Програмування. Частина 2. Основи вебпрограмування та розроблення мобільних застосунків» теоретичні знання та практичні уміння забезпечують успішне вивчення дисциплін «Програмне забезпечення мультимедійних систем», «Стандартизація та технології розроблення мультимедійних та інформаційно-пошукових програмних продуктів», «Програмне забезпечення інформаційно-пошукових систем», проходження переддипломної практики, виконання курсових проєктів та дипломних проєктів за спеціальністю 121 Інженерія програмного забезпечення.

3. Зміст навчальної дисципліни

Дисципліна «Програмування. Частина 2. Основи вебпрограмування та розроблення мобільних застосунків» передбачає вивчення таких тем:

Тема 1. Об'єктне проектування засобами мови Java

Тема 2. Багатопотокове та паралельне програмування засобами мови Java

Тема 3. Веб-програмування засобами мови Java

Тема 4. Розроблення мобільних застосунків для ОС Android

Залік

4. Навчальні матеріали та ресурси

Базова література:

1. Іщеряков С.М. Матеріали курсу «Основи вебпрограмування та розроблення мобільних застосунків». Надаються зареєстрованим студентам.

Додаткова література:

2. Herbert Schildt. Java: The Complete Reference, Twelfth Edition: McGraw Hill, 2021.- 1280 p.
3. Cay S. Horstmann. Core Java, Volume I—Fundamentals, Eleventh Edition: Pearson, 2018. – 928 p.
4. Cay S. Horstmann. Core Java, Volume I I—Advanced Features, Eleventh Edition: Pearson, 2019. – 1040 p.
5. Richard Warburton. Java 8 Lambdas: O'Reilly Media, Inc, 2014. -168 p.
6. Brian Goetz, Tim Peierls, Joshua Bloch, Joseph Bowbeer, David Holmes, Doug Lea. Java Concurrency in Practice: Addison-Wesley Professional, 2006. -432 p.
7. Bryan Sills, Brian Gardner, Kristin Marsicano, Chris Stewart. Android Programming: The Big Nerd Ranch Guide, 5th Edition: Addison-Wesley Professional, 2022. – 688 p.
8. Reto Meier. Professional Android 4 Application Development, 3rd edition: Wrox, 2012. – 864 p.
9. S.G.Ganesh, Tushar Sharma. Java SE7 Programmer Exams 1Z0-804 and 1Z0-805. A Comprehensive OCPJP 7 Certification Guide.– APRESS,New York,2012.-644 p.

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

№ з/п	Тип навчального заняття	Опис навчального заняття
<i>Тема 1. Об'єктне проектування засобами мови Java</i>		
1	Лекція 1. Реалізація алгоритмів. Основні оператори мови Java.	Операції порівняння. Тип даних <i>boolean</i> . Оператори розгалуження <i>if</i> та <i>if-else</i> . Цикли. Оператор <i>for</i> . Область видимості змінної. Вкладені цикли.

		<p>Оператори <i>while</i> та <i>do-while</i>. Види масивів, способи їх опису. Одновимірні масиви. Поле <i>length</i>. Оператори ділення за модулем <i>%</i>, інкремента, декремента, операцій з присвоєнням, тернарний оператор. Логічна функція <i>XOR</i>, <i>short-circuit</i> логічні функції. Оператори <i>switch-case</i>, <i>break</i>, <i>continue</i>, <i>foreach</i>.</p> <p>Завдання на СРС: п. 6 №1.</p>
2	Лекція 2. Інкапсуляція, шаблон MODEL-VIEW-CONTROLLER.	<p>Ознайомлення із Java API, пакетом <i>java.lang</i>, поняттями перезавантаження методів, імпорт, ключовим словом <i>static</i> та класами <i>java.util.Arrays</i>, <i>java.util.ArrayList</i>. Поняття інкапсуляції, структура класу. Створення власних класів. Оголошення та ініціалізація полів. Оголошення методів. Оголошення та створення об'єктів. Оператор <i>new</i>. Літерал <i>null</i>. Архітектурний шаблон MODEL-VIEW-CONTROLLER (MVC) для побудови власних класів. Ознайомлення із конструкторами, передаванням в метод посилань на об'єкти.</p> <p>Завдання на СРС: п.6 №2.</p>
3	Комп'ютерний практикум 1 (частина 1). Використання шаблону MODEL-VIEW-CONTROLLER для побудови класів та реалізації алгоритмів.	<p>Завдання: розробити Java-код для визначення типу перетину прямокутників та для виведення крапкових геометричних фігур на основі шаблону MODEL-VIEW-CONTROLLER.</p> <p>Завдання на СРС: п.6 №3.</p>
4	Лекція 3. Основні поняття та синтаксичні інструменти мови Java.	<p>Ключове слово <i>static</i>. Локальні змінні, змінні об'єкту та класу. Особливості передавання в метод в якості аргументів примітивів та посилань. Типи методів. Перезавантажені конструктори. Перезавантаження методів. Метод <i>main()</i>. Аргументи в методі <i>main()</i>. Ініціалізація змінних. Блоки ініціалізації. Порядок ініціалізації. Пакети. <i>Import</i>. <i>Static import</i>. Клас <i>Math</i>.</p> <p>Завдання на СРС: п.6 №4.</p>
5	Лекція 4. Наслідування.	<p>Поняття наслідування. Відношення <i>has-a</i>, <i>is-a</i>. UML-діаграми. Класи <i>Calendar</i>, <i>GregorianCalendar</i>, <i>Date</i>. Модифікатори доступу. Ключові слова <i>this</i>, <i>super</i>. Клас <i>Object</i>. Перевизначення методів. Метод <i>toString()</i>.</p> <p>Завдання на СРС: п.6 №5.</p>
6	Комп'ютерний практикум 1 (частина 2). Використання наслідування для побудови ієрархічних структур класів.	<p>Завдання: Для ієрархії класів: Транспортний засіб => Літак, Автомобіль, Корабель; Літак => Пасажирський літак, Транспортний літак, Винищувач, Автомобіль => Легковий автомобіль, Автобус, Вантажний автомобіль, Корабель => Пасажирський лайнер, Буксир, Танкер – оголосити в кожному класі 1-3 найбільш доцільних поля, в кожному класі ввести методи <i>getters/setters</i> для всіх</p>

		<p>полів класу. Полям всіх класів надати доступ рівня <code>private</code>, методам <code>getters/setters</code> – доступ рівня <code>public</code> для всіх класів, крім класу <code>Fighter</code>, в якому змінювати значення полів можуть тільки класи одного пакету, а читати значення полів – також і підкласи класу <code>Fighter</code> з інших пакетів, в кожному класі ввести конструктор для ініціалізації всіх полів класу та метод <code>toString()</code>.</p> <p>Завдання на СРС: п.6 №6.</p>
7	Лекція 5. Поліморфізм.	<p>Поліморфні посилання. Поліморфізм. Метод <code>equals()</code>. <code>Covariant returns</code>. Приведення типів. Оператор <code>instanceof</code>. Ключове слово <code>final</code>.</p> <p>Завдання на СРС: п.6 №7.</p>
8	Лекція 6. Інтерфейси.	<p>Абстрактні класи. Інтерфейси. Інтерфейси <code>Comparable</code>, <code>Comparator</code>. Внутрішні класи (регулярні, локальні, анонімні, статичні). Функціональні інтерфейси та лямбда-вирази.</p> <p>Завдання на СРС: п.6 №8.</p>
9	Комп'ютерний практикум 2 (частина 1). Використання поліморфізму в ієрархічних структурах класів.	<p>Для завдання формування крапкових геометричних фігур розробити ефективну ієрархію класів <code>Shape</code> із перевизначеними методами.</p> <p>Завдання на СРС: п.6 №9.</p>
10	Лекція 7. Обробка виняткових ситуацій. Робота з текстовими файлами.	<p><code>Exceptions</code>. Робота з текстовими файлами.</p> <p>Завдання на СРС: п.6 №10.</p>
11	Лекція 8. Особливості деяких стандартних класів Java API.	<p><code>Strings</code>, <code>StringBuffers</code>, <code>StringBuilders</code>. <code>Wrappers</code>, <code>autoboxing</code>, <code>autounboxing</code>. <code>Enums</code>. Клас <code>Arrays</code>, <code>Var-args</code>, <code>Garbage collector</code>, API Java SE8 для роботи із датами та часом. Класи пакету <code>java.time</code>: <code>LocalDate</code>, <code>LocalTime</code>, <code>LocalDateTime</code>, <code>Period</code>. Клас <code>java.time.format.DateTimeFormatter</code>.</p> <p>Завдання на СРС: п.6 №11.</p>
12	Комп'ютерний практикум 2 (частина 2). Робота з текстовими файлами.	<p>Завдання: Прочитати та обробити дані з трьох існуючих файлів та занести оброблені дані у файли, створені засобами Java.</p> <p>Завдання на СРС: п.6 №12.</p>
<p>Тема 2. Багатопотокове та паралельне програмування засобами мови Java</p>		
13	Лекція 9. Алгоритми.	<p>Опрацювання текстових даних (<code>parsing</code>) на основі технології <code>regex</code>. Серіалізація файлів. Інтерфейси <code>Serializable</code> та <code>Externalizable</code>. Особливості композиції та наслідування при серіалізації.</p> <p>Завдання на СРС: п.6 №13.</p>

14	Лекція 10. Контейнери.	<p>Фреймворк колекцій. Принципи побудови hash-таблиць. Інтерфейси Set, Queue, Map, Iterator. Класи TreeSet, PriorityQueue, Dequeue, TreeMap, HashMap. Побудова та використання generic-класів та generic-методів. Особливості поліморфізму при використанні generic-колекцій.</p> <p>Завдання на СРС: п.6 №14.</p>
15	Комп'ютерний практикум 3 (частина 1). Використання особливостей колекцій в комплексних алгоритмічних задачах.	<p>Завдання: На площині задано N точок. Необхідно у файл LINES записати два HashMap:</p> <ul style="list-style-type: none"> - із ключами у вигляді об'єктів класу Point, що визначаються цілими координатами точок, та значеннями у вигляді кількості прямих, що проходять через цю точку та, щонайменше, ще через одну точку, - із ключами у вигляді об'єктів класу Line, що визначаються параметрами K та B прямої $y = K \cdot x + B$, та значеннями у вигляді кількості точок, що належать цій прямій. <p>Завдання на СРС: п.6 №15.</p>
16	Лекція 11. Багатопотокове програмування на основі технології Threads.	<p>Основи створення та використання потоків. Клас Thread. Інтерфейси Runnable, Callable, Executor, ExecutorService. Основи синхронізації потоків. Ключове слово synchronized. Синхронізація статичних методів. Пакет java.util.concurrent.atomic. Реалізація синхронізації на основі ресурсів пакету java.util.concurrent. Потокобезпечні колекції. Реалізація синхронізації на основі ресурсів пакету java.util.concurrent.lock. Взаємодія потоків. Методи wait, notify класу Object. Поняття взаємного блокування (deadlock).</p> <p>Завдання на СРС: п.6 №16.</p>
17	Лекція 12. Фреймворк API Stream.	<p>Проміжні та термінальні методи інтерфейсу Stream для формування та опрацювання потоків. Клас Optional. Особливості технології потокового зведення. Метод reduce. Методи класу Collectors. Розпаралелювання потоків. Порівняльний аналіз послідовних та паралельних потоків, визначення доцільності розпаралелювання потоків.</p> <p>Завдання на СРС: п.6 №17.</p>
18	Комп'ютерний практикум 3 (частина 2). Побудова багатопотокових паралельних програмних систем.	<p>Завдання: Пасажирів, прибувших на трьох (чотирьох, п'яти) літаках одночасно, розвозять мікроавтобуси. Завантаженість літаків – рівно 100 пасажирів. Мікроавтобуси мають місткість 6, або 7, або 8 пасажирів і їдуть до 4 різних міст. Завантаженість мікроавтобусів повинна бути 100%, за винятком мікроавтобусів з останніми пасажирами. Пасажири їдуть сім'ями від 1 до 4 осіб. Розділяти сім'ї по мікроавтобусах не можна. Зробити спочатку однопотокову систему, потім – максимально</p>

		паралелізовану синхронну багатопотокову, із синхронізацією по літаках або по автобусах. Завдання на СРС: п.6 №18.
<i>Тема 3. Веб-програмування засобами мови Java</i>		
19	<i>Лекція 13. Сервлети та JSP.</i>	<i>Інтерфейси ServletContext, ServletConfig, ServletRequest, HttpServletRequest, ServletResponse, HttpServletResponse. Опрацювання запитів. Стандартні елементи action. JSP-документ. Бібліотека тегів JSTL. Expression Language.</i> <i>Завдання на СРС: п.6 №19.</i>
20	<i>Лекція 14. Технологія JDBC для роботи з базами даних.</i>	<i>Драйвери та конфігурування JDBC. Інтерфейси та класи пакету java.sql. Оператори запитів.</i> <i>Завдання на СРС: п.6 №20.</i>
21	<i>Комп'ютерний практикум 4. Робота з віддаленими базами даних.</i>	<i>Завдання: Прочитати та занести у відповідні колекції дані з віддаленої бази у форматі:</i> <i>N Bottle Volume Material</i> <i>1 Wine 0.75 Glass</i> <i>2 Juice 0.25 Metal</i> <i>Завдання для формування запитів:</i> <i>- зчитувати пляшки місткістю: не більше 0.5 л, в межах від 0.51л до 0.99 л, не меншою за 1.0 л,</i> <i>- зчитувати пляшки для: вина, соків, води,</i> <i>- зчитувати пляшки, зроблені з: металу, скла, пластику.</i> <i>Занести до бази дані з текстових файлів.</i> <i>Завдання на СРС: п.6 №21.</i>
<i>Тема 4. Розроблення мобільних застосунків для ОС Android</i>		
22	<i>Лекція 15. Побудова інтерфейсу користувача. Базові віджети</i>	<i>Структура Андроїд-проекту. Мова XML. Віджет TextView. Класи Activity, Application. Ресурси. Робота із зображеннями. Віджет ImageView. Поняття розмітки (менеджеру компоновки). Керування розмітками. Розмітка ConstraintLayout. Введення текстів. Смуги прокрутки текстів. Віджети EditText, ScrollView. Робота з кнопками. Віджети Button, RadioButton, ToggleButton, ImageButton. Індикатори. Віджет ProgressBar. Робота з меню. Інтерфейс Menu, класи ContextMenu, SubMenu. Повідомлення. Віджет Toast. Створення нестандартних повідомлень. Діалогові вікна з кнопками. Віджет AlertDialog. Діалогове вікно з індикаторами прогресу. Віджет ProgressDialog.</i> <i>Завдання на СРС: п.6 №22.</i>
23	<i>Лекція 16. Навігація. Фрагменти.</i>	<i>Принципи навігації. Клас Intent. Навігація без передавання даних. Навігація із передаванням даних.</i>

		Фрагменти. Класи <i>Fragment</i> , <i>FragmentActivity</i> та <i>FragmentManager</i> . Завдання на СРС: п.6 №23.
24	Комп'ютерний практикум 5 (частина 1). Розробка багатовіконних <i>Android</i> -застосунків.	Завдання: із використанням технології фрагментів розробити двовіконний калькулятор для виконання арифметичних (перше вікно) та алгебраїчних (наприклад, піднесення до степеню, друге вікно) функцій. Завдання на СРС: п.6 №24.
25	Лекція 17. Адаптери	Призначення та типи адаптерів. Класи <i>ListView</i> та <i>ArrayAdapter</i> . Сумісне використання адаптерів та колекцій на базі інтерфейсу <i>Map</i> . Адаптери для складених колекцій. Клас <i>BaseAdapter</i> . Завдання на СРС: п.6 №25.
26	Комп'ютерний практикум 5 (частина 2). Використання колекцій в <i>Android</i> -застосунках.	Завдання: Розробити телефонну книгу із можливістю видалення та редагування адресатів. Завдання на СРС: п.6 №26.

6. Самостійна робота студента/аспіранта

Дисципліна «Програмування. Частина 2. Основи вебпрограмування та розроблення мобільних застосунків» ґрунтується на самостійних підготовках до аудиторних занять на теоретичні та практичні теми.

№ з/п	Назва теми, що виноситься на самостійне опрацювання	Кількість годин	Література
1	Підготовка до лекції 1	1	1
2	Підготовка до лекції 2	1	1
3	Підготовка до комп'ютерного практикуму 1 (частина 1)	3	1
4	Підготовка до лекції 3	1	1
5	Підготовка до лекції 4	1	1
6	Підготовка до комп'ютерного практикуму 1 (частина 2)	3	1
7	Підготовка до лекції 5	1	1
8	Підготовка до лекції 6	1	1
9	Підготовка до комп'ютерного практикуму 2 (частина 1)	3	1
10	Підготовка до лекції 7	1	1
11	Підготовка до лекції 8	1	1
12	Підготовка до комп'ютерного практикуму 2 (частина 2)	3	1
13	Підготовка до лекції 9	1	1
14	Підготовка до лекції 10	1	1
15	Підготовка до комп'ютерного практикуму 3 (частина 1)	3	1
16	Підготовка до лекції 11	1	1

17	Підготовка до лекції 12	1	1
18	Підготовка до комп'ютерного практикуму 3 (частина 2)	3	1
19	Підготовка до лекції 13	1	1
20	Підготовка до лекції 14	1	1
21	Підготовка до комп'ютерного практикуму 4	3	1
22	Підготовка до лекції 15	1	1
23	Підготовка до лекції 16	1	1
24	Підготовка до комп'ютерного практикуму 5 (частина 1)	3	1
25	Підготовка до лекції 17	1	1
26	Підготовка до комп'ютерного практикуму 5 (частина 2)	3	1
27	Підготовка до МКР	4	1
28	Підготовка до заліку	3	1

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

Відвідування лекційних занять є обов'язковим.

Відвідування занять комп'ютерного практикуму може бути епізодичним та за потреби консультації/захисту робіт комп'ютерного практикуму.

Правила поведінки на заняттях: активність, повага до присутніх, відключення телефонів.

Дотримання політики академічної доброчесності.

Правила захисту робіт комп'ютерного практикуму: роботи повинні бути зроблені відповідно до поставлених задач.

8. Види контролю та рейтингова система оцінювання результатів навчання (PCO)

Протягом семестру студенти виконують 5 комп'ютерних практикумів. Максимальна кількість балів за кожний комп'ютерний практикум: 15 балів.

Бали нараховуються за:

- якість виконання комп'ютерного практикуму: 0-10 балів;*
- відповідь під час захисту комп'ютерного практикуму: 0-4 балів;*
- своєчасне представлення роботи до захисту: 0-1 бали.*

Критерії оцінювання якості виконання:

- 9-10 балів – робота виконана якісно, в повному обсязі;*
- 7-8 балів – робота виконана якісно, в повному обсязі, але має недоліки;*
- 5-6 балів – робота виконана в повному обсязі, але містить незначні помилки;*
- 1-4 бали – робота виконана в повному обсязі, але містить суттєві помилки;*
- 0 балів – робота виконана не в повному обсязі.*

Критерії оцінювання відповіді:

- 5-6 балів – відповідь повна, добре аргументована;*
- 3-4 бали – відповідь вірна, але має недоліки або незначні помилки;*
- 1-2 бали – у відповіді є суттєві помилки;*
- 0 балів – немає відповіді або відповідь невірна.*

Критерії оцінювання своєчасності представлення роботи до захисту:

4 бали – робота представлена до захисту не пізніше вказаного терміну;
0 балів – робота представлена до захисту пізніше вказаного терміну.

Максимальна кількість балів за виконання та захист комп'ютерних практикумів:
 $R_c = 15 \text{ балів} \times 5 \text{ комп. практи.} = 75 \text{ балів.}$

Модульна контрольна робота складається з 1 теоретичного питання та 1 практичного завдання. Максимальна кількість балів за теоретичне запитання 10 балів.
Критерії оцінювання теоретичного запитання:

Критерії оцінювання практичного завдання:
10 балів – відповідь вірна, повна, добре аргументована;
6-9 балів – в цілому відповідь вірна, але має недоліки;
1-5 балів – у відповіді є суттєві помилки;
0 балів – немає відповіді або відповідь невірна.

Максимальна кількість балів за практичне завдання 15 балів.
Критерії оцінювання практичного завдання:
14–15 балів – відповідь вірна;
12–13 балів – в цілому відповідь вірна, але має недоліки;
9–11 балів – у відповіді є суттєві помилки;
0 балів – немає відповіді або відповідь невірна.

$R = R_c = R_{\text{комп. практи.}} + R_{\text{МКР}} = 75 \text{ балів} + 25 \text{ балів} = 100 \text{ балів.}$

Семестровий контроль: залік

Умови допуску до семестрового контролю: при семестровому рейтингу (R_c) не менше 60 балів та зарахуванні усіх робіт комп'ютерного практикуму, студент отримує залік «автоматом» відповідно до таблиці (Таблиця відповідності рейтингових балів оцінкам за університетською шкалою). В іншому разі він має виконувати залікову контрольну роботу. Необхідною умовою допуску до залікової контрольної роботи є виконання і захист комп'ютерного практикуму. Якщо студент не погоджується з оцінкою «автоматом», то може спробувати підвищити свою оцінку шляхом написання залікової контрольної роботи, при цьому його бали, отримані за семестр, зберігаються, а з двох отриманих студентом оцінок виставляється краща («м'яка» система оцінювання).

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

Кількість балів	Оцінка
100-95	Відмінно
94-85	Дуже добре
84-75	Добре
74-65	Задовільно
64-60	Достатньо
Менше 60	Незадовільно
Не виконані умови допуску	Не допущено

9. Додаткова інформація з дисципліни (освітнього компонента)

Перелік питань, які виносяться на семестровий контроль, наведено у Додатку 1.

Робочу програму навчальної дисципліни (силабус):

Складено к.т.н., доцент, Іщеряков С.М.

Ухвалено кафедрою ПЗКС (протокол № 12 від 26.04.2023 р.)

Погоджено Методичною комісією факультету прикладної математики (протокол № 10 від 26.05.2023 р.)

Додаток 1. Перелік питань, які виносяться на семестровий контроль

1. Реалізація алгоритмів. Основні оператори мови Java.
2. Інкапсуляція.
3. Шаблон MODEL-VIEW-CONTROLLER.
4. Класи *Calendar*, *GregorianCalendar*, *Date*.
5. UML-діаграми.
6. Поняття наслідування. Відношення *has-a*, *is-a*. Метод *toString()*.
7. Модифікатори доступу.
8. Ключове слово *this*.
9. Ключове слово *super*.
10. Метод *toString()*.
11. Ключове слово *static*.
12. Особливості передавання в метод в якості аргументів примітивів та посилань.
13. Перезавантаження методів.
14. Метод *main()*. Аргументи в методі *main()*.
15. Ініціалізація змінних. Блоки ініціалізації. Порядок ініціалізації.
16. Пакети. *Import*. *Static import*.
17. Клас *Math*.
18. Поліморфні посилання. Поліморфізм.
19. Метод *equals()*.
20. *Covariant returns*.
21. Приведення типів. Оператор *instanceof*.
22. Ключове слово *final*.
23. Абстрактні класи.
24. Інтерфейси.
25. Інтерфейси *Comparable*, *Comparator*.
26. Внутрішні класи (регулярні, локальні, анонімні, статичні).
27. Функціональні інтерфейси та лямбда-вирази.
28. Обробка виняткових ситуацій.
29. Робота з текстовими файлами.
30. *Strings*, *StringBuffers*, *StringBuilders*.
31. *Wrappers*, *autoboxing*, *autounboxing*.
32. *Enums*.
33. Клас *Arrays*.
34. *Var-args*.
35. *Garbage collector*.
36. Класи пакету *java.time*: *LocalDate*, *LocalTime*, *LocalDateTime*, *Period*.
37. Клас *java.time.format.DateTimeFormatter*.
38. Опрацювання текстових даних (*parsing*) на основі технології *regex*.
39. Сериалізація файлів. Інтерфейси *Serializable* та *Externalizable*.
40. Фреймворк колекцій. Інтерфейси *Set*, *Queue*, *Iterator*. Класи *TreeSet*, *PriorityQueue*, *Deque*.
41. Принципи побудови *hash*-таблиць. Інтерфейс *Map*. Класи *TreeMap*, *HashMap*.
42. Побудова та використання *generic*-класів та *generic*-методів.
43. Особливості поліморфізму при використанні *generic*-колекцій.
44. Основи створення та використання потоків. Клас *Thread*. Інтерфейси *Runnable*, *Callable*, *Executor*, *ExecutorService*.
45. Основи синхронізації потоків. Ключове слово *synchronized*. Синхронізація статичних методів.
46. Пакет *java.util.concurrent.atomic*.
47. Реалізація синхронізації на основі ресурсів пакету *java.util.concurrent*. Потокобезпечні колекції.
48. Реалізація синхронізації на основі ресурсів пакету *java.util.concurrent.lock*.
49. Взаємодія потоків. Методи *wait*, *notify*. Поняття взаємного блокування (*deadlock*).

50. Проміжні та термінальні методи інтерфейсу *Stream*
51. Клас *Optional*.
52. Особливості технології потокового зведення. Метод *reduce*.
53. Методи класу *Collectors*.
54. Розпаралелювання потоків. Порівняльний аналіз послідовних та паралельних потоків.
55. Інтерфейси *ServletContext*, *ServletConfig*, *ServletRequest*, *HttpServletRequest*, *ServletResponse*, *HttpServletResponse*. Опрацювання запитів.
56. Стандартні елементи *action*. JSP-документ. Бібліотека тегів *JSTL*. *Expression Language*.
57. Принципи навігації. Клас *Intent*. Навігація.
58. Фрагменти. Класи *Fragment*, *FragmentActivity* та *FragmentManager*.
59. Призначення та типи адаптерів. Класи *ListView* та *ArrayAdapter*.
60. Сумісне використання адаптерів та колекцій на базі інтерфейсу *Map*.
61. Адаптери для складених колекцій. Клас *BaseAdapter*.
62. Введення текстів. Віджет *TextView*. Смуги прокрутки текстів. Віджети *EditText*, *ScrollView*.
63. Класи *Activity*, *Application*. Ресурси.
64. Робота із зображеннями. Віджет *ImageView*.
65. Поняття менеджера компоновки. Керування розмітками. Розмітка *ConstraintLayout*.
66. Робота з кнопками. Віджети *Button*, *RadioButton*, *ToggleButton*, *ImageButton*.
67. Індикатори. Віджет *ProgressBar*.
68. Робота з меню. Інтерфейс *Menu*, класи *ContextMenu*, *SubMenu*.
69. Повідомлення. Віджет *Toast*. Створення нестандартних повідомлень.
70. Діалогові вікна з кнопками. Віджет *AlertDialog*.
72. Діалогове вікно з індикаторами прогресу. Віджет *ProgressDialog*.