

## СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ «Прикладне програмування Java»

|   |  |       |   |                        |  |                        |   |  |
|---|--|-------|---|------------------------|--|------------------------|---|--|
| <b>Лектор курсу</b>   |  |       | Ільїн Олег Олександрович,<br>доктор технічних наук, професор<br>кафедри комп'ютерних наук   |                        | <b>Контактна інформація<br/>лектора (e-mail),<br/>сторінка курсу в<br/>Moodle</b>  |                        | e-mail: oleg.ilin.ua@gmail.com;<br><b>сторінка курсу в Moodle</b> –<br>http://dl.dut.edu.ua/course/view.php<br>?id=1373 |  |
| <b>Галузь знань</b>   |  |       | 12 «Інформаційні технології»  |                        | <b>Рівень вищої освіти</b>   |                        | бакалавр  |  |
| <b>Спеціальність</b>  |  |       | 122 Комп'ютерні науки   |                        | <b>Семестр</b>   |                        | 1,2   |  |
| <b>Освітня програма</b>   |  |       | Комп'ютерні науки   |                        | <b>Тип дисципліни</b>  |                        | Обов'язкова   |  |
| <b>3. Обсяг:</b>  | Кредитів ECTS  | Годин | За видами занять:   |                        |  |                        |   |  |
|   |  |       | Лекцій  | Семінарських<br>занять | Практичних занять  | Лабораторних<br>занять | Самостійна<br>підготовка  |  |
|   | 11   | 330   | 54  |                        | 108  |                        | 168   |  |
| <b>АНОТАЦІЯ КУРСУ</b>   |  |       |   |                        |  |                        |   |  |
| <b>Взаємозв'язок у структурно-логічній схемі</b>  |  |       |   |                        |  |                        |   |  |
| Освітні компоненти, які передують вивченню  |  |       | Дискретні структури<br>Групова динаміка і комунікації<br>Іноземна мова<br>Соціально-екологічна безпека життєдіяльності  |                        |  |                        |   |  |
| Освітні компоненти для яких є базовою   |  |       | Прикладні алгоритми та структури даних<br>Системне програмування та архітектура комп'ютерів<br>Проектування інтерфейсу користувача<br>Програмування мобільних пристроїв<br>Організація баз даних та знань |                        |  |                        |   |  |
| <b>Мета курсу:</b>  | Отримання теоретичних знань і практичних навичок розв'язання задач із різних галузей знань та прикладного застосування із застосуванням програмного підходу (програмування) на мові Java |       |   |                        |  |                        |   |  |
| <b>Компетенції відповідно до освітньої програми</b>   |  |       |   |                        |  |                        |   |  |
| <b>Soft- skills / Загальні компетентності (ЗК)</b>  |  |       |   |                        | <b>Hard-skills / Спеціальні компетентності (ПП)</b>  |                        |   |  |
| ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.<br>ЗК2. Здатність застосовувати знання у практичних ситуаціях.<br>ЗК4. Здатність спілкуватися державною мовою як усно, так і письмово.<br>ЗК5. Здатність спілкуватися іноземною мовою.<br>ЗК6. Здатність вчитися й оволодівати сучасними знаннями.<br>ЗК7. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.<br>ЗК8. Здатність генерувати нові ідеї (креативність). |  |       |   |                        | ПП8. 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.<br>ПП14. Здатність застосовувати методи та засоби забезпечення інформаційної безпеки, розробляти й експлуатувати спеціальне програмне забезпечення |                        |   |  |

|   |  |
|---|--|
| <p>ЗК9. Здатність працювати в команді.<br/> ЗК11. Здатність приймати обґрунтовані рішення.<br/> ЗК12. Здатність оцінювати та забезпечувати якість виконуваних робіт.<br/> ЗК13. Здатність діяти на основі етичних міркувань.<br/> ЗК14. Здатність реалізувати свої права і обов'язки як члена суспільства, усвідомлювати цінності громадянського (вільного демократичного) суспільства та необхідність його сталого розвитку, верховенства права, прав і свобод людини і громадянина в Україні.<br/> ЗК15. Здатність зберігати та примножувати моральні, культурні, наукові цінності і досягнення суспільства на основі розуміння історії та закономірностей розвитку предметної області, її місця у загальній системі знань про природу і суспільство та у розвитку суспільства, техніки і технологій, використовувати різні види та форми рухової активності для активного відпочинку та ведення здорового способу життя.</p> | <p>захисту інформаційних ресурсів об'єктів критичної інформаційної інфраструктури.<br/> ПП16. Здатність реалізовувати високопродуктивні обчислення на основі хмарних сервісів і технологій, паралельних і розподілених обчислень при розробці й експлуатації розподілених систем паралельної обробки інформації.</p> |
|---|--|

### Програмні результати навчання (ПРН)

|  |
|--|
| <p>ПРН1. Застосовувати знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу, обробки та синтезу інформації в предметній області комп'ютерних наук.<br/> ПРН9. 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.<br/> ПРН16. Виконувати паралельні та розподілені обчислення, застосовувати чисельні методи та алгоритми для паралельних структур, мови паралельного програмування при розробці та експлуатації паралельного та розподіленого програмного забезпечення.</p> |
|--|

### ОРГАНІЗАЦІЯ НАВЧАННЯ

| Тема, опис теми | Вид заняття | Оцінювання за тему | Форми і методи навчання/питання до самостійної роботи |
|-----------------|-------------|--------------------|---|
|-----------------|-------------|--------------------|---|

### Розділ 1 (Семестр 1). Основи структурного програмування на мові Java

|   |
|---|
| <p><b>Тема 1. Загальні принципи розробки програм, в тому числі і на Java</b><br/> Знати:<br/> 1. Призначення мов програмування<br/> 2. Особливості мови Java<br/> 3. Узагальнений процес розробки програм<br/> 4. Структура програми на Java. Перша програма "Hello, World!"<br/> 5. Код програми та коментарі. Правила оформлення коду.<br/> 6. Різновиди помилок та базові правила їх виправлення</p> |
|---|

|   |                                 |  |  |
|---|---------------------------------|--|--|
| <p>7. Порядок запуску програми з командного рядку та через IDE Eclipse</p> <p>Вміти:</p> <ol style="list-style-type: none"> <li>1. Встановлення JDK, IDE Eclipse</li> <li>2. Порядок запуску програми з командного рядку та через IDE Eclipse</li> <li>3. Способи детектування, виявлення та виправлення похибок</li> <li>4. Загальну структуру програми на Java. Код першої програми "Hello, World!"</li> <li>5. Правила оформлення коду та коментарів згідно code convention.</li> </ol> <p>Формування компетенцій: ЗК1,ЗК2,ЗК4,ЗК6,ЗК7,ЗК15,ЗК16,ЗК17,ПП8</p> <p>Результати навчання: ПРН1, ПРН9</p> <p><b>Рекомендовані джерела:</b> 1-6</p>  |                                 |  |  |
| Заняття 1.1 Ведення до мови програмування Java  | Лекція 1<br>2 год               |  | Лекція-візуалізація  |
| Заняття 1.2 Створення та запуск програм на Java   | Практичне<br>заняття 1<br>4 год |  | Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу, тренінгові технології                           |
| Тема 1. Загальні принципи розробки програм, в тому числі і на Java  | Самостійна<br>робота            |  | Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань |
| <p><b>Тема 2. Поняття типу даних. Вбудовані типи даних String, int.</b></p> <p>Знати:</p> <ol style="list-style-type: none"> <li>1. Поняття типу даних та змінної у Java. Вбудовані та примітивні типи даних.</li> <li>2. Цілочисельний тип даних int. Основні характеристики, операції та сфери застосування.</li> <li>3. Передачі даних у програму через командний рядок</li> <li>4. Рядковий тип даних String. Основні характеристики, операції та сфери застосування.</li> <li>5. Приклади типового застосування цілочисельних змінних та роботи із текстовими даними.</li> </ol> <p>Вміти:</p> <ol style="list-style-type: none"> <li>1. Декларувати та ініціалізувати змінні</li> <li>2. Здійснювати базові операції із цілочисельним типом int</li> <li>3. Здійснювати базові операції із рядковим типом даних String</li> <li>4. Передавати данні у програму через командний рядок</li> <li>5. Конвертувати данні із рядкового представлення у чисельний</li> </ol> <p>Формування компетенцій: ЗК1,ЗК2,ЗК4,ЗК6,ЗК7,ЗК15,ЗК16,ЗК17,ПП8</p> <p>Результати навчання: ПРН1, ПРН9</p> <p><b>Рекомендовані джерела:</b> 1-6</p> |                                 |  |  |
| Заняття 2.1 Поняття типу даних. Вбудовані типи даних String, int.   | Лекція 2<br>2 год               |  | Лекція-візуалізація  |

|   |                              |  |  |
|---|------------------------------|--|--|
| Заняття 2.2 Застосування змінних типу int, String у програмах   | Практичне заняття 2<br>4 год |  | Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу, тренінгові технології                           |
| Тема 2. Поняття типу даних. Вбудовані типи даних String, int.   | Самостійна робота            |  | Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань |
| <b>Тема 3. Вбудовані типи даних double, boolean</b><br>Знати:<br>1. Тип даних для роботи з дійсними числами double. Основні характеристики, операції та сфери застосування.<br>2. Принцип застосування математичних функцій з бібліотеки Math<br>3. Тип даних boolean для роботи із логічними значеннями. Основні характеристики, операції та сфери застосування.<br>Вміти:<br>1. Здійснювати базові операції із типом дійсних чисел double.<br>2. Застосовувати математичні функції із бібліотеки Math<br>3. Декларувати та ініціалізувати змінні логічного типу boolean<br>Формування компетенцій: ЗК1,ЗК2,ЗК4,ЗК6,ЗК7,ЗК15,ЗК16,ЗК17,ПП8<br>Результати навчання: ПРН1, ПРН9<br><i>Рекомендовані джерела:</i> 1-6 |                              |  |  |
| Заняття 3.1 Вбудовані типи даних double, boolean. Бібліотека Math   | Лекція 3<br>2 год            |  | Лекція-візуалізація  |
| Заняття 3.2 Застосування змінних типу double, boolean у програмах   | Практичне заняття 3<br>4 год |  | Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу  |
| Тема 3. Вбудовані типи даних double, boolean  | Самостійна робота            |  | Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань |
| <b>Тема 4. Логічні операції та операції порівняння. Перетворення типів.</b><br>Знати:<br>1. Логічні операції. Приклади застосування<br>2. Операції порівняння. Приклади застосування<br>3. Призначення перевірки типів у Java.<br>4. Правила перетворень типів у Java. Приклади правильного та неправильного перетворення типів. Наслідки невірної перетворення типів.<br>Вміти:<br>1. Здійснювати логічні операції у відповідних виразах<br>2. Здійснювати операції порівняння   |                              |  |  |

|  |                              |  |  |
|--|------------------------------|--|--|
| <p>3. Здійснювати перетворень типів змінних у Java різними способами: явне та неявне перетворення типів</p> <p>4. Вміти обирати потрібний тип перетворення в залежності від задачі</p> <p>Формування компетенцій: ЗК1,ЗК2,ЗК4,ЗК6,ЗК7,ЗК15,ЗК16,ЗК17,ПП8</p> <p>Результати навчання: ПРН1, ПРН9</p> <p><b>Рекомендовані джерела:</b> 1-6</p>   |                              |  |  |
| Заняття 4.1 Логічні операції та операції порівняння. Перетворення типів  | Лекція 4<br>2 год            |  | Лекція-візуалізація  |
| Заняття 4.2 Здійснення операцій порівняння, логічних операцій, перетворення типів  | Практичне заняття 4<br>4 год |  | Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу  |
| Тема 4. Логічні операції та операції порівняння. Перетворення типів.   | Самостійна робота            |  | Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань |
| <p><b>Тема 5. Базові конструкції розгалуження потоку виконання програми. Принципи застосування в програмах імітаційного моделювання.</b></p> <p>Знати:</p> <ol style="list-style-type: none"> <li>1. Поняття потоку виконання програми. Блок-схема прямолінійного потоку та потік із розгалуженнями.</li> <li>2. Типи розгалужень. Програмні конструкції для розгалужень у Java</li> <li>3. Конструкція для умовного переходу if, if-else. Синтаксис та принцип застосування. Приклади застосувань: проста імітація випадкового процесу, сортування двох та трьох чисел</li> <li>4. Конструкція для циклічного виконання коду while. Синтаксис та принцип застосування. Приклад застосувань: генерація ступенів двійки.</li> <li>5. Особливості використання фігурних дужек в коді Java для створення багаторядкових блоків коду</li> </ol> <p>Вміти:</p> <ol style="list-style-type: none"> <li>1. Використовувати конструкцію розгалуження потоку виконання if, if-else.</li> <li>2. Використовувати фігурні дужки для формування багаторядкових блоків коду</li> <li>3. Використовувати генератор випадкових чисел</li> <li>4. Будувати програми, що імітують прості випадкові процеси (підкидування монетки, грального кубіку тощо)</li> <li>5. Використовувати конструкцію для циклічного виконання коду while</li> <li>6. Будувати програми для обчислення значень функцій для аргументів на певному інтервалі та виведення результату у табличному вигляді у консоль *(на прикладі розрахунку ступенів двійки)</li> </ol> <p>Формування компетенцій: ЗК1,ЗК2,ЗК4,ЗК6,ЗК7,ЗК15,ЗК16,ЗК17,ПП8</p> <p>Результати навчання: ПРН1, ПРН9</p> <p><b>Рекомендовані джерела:</b> 1-6</p> |                              |  |  |
| Заняття 5.1 Розгалуження потоку виконання програми.  | Лекція 5<br>2 год            |  | Лекція-візуалізація, експрес-опитування студентів  |
| Заняття 5.2 Застосування конструкцій розгалуження потоку виконання у програмах на Java   | Практичне заняття 5          |  | Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу  |

|  |                              |  |  |
|--|------------------------------|--|--|
|  | 4 год                        |  |  |
| Тема 5. Базові конструкції розгалуження потоку виконання програми. Принципи застосування в програмах імітаційного моделювання.   | Самостійна робота            |  | Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань |
| <b>Тема 6. Специфічні конструкції розгалуження потоку виконання програм у Java. Область видимості змінних</b><br>Знати:<br>1. Конструкція формування розгалужень switch-case. Синтаксис та принцип застосування. Приклади застосувань. Порівняння із конструкцією if-else<br>2. Конструкція для циклічного виконання коду for. Синтаксис та принцип застосування. Порівняння із конструкцією while.<br>3. Типові приклади застосувань: обчислення сум, добутків (факторіалу), розрахунок та вивід табличного представлення аргументів та значень функцій<br>4. Поняття області видимості змінної.<br>Вміти:<br>1. Застосовувати конструкцію формування розгалужень switch-case.<br>2. Обирати між конструкціями if-else та switch-case в залежності від задачі та вимог до оптимальності<br>3. Застосовувати конструкцію для циклічного виконання коду for.<br>4. Обирати між конструкціями while та for в залежності від задачі та вимог до оптимальності або зручності<br>5. Здійснювати обчислення сум, добутків (факторіалу), здійснювати вивід табличного представлення аргументів та значень функцій<br>6. Звужувати та розширювати області видимості змінних.<br>Формування компетенцій: ЗК1,ЗК2,ЗК4,ЗК6,ЗК7,ЗК15,ЗК16,ЗК17,ПП8<br>Результати навчання: ПРН1, ПРН9<br><i>Рекомендовані джерела:</i> 1-6 |                              |  |  |
| Заняття 6.1 Специфічні конструкції розгалуження потоку виконання програм у Java. Область видимості змінних   | Лекція 6<br>2 год            |  | Лекція-візуалізація  |
| Заняття 6.2 Застосування специфічних конструкцій розгалуження потоку виконання програм у Java та управління областю видимості змінних  | Практичне заняття 6<br>4 год |  | Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу  |
| Тема 6. Специфічні конструкції розгалуження потоку виконання програм у Java. Область видимості змінних   | Самостійна робота            |  | Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань |
| <b>Тема 7. Вкладені блоки коду. Відлагодження коду.</b><br>Знати: 1. Поняття вкладених блоків коду. Задачі, для яких вони застосовуються<br>2. Приклади вкладених умовних конструкцій та циклів.<br>3. Поняття методу Монте-Карло та принцип реалізації в імітаційних програмах  |                              |  |  |

4. Приклад розрахунку податку на суму
5. Задача про розорення гравця. Код імітаційного моделювання розорення/виграшу. Збір статистичної інформації. Візуалізація процесу гри у консолі
6. Поняття та основні способи відлагодження програм. Тимчасові вставки в код та Debugger в IDE Eclipse

Вміти:

1. Утворювати вкладені блоки коду під час написання коду, бачити необхідність їх застосування для відповідних задач
2. Приклади вкладених умовних конструкцій та циклів.
3. Створювати прості імітаційні програми із застосуванням методу Монте-Карло
4. Візуалізувати процес роботи імітаційної моделі, виводити допустиму статистичну інформацію
5. Застосовувати різні способи відлагодження програм (тимчасові вставки в код та Debugger в IDE Eclipse)

Формування компетенцій: ЗК1,ЗК2,ЗК4,ЗК6,ЗК7,ЗК15,ЗК16,ЗК17,ПП8

Результати навчання: ПРН1, ПРН9

**Рекомендовані джерела:** 1-6

|   |                              |  |  |
|---|------------------------------|--|--|
| Заняття 7.1 Вкладені блоки коду. Відлагодження коду.  | Лекція 7<br>2 год            |  | Лекція-візуалізація  |
| Заняття 7.2 Застосування вкладених блоків у кодї програм, застосування механізмів від лагодження коду | Практичне заняття 7<br>4 год |  | Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу  |
| Тема 7. Вкладені блоки коду. Відлагодження коду.  | Самостійна робота            |  | Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань |

### **Тема 8. Масиви. Базові операції та шаблони застосування одномірних масивів.**

Знати: 1. Поняття структури даних. Поняття масиву.

2. Математичні абстракції та аналоги в мові програмування Java. Масив як математична множина.
3. Порівняння ефективності різних способів обробки множини значень одного типу в кодї Java.
4. Підтримка масивів в Java. Представлення масиву у пам'яті. Особливості копіювання масивів в Java.
5. Розв'язання задач із масивами. Типові шаблони застосування
6. Типові помилки при роботі з масивами, шляхи виявлення та подолання в Java

Вміти:

1. Створювати масиви різної довжини, ініціалізувати даними під час декларування
2. Застосовувати масив у програмах для збереження множин значень, в т.ч. компонентів векторів
3. Працювати із аргументами командного рядка, які зберігаються у масиві
4. Заповнювати масив випадковими значеннями із певного діапазону
5. Обробляти значення масиву у циклі за різними математичними формулами (в т.ч. векторні операції)
6. Копіювати масив в інший масив, розбивати масив на кілька масивів, копіювати масиви з кількох в один
7. Виводити елементи масиву у консоль в рядок, в стовпчик

|  |                              |  |  |
|--|------------------------------|--|--|
| 8. Знаходити мінімальний/максимальний елемент у масиві<br>9. Бачити похибки у роботі із масивами, виявляти їх у кодї та виправляти<br>Формування компетенцій: ЗК1,ЗК2,ЗК4,ЗК6,ЗК7,ЗК15,ЗК16,ЗК17,ПП8<br>Результати навчання: ПРН1, ПРН9<br><b>Рекомендовані джерела:</b> 1-6   |                              |  |  |
| Заняття 8.1 Введення у масиви. Створення та застосування одновимірних масивів  | Лекція 8<br>2 год            |  | Лекція-візуалізація  |
| Заняття 8.2 Застосування одновимірних масивів  | Практичне заняття 8<br>4 год |  | Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу  |
| Тема 8. Масиви. Базові операції та шаблони застосування одновимірних масивів.  | Самостійна робота            |  | Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань |
| <b>Тема 9. Використання одновимірних масивів у класичних задачах комп'ютерних наук</b><br>Знати:<br>1. Застосування масивів для зберігання різних значень однієї сутності<br>2. Вибір довільних елементів з множини значень одного типу<br>3. Тасування значень у множині та формування виборки<br>4. Використання масивів для більш ефективного написання коду<br>5. Задача про збирача купонів: імітаційне моделювання на Java<br>Вміти:<br>1. Застосовувати масив для зберігання різних значень, формування множин значень сутностей, окремі компоненти яких зберігаються в різних масивах<br>2. Здійснювати вибір випадкових значень із масиву різними способами (з можливими повторами, з уникненням повторів використовуючи тасування або додатковий масив логічних значень)<br>3. Здійснювати вибір значень із масиву за певним принципом (задача про збирача купонів), візуалізувати процес<br>4. Використовувати масив для більш ефективного написання коду<br>Формування компетенцій: ЗК1,ЗК2,ЗК4,ЗК6,ЗК7,ЗК15,ЗК16,ЗК17,ПП8<br>Результати навчання: ПРН1, ПРН9<br><b>Рекомендовані джерела:</b> 1-6 |                              |  |  |
| Заняття 9.1 Особливості застосування масивів у задачах комп'ютерних наук   | Лекція 9<br>2 год            |  | Лекція-візуалізація, експрес-опитування студентів  |
| Заняття 9.2 Застосування масивів у задачах комп'ютерних наук   | Практичне заняття 9<br>4 год |  | Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу  |



|   |                   |  |  |
|---|-------------------|--|--|
| Тема 9. Використання одномірних масивів у класичних задачах комп'ютерних наук | Самостійна робота |  | Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань |
|---|-------------------|--|--|

**Тема 10. Двомірні масиви. Приклади використання в задачах комп'ютерних наук.**

Знати:

1. Двомірні масиви. Аналоги з математики. Синтаксис використання в Java
2. Застосування двомірних масивів для матричних обчислень, візуалізація матриць у консолі
3. Принцип зберігання даних таблиць у масивах та їх візуалізація в текстовому вигляді у консолі
4. Задача про випадкове блукання з уникненням перетинів маршруту: імітаційне моделювання та вирішення за допомогою масивів.

Вміти:

1. Здійснювати декларування та ініціалізацію двомірних масивів різними способами
2. Представляти у двомірних масивах матриці, здійснювати матричні операції, візуалізувати матриці у консолі
3. Використовувати двомірний масив для збереження табличних значень, обробляти їх у циклах за математичними формулами, виводити таблицю у консоль
4. Реалізувати програму, що імітує випадковий рух об'єкту у площині за різними правилами, візуалізувати процес

Формування компетенцій: ЗК1,ЗК2,ЗК4,ЗК6,ЗК7,ЗК15,ЗК16,ЗК17,ПП8

Результати навчання: ПРН1, ПРН9

**Рекомендовані джерела:** 1-6

|  |                               |  |  |
|--|-------------------------------|--|--|
| Заняття 10.1 Створення та застосування двовимірних масивів                   | Лекція 10<br>2 год            |  | Лекція-візуалізація  |
| Заняття 10.2 Створення та застосування двовимірних масивів                   | Практичне заняття 10<br>4 год |  | Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу  |
| Тема 10. Двомірні масиви. Приклади використання в задачах комп'ютерних наук. | Самостійна робота             |  | Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань |

**Тема 11. Поняття вводу-виводу, модель вводу-виводу. Бібліотека StdOut та форматований вивід**

Знати:

1. Поняття операцій вводу та виводу даних. Пристрої для реалізації операцій вводу/виводу
2. Модель вводу/виводу. Абстракції вхідного та вихідного потоків
3. Стандартний ввід у Java, формування стандартного вхідного потоку. Бібліотека StdIn та інші засоби (командний рядок)
4. Стандартний вивід у Java, формування стандартного вихідного потоку. Бібліотека StdOut та штатні засоби.
5. Форматування даних у вихідному потоці згідно шаблонів

Вміти:

1. Здійснювати операцію вводу значень у програму під час її запуску, виявляти задачі, для яких такий спосіб є прийнятним
2. Виводити данні у вихідний потік (консоль) засобами бібліотеки StdOut, використовуючи шаблони форматування
3. Застосовувати шаблони форматування для виводу таблиць у вихідний потік (консоль)

Формування компетенцій: ЗК1,ЗК2,ЗК4,ЗК6,ЗК7,ЗК15,ЗК16,ЗК17,ПП8

Результати навчання: ПРН1, ПРН9

**Рекомендовані джерела:** 1-6

|   |                               |  |  |
|---|-------------------------------|--|--|
| Заняття 11.1 Поняття та модель вводу-виводу. Операції виводу даних з програми.              | Лекція 11<br>2 год            |  | Лекція-візуалізація  |
| Заняття 11.2 Застосування команд вводу-виводу для введення даних у програму та їх вивід     | Практичне заняття 11<br>4 год |  | Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу  |
| Тема 11. Поняття вводу-виводу, модель вводу-виводу. Бібліотека StdOut та форматований вивід | Самостійна робота             |  | Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань |

## **Тема 12. Стандартний ввід за допомогою бібліотеки StdIn. Основи обробки даних з файлів текстового формат.**

Знати:

1. API методів бібліотеки стандартного вводу StdIn. Структура даних у вхідному потоці. Забезпечення вводу даних через консоль за допомогою клавіатури
2. Реалізація інтерактивного вводу/виводу даних в програмах на Java
3. Приклади програм із інтерактивним вводом/виводом: обчислення суми та середнього значення, гра "вгадай число"
4. Утворення файлу з вмісту вихідного потоку та формування вхідний потоку із даних файлу: особливості реалізації програм та їх запуску у Java
5. Типові конструкції для завантаження даних з текстових файлів у програму на Java, вивід даних у файл
6. Утворення каналів між програмами

Вміти:

1. Здійснювати ввід даних у програму через консоль, за допомогою клавіатури
2. Розробляти програми із елементами інтерактивної взаємодії (запитання/відповідь/реакція)
3. Розробляти програми обчислення за введеними вхідними даними, прості ігри типу "вгадай число"
4. Утворювати файл з вмісту вихідного потоку
5. Утворювати вхідний потік із даних файлу
6. Обробляти дані із вхідного потоку, утвореного із файлу (розбиття на токени, аналіз кожного токена і т.п.), зберігати у масиві або виводити у консоль
7. Застосування Scanner

Формування компетенцій: ЗК1,ЗК2,ЗК4,ЗК6,ЗК7,ЗК15,ЗК16,ЗК17,ПП8

Результати навчання: ПРН1, ПРН9

**Рекомендовані джерела:** 1-6

|  |                                  |  |  |
|--|----------------------------------|--|--|
| Заняття 12.1 Операції вводу даних у програму.  | Лекція 12<br>2 год               |  | Лекція-візуалізація  |
| Заняття 12.2 Застосування операцій вводу для вирішення типових задач   | Практичне<br>заняття 12<br>4 год |  | Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу  |
| Тема 12. Стандартний ввід за допомогою бібліотеки StdIn.<br>Основи обробки даних з файлів текстового формат.   | Самостійна<br>робота             |  | Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань |
| <b>Тема 13. Стандартна графіка бібліотеки StdDraw. Основи використання для візуалізації даних.</b><br>Знати:<br>1. Місце стандартної графіки у моделі вводу/виводу Java<br>2. API методів бібліотеки StdDraw<br>3. Структура типової програми для візуалізації даних<br>4. Огляд найуживаніших методів бібліотеки StdDraw<br>5. Вивід графіків функцій: особливості реалізації на Java<br>6. Реалізація візуалізації даних з файлів<br>7. Формування зображень, побудованих за ітераційним принципом<br>Вміти:<br>1. Використовувати необхідні методи бібліотеки для створення холсту певного розміру, встановлення масштабу, налаштувати колір фону та пера<br>2. Структура типової програми для візуалізації даних<br>3. Огляд найуживаніших методів бібліотеки StdDraw<br>4. Зображувати примітивні графічні фігури (точка, відрізок, вектор, трикутник і т.п.)<br>5. Виводити графіки функцій, підписувати вісі і т.п.<br>6. Візуалізувати дані, які зберігаються у файлі (карти місцевості, інші об'єкти)<br>7. Візуалізувати зображення, що мають фрактальну природу та будуються за ітеративним принципом<br>Формування компетенцій: ЗК1,ЗК2,ЗК4,ЗК6,ЗК7,ЗК15,ЗК16,ЗК17,ПП8<br>Результати навчання: ПРН1, ПРН9<br><b>Рекомендовані джерела:</b> 1-6 |                                  |  |  |
| Заняття 13.1 Графічні можливості Java.   | Лекція 13<br>2 год               |  | Лекція-візуалізація, експрес-опитування студентів  |
| Заняття 13.2 Застосування засобів бібліотеки StdDraw у програмах   | Практичне<br>заняття 13<br>4 год |  | Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу  |
| Тема 13. Стандартна графіка бібліотеки StdDraw. Основи використання для візуалізації даних.  | Самостійна<br>робота             |  | Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань |

**Тема 14. Анімація та стандартний звук**

Знати:

1. Застосування подвійної буферизації: призначення та спосіб реалізації на Java
2. Класичний метод створення анімаційних ефектів
3. Приклад анімації на Java: задача пружного відскоку шару від стін
4. Застосування звуку у програмах на Java
5. Реалізація годинника із стрілками на Java

Вміти:

1. Створювати програми для візуалізації анімаційних ефектів (покрокова побудова графіку, рух графічних об'єктів і т.п.)
2. Використовувати растрові зображення для візуалізації окремих об'єктів на сцені
3. Використовувати звукові файли (wav) для додавання звукових повідомлень у програму
4. Створювати програму-годинник із стрілками, що рухаються

Формування компетенцій: ЗК1,ЗК2,ЗК4,ЗК6,ЗК7,ЗК15,ЗК16,ЗК17,ПП8

Результати навчання: ПРН1, ПРН9

**Рекомендовані джерела:** 1-6

|  |                                  |  |  |
|--|----------------------------------|--|--|
| Заняття 14.1 Поняття та алгоритми графічної анімації роботи моделей на мові Java | Лекція 14<br>2 год               |  | Лекція-візуалізація  |
| Заняття 14.2 Створення анімації роботи годинника                                 | Практичне<br>заняття 14<br>4 год |  | Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу  |
| Тема 14. Анімація та стандартний звук  | Самостійна<br>робота             |  | Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань |

**Тема 15. Поняття статичного методу Java та принципи їх розробки**

Знати:

1. Актуальність розробки програм за модульним принципом, із використанням методів та зовнішніх бібліотек
2. Поняття статичного методу в Java, сигнатура, синтаксис методу
3. Типи повернень, реалізація повернень значень із методів
4. Трасування виконання програми із методами
5. Виконання методів із інших бібліотек (класів)
6. Передача параметрів в метод: за значенням (для примітивів), за посиланням (для масивів та об'єктів)
7. Принцип перетворення програм без методів на програму із методами: знаходження простих чисел, знаходження найбільшого спільного дільника

Вміти:

1. Розробляти статичні методи, що здійснюють повернення значень (розрахунок за формулами і т.п.) та не здійснюють (вивід даних у консоль)
2. Розробляти методи із різним переліком вхідних параметрів
3. Здійснювати виклик статичних методів, що зберігаються у даному класі або у іншому

|  |                               |  |  |
|--|-------------------------------|--|--|
| <p>4. Передавати у метод та повертати з методу масиви</p> <p>5. Здійснювати розбиття програми, що написана у межах одного методу, на кілька методів (декомпозиція)</p> <p>Формування компетенцій: ЗК1,ЗК2,ЗК4,ЗК6,ЗК7,ЗК15,ЗК16,ЗК17,ПП8</p> <p>Результати навчання: ПРН1, ПРН9</p> <p><b>Рекомендовані джерела:</b> 1-6</p>   |                               |  |  |
| Заняття 15.1 Застосування модульного підходу до програмування на мові Java. Поняття та призначення методів   | Лекція 15<br>2 год            |  | Лекція-візуалізація  |
| Заняття 15.2 Принцип розробки статичних методів  | Практичне заняття 15<br>4 год |  | Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу  |
| Тема 15. Поняття статичного методу Java та принципи їх розробки  | Самостійна робота             |  | Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань |
| <p><b>Тема 16. Перевантаження статичних методів та особливості проектування програм із методами</b></p> <p>Знати:</p> <ol style="list-style-type: none"> <li>Особливість реалізації повернень із методу за допомогою switch-case та кількох операторів return</li> <li>Поняття перевантаження методів: принцип реалізації на прикладі програм для перетворення чисел з hex у dec системи числення, пошуку максимального значення</li> <li>Ситуація неоднозначного виклику методів: причина та шляхи вирішення</li> <li>Поняття локальної змінної у Java, область її видимості</li> <li>Переваги використання методів</li> <li>Принципи розробки програм із методами: підходи top-down та bottom-up</li> </ol> <p>Вміти:</p> <ol style="list-style-type: none"> <li>Здійснювати повернення значень із методу і складних випадках (за допомогою switch-case та кількох операторів return)</li> <li>Реалізовувати перевантажені методи, виявляти таку необхідність відповідно до поставлених задач</li> <li>Розширювати та звужувати область видимості змінних у методах, не допускати повторного декларування</li> <li>Розробляти програми із методами способами top-down або bottom-up</li> </ol> <p>Формування компетенцій: ЗК1,ЗК2,ЗК4,ЗК6,ЗК7,ЗК15,ЗК16,ЗК17,ПП8</p> <p>Результати навчання: ПРН1, ПРН9</p> <p><b>Рекомендовані джерела:</b> 1-6</p> |                               |  |  |
| Заняття 16.1 Особливості розробки статичних методів, їх перевизначення   | Лекція 16<br>2 год            |  | Лекція-візуалізація  |
| Заняття 16.2 Реалізація перевизначення методів у коді  | Практичне заняття 16<br>4 год |  | Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу  |

|  |                               |  |  |
|--|-------------------------------|--|--|
| Тема 16. Перевантаження статичних методів та особливості проектування програм із методами  | Самостійна робота             |  | Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань |
| <p><b>Тема 17. Поняття рекурсії та класичні приклади застосування.</b></p> <p>Знати:</p> <ol style="list-style-type: none"> <li>1. Поняття рекурсії: огляд математичної задачі обрахунку факторіалу та реалізації на Java</li> <li>2. Основні властивості рекурсії: математичні особливості та особливості реалізації на Java</li> <li>3. Порядок виконання рекурсивних програм на Java: візуалізація процесу</li> <li>4. Приклад рекурсивної графіки: візуалізація Н-дерева, побудова броунівського моста</li> </ol> <p>Вміти:</p> <ol style="list-style-type: none"> <li>1. Створювати програми із рекурсією (на прикладі обрахунку факторіалу)та реалізації на Java</li> <li>2. Візуалізувати процес виконання рекурсивної програми</li> <li>3. Оцінювати ступінь використання ресурсів</li> <li>4. Розробляти програми для візуалізації рекурсивних обчислень (Н-дерево, броунівський міст)</li> </ol> <p>Формування компетенцій: ЗК1,ЗК2,ЗК4,ЗК6,ЗК7,ЗК15,ЗК16,ЗК17,ПП8</p> <p>Результати навчання: ПРН1, ПРН9</p> <p><i>Рекомендовані джерела:</i> 1-6</p> |                               |  |  |
| Заняття 17.1 Рекурсія та її реалізація   | Лекція 17<br>2 год            |  | Лекція-візуалізація  |
| Заняття 17.2 Реалізація рекурсії у програмах   | Практичне заняття 17<br>4 год |  | Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу  |
| Тема 17. Поняття рекурсії та класичні приклади застосування.   | Самостійна робота             |  | Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань |
| <p><b>Тема 18. Особливості реалізації рекурсії в Java</b></p> <p>Знати:</p> <ol style="list-style-type: none"> <li>1. Типи проблем з рекурсією: особливості виконання програм на Java та принцип усунення</li> <li>2. Поняття динамічного програмування, особливості реалізації на Java</li> <li>3. Особливості обчислення чисел Фібоначчі різними методами на Java</li> </ol> <p>Вміти:</p> <ol style="list-style-type: none"> <li>1. Виявляти проблеми у програмах із рекурсією (нестача пам'яті, нескінченість рекурсивних викликів), уникати або виправляти похибки</li> <li>2. Створювати програми, в яких використовується принцип кешування даних (динамічне програмування)</li> </ol> <p>Формування компетенцій: ЗК1,ЗК2,ЗК4,ЗК6,ЗК7,ЗК15,ЗК16,ЗК17,ПП8</p> <p>Результати навчання: ПРН1, ПРН9</p> <p><i>Рекомендовані джерела:</i> 1-6</p>  |                               |  |  |

|  |                                  |  |  |
|--|----------------------------------|--|--|
| Заняття 18.1 Особливості реалізації рекурсії в Java  | Лекція 18<br>2 год               |  | Лекція-візуалізація  |
| Заняття 18.2 Виявлення проблем у програмах із рекурсією  | Практичне<br>заняття 18<br>4 год |  | Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу  |
| Тема 18. Особливості реалізації рекурсії в Java  | Самостійна<br>робота             |  | Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань |
| <b>Розділ 2 (Семестр 2). Основи об'єктно-орієнтованого програмування на мові Java</b>  |                                  |  |  |
| <p><b>Тема 19. Класи та об'єкти</b></p> <p>Знати:</p> <ol style="list-style-type: none"> <li>1. Поняття об'єкту та класу, UML-діаграма класів</li> <li>2. Поняття інкапсуляції, структура класу (поля та методи), область їх видимості</li> <li>3. Змінна об'єктного типу</li> <li>4. Поняття та призначення конструкторів, їх різновиди</li> <li>5. Статичні та нестатичні елементи класу, область їх видимості</li> <li>6. Призначення модифікаторів видимості</li> <li>7. Поняття та призначення методів get, set</li> <li>8. Ключове слово this</li> </ol> <p>Вміти:</p> <ol style="list-style-type: none"> <li>1. Розробляти клас певної сутності на Java, UML</li> <li>2. Створювати об'єкти за допомогою конструкторів, створювати власні конструктори</li> <li>3. Використовувати класи бібліотек Java Date, Random та Point2D</li> <li>4. Обмежувати доступ до полів та методів, застосовувати get/set методи</li> <li>5. Створювати масиви об'єктів</li> </ol> <p>Формування компетенцій: ЗК1,ЗК2,ЗК4,ЗК6,ЗК7,ЗК15,ЗК16,ЗК17,ПП8</p> <p>Результати навчання: ПРН1, ПРН9</p> <p><b>Рекомендовані джерела:</b> 1-6</p> |                                  |  |  |
| Заняття 19.1 Класи та об'єкти  | Лекція 19<br>2 год               |  | Лекція-візуалізація  |
| Заняття 19.2 Розробка класів та створення об'єктів   | Практичне<br>заняття 19<br>2 год |  | Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу  |
| Заняття 19.3 Реалізація принципу інкапсуляції  | Практичне<br>заняття 20<br>2 год |  | Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу  |

|   |                               |  |  |
|---|-------------------------------|--|--|
| Тема 19. Класи та об'єкти   | Самостійна робота             |  | Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань |
| <b>Тема 20. Принципи об'єктного програмування</b><br>Знати:<br>1. Поняття ООП, абстрактності<br>2. Принципи розробки програмного забезпечення у ООП<br>3. Зв'язки та відношення між класами<br>4. Класи-обгортки, автоматичне перетворення типів з примітивів до об'єктних<br>Вміти:<br>1. Застосовувати парадигму ООП для розробки класів програми, використовувати різні зв'язки та відношення<br>2. Розробляти власні конструктори, застосовувати ключове слово this<br>3. Застосовувати класи-обгортки Byte, Short, Integer, Long, Float, Double, Character, and Boolean<br>4. Використовувати класи BigInteger та BigDecimal для обчислень із великими числами<br>5. Використовувати класи для роботи із змінними та незмінними рядками: String, StringBuilder та StringBuffer<br>Формування компетенцій: ЗК1,ЗК2,ЗК4,ЗК6,ЗК7,ЗК15,ЗК16,ЗК17,ПП8<br>Результати навчання: ПРН1, ПРН9<br><i>Рекомендовані джерела:</i> 1-6 |                               |  |  |
| Заняття 20.1 Принципи парадигми ООП   | Лекція 20<br>2 год            |  | Лекція-візуалізація  |
| Заняття 20.2 Реалізація та застосування власних конструкторів   | Практичне заняття 21<br>2 год |  | Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу  |
| Заняття 20.3 Застосування «класів-обгортки», робота із рядками  | Практичне заняття 22<br>2 год |  | Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу  |
| Тема 20. Принципи об'єктного програмування  | Самостійна робота             |  | Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань |
| <b>Тема 21. Наслідування та поліморфізм</b><br>Знати:<br>1. Поняття суперкласу, класу-нащадку, наслідування<br>2. Ключове слово super, створення об'єктів із класів-нащадків<br>3. Поняття перевизначення методів: суть та механізм застосування, відмінність від перевантаження<br>4. Клас Object: призначення, основні методи   |                               |  |  |



5. Обмеження наслідування

6. Сутність поліморфізму, кастинг об'єктів

Вміти:

1. Проектувати ієрархію наслідування, розробляти вміст суперкласу та класів – нащадків, зображати класи та зв'язки у вигляді UML-діаграми

2. Використовувати ключове слово `super`

3. Обмежувати області видимості у класах-нащадках за допомогою модифікаторів видимості `protected`

4. Обмежувати наслідування та перевизначення методів за допомогою ключового слова `final`

5. Перевизначають методи `toString()`, `equals()`

6. Застосовувати поліморфні посилання, використовувати метод `instanceof`

7. Застосовувати `ArrayList` для збереження даних, розробляти структуру даних «стек»

8. Обмежувати можливості наслідування та перевизначення

Формування компетенцій: ЗК1,ЗК2,ЗК4,ЗК6,ЗК7,ЗК15,ЗК16,ЗК17,ПП8

Результати навчання: ПРН1, ПРН9

**Рекомендовані джерела:** 1-6

|  |                               |  |  |
|--|-------------------------------|--|--|
| Заняття 21.1 Основи наслідування та поліморфізму   | Лекція 21<br>2 год            |  | Лекція-візуалізація  |
| Заняття 21.2 Розробка ієрархії класів на основі принципу успадкування та створення відповідних об'єктів із застосуванням принципу поліморфізму | Практичне заняття 23<br>2 год |  | Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу  |
| Заняття 21.3 Застосування структури даних «масив», розробка структури даних «стек»   | Практичне заняття 24<br>2 год |  | Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу  |
| Тема 21. Наслідування та поліморфізм   | Самостійна робота             |  | Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань |

## Тема 22. перехоплення виключень та операції вводу/виводу

Знати:

1. Поняття та призначення виключень, особливості та переваги використання під час обробки помилок

2. Ієрархія класів виключень, виключення що перевіряються обов'язково та не обов'язково

3. Синтаксис конструкції перехоплення `try-catch`

4. Викидання виключень

5. Розробка власних виключень

Вміти:

1. Обробляти виключення там де потрібно за допомогою `try-catch` конструкції

2. Читати та записувати данні у файли за допомогою класів `File`, `PrintWriter`, `Scanner`

Формування компетенцій: ЗК1,ЗК2,ЗК4,ЗК6,ЗК7,ЗК15,ЗК16,ЗК17,ПП8

|   |                               |  |  |
|---|-------------------------------|--|--|
| Результати навчання: ПРН1, ПРН9<br><i>Рекомендовані джерела:</i> 1-6  |                               |  |  |
| Заняття 22.1 Поняття «виключення» та робота із файлами  | Лекція 22<br>2 год            |  | Лекція-візуалізація  |
| Заняття 22.2 Застосування виключень під час роботи із файлами   | Практичне заняття 25<br>2 год |  | Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу  |
| Заняття 22.3 Реалізація власних виключень, I/O операції, читання веб-сторінок з веб   | Практичне заняття 26<br>2 год |  | Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу  |
| Тема 22. Перехоплення виключень та операції вводу/виводу  | Самостійна робота             |  | Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань |
| <b>Тема 23. Абстрактні класи та інтерфейси</b><br>Знати:<br>1. Поняття абстрактного класу, призначення, особливості в коді<br>2. Поняття інтерфейсу, призначення, особливості у коді<br>3. Інтерфейси Comparable, Cloneable: призначення та принцип використання<br>Вміти:<br>1. Створювати абстрактні класи та інтерфейси<br>2. Використовувати класи Calendar, GregorianCalendar<br>3. Порівнювати об'єкти за допомогою інтерфейсу Comparable<br>4. Створювати копії об'єктів за допомогою інтерфейсу Cloneable<br>Формування компетенцій: ЗК1,ЗК2,ЗК4,ЗК6,ЗК7,ЗК15,ЗК16,ЗК17,ПП8<br>Результати навчання: ПРН1, ПРН9<br><i>Рекомендовані джерела:</i> 1-6 |                               |  |  |
| Заняття 23.1 Абстрактні класи та інтерфейси   | Лекція 23<br>2 год            |  | Лекція-візуалізація  |
| Заняття 23.2 Розробка абстрактних класів та інтерфейсів   | Практичне заняття 27<br>2 год |  | Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу  |
| Заняття 23.3 Застосування інтерфейсу Comparable для порівняння об'єктів та інтерфейсу Cloneable для створення копій   | Практичне заняття 28<br>2 год |  | Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу  |

|   |                               |  |  |
|---|-------------------------------|--|--|
| Тема 23. Абстрактні класи та інтерфейси виводу  | Самостійна робота             |  | Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань |
| <b>Тема 24. Основи роботи із графічним фреймворком Java SWING</b><br>Знати:<br>1. Переваги застосування фреймворку для роботи із графікою, історія розвитку (від AWT до JavaFX), причини вибору SWING<br>2. Поняття GUI, ієрархія класів та Java GUI API<br>3. Поняття та класи фрейму (Frame), менеджерів розташування, панелей Panel як контейнеру для елементів управління<br>4. Класи Color,Font<br>5. Загальні властивості GUI-компонентів<br>6. Значки зображень<br>7. Компоненти JButton, JCheckBox, JRadioButton, JLabel, JTextField<br>Вміти:<br>1. Здійснювати основні етапи у проектуванні GUI<br>2. Застосовувати фрейми, панелі, менеджери розташування<br>3. Розміщувати основні компоненти – кнопки, надпис, поле текстового вводу<br>Формування компетенцій: ЗК1,ЗК2,ЗК4,ЗК6,ЗК7,ЗК15,ЗК16,ЗК17,ПП8<br>Результати навчання: ПРН1, ПРН9<br><i>Рекомендовані джерела:</i> 1-6 |                               |  |  |
| Заняття 24.1 Основи роботи із графічним фреймворком Java SWING  | Лекція 24<br>2 год            |  | Лекція-візуалізація  |
| Заняття 24.2 Основи створення однофреймового додатку та розташування панелі, візуальних графічних компонентів   | Практичне заняття 29<br>2 год |  | Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу  |
| Заняття 24.3 Особливості застосування різних менеджерів, панелей, графічних компонентів   | Практичне заняття 30<br>2 год |  | Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу  |
| Тема 24. Основи роботи із графічним фреймворком Java SWING  | Самостійна робота             |  | Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань |
| <b>Тема 25. Засоби SWING для відображення графічних примітивів</b><br>Знати:<br>1. Поняття графічного контексту, клас Graphics<br>2. Методи зображення рядків, ліній, прямокутників, овалів,дуг, багатокутників та поліліній<br>3. Застосування класу FontMetrics<br>4. Відображення растрових зображень  |                               |  |  |

|   |                               |  |  |
|---|-------------------------------|--|--|
| <p>Вміти:</p> <ol style="list-style-type: none"> <li>1. Отримувати доступ до графічного контексту</li> <li>2. Зображати графічні примітиви-крапки, відрізки, замкнені контури, змінювати фон та колір об'єктів</li> <li>3. Виводити текстові надписи у графічному контексті, змінювати шрифт, колір тексту</li> <li>4. Відобразити растрові зображення з графічних файлів, виділяти зони для відображення</li> </ol> <p>Формування компетенцій: ЗК1,ЗК2,ЗК4,ЗК6,ЗК7,ЗК15,ЗК16,ЗК17,ПП8<br/> Результати навчання: ПРН1, ПРН9<br/> <b>Рекомендовані джерела:</b> 1-6</p>  |                               |  |  |
| Заняття 25.1 Засоби SWING для відображення графічних примітивів   | Лекція 25<br>2 год            |  | Лекція-візуалізація  |
| Заняття 25.2 Реалізація доступу до графічного контексту панелі та вивід графічних примітивів  | Практичне заняття 31<br>2 год |  | Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу  |
| Заняття 25.3 Реалізація додатку для перегляду растрових зображень   | Практичне заняття 32<br>2 год |  | Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу  |
| Тема 25. Засоби SWING для відображення графічних примітивів   | Самостійна робота             |  | Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань |
| <p><b>Тема 26. Програмування на основі подій. Внутрішні класи, анонімні класи та їх застосування</b></p> <p>Знати:</p> <ol style="list-style-type: none"> <li>1. Поняття події Event та джерела подій Event Source</li> <li>2. Поняття прослуховувачів Listeners, їх реєстрація у графічних компонентах та обробка подій</li> <li>3. Поняття внутрішніх та анонімних класів, реалізація прослуховувачів на їх основі</li> <li>4. Різні способи визначення прослуховувачів</li> <li>5. Події від миши та події від клавіатури</li> <li>6. Реалізація анімації із використанням таймеру</li> </ol> <p>Вміти:</p> <ol style="list-style-type: none"> <li>1. Створювати та реєструвати прослуховувачі подій для натиснення на кнопки</li> <li>2. Отримувати інформацію про джерело події</li> <li>3. Створювати внутрішні та анонімні класи та застосовувати їх для обробки подій</li> <li>4. Створювати реакції на події від миши – натиснення кнопок, рух</li> <li>5. Створювати реакції на події від натиснення клавіш клавіатури</li> <li>6. Застосовувати Timer для створення анімації</li> </ol> <p>Формування компетенцій: ЗК1,ЗК2,ЗК4,ЗК6,ЗК7,ЗК15,ЗК16,ЗК17,ПП8<br/> Результати навчання: ПРН1, ПРН9</p> |                               |  |  |

|   |                               |  |  |
|---|-------------------------------|--|--|
| <b>Рекомендовані джерела:</b> 1-6   |                               |  |  |
| Заняття 26.1 Програмування на основі подій. Внутрішні класи, анонімні класи та їх застосування  | Лекція 26<br>2 год            |  | Лекція-візуалізація  |
| Заняття 26.2 Розробка та реалізація реакції на подію від натиснення на кнопку   | Практичне заняття 33<br>2 год |  | Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу  |
| Заняття 26.3 Обробка подій від миши та клавіатури, створення анімації на базі Timer   | Практичне заняття 34<br>2 год |  | Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу  |
| Тема 26. Програмування на основі подій. Внутрішні класи, анонімні класи та їх застосування  | Самостійна робота             |  | Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань |
| <p><b>Тема 27. Розподілені обчислення, мережна архітектура «клієнт-сервер»</b></p> <p>Знати: 1. Поняття розподілених та паралельних обчислень<br/> 2. Поняття архітектури типу «клієнт – сервер»<br/> 3. Базові відомості про мережу та технології мережевої взаємодії – ір-адреса, протоколи TCP/IP, DNS, порти<br/> 4. Поняття сокету, клас ServerSocket, передача даних через сокети<br/> 5. Реалізація клієнт-серверного додатку, принцип «серверних» (або «хмарних») обчислень, переваги та недоліки<br/> 6. Клас InetAddress<br/> 7. Принцип передачі та отримання об'єктів через мережу</p> <p>Вміти:<br/> 1. Отримувати інформацію про віддалені хости<br/> 2. Проектувати та реалізовувати логіку додатків із архітектурою «клієнт-сервер», «хмарних» обчислень</p> <p>Формування компетенцій: ЗК1,ЗК2,ЗК4,ЗК6,ЗК7,ЗК15,ЗК16,ЗК17,ПП8</p> <p>Результати навчання: ПРН1, ПРН9</p> |                               |  |  |
| <b>Рекомендовані джерела:</b> 1-6   |                               |  |  |
| Заняття 27.1 Розподілені обчислення, мережна архітектура «клієнт-сервер»  | Лекція 27<br>2 год            |  | Лекція-візуалізація  |
| Заняття 27.2 Створення сокетів для клієнт-серверної взаємодії   | Практичне заняття 35<br>2 год |  | Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу  |
| Заняття 27.3 Розробка клієнт-серверних додатків   | Практичне заняття 36<br>2 год |  | Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу  |

|  |  |                                |  |
|--|--|--------------------------------|--|
| Тема 27. Розподілені обчислення, мережна архітектура «клієнт-сервер»   | Самостійна робота  |                                | Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань |
| <b>МАТЕРІАЛЬНО-ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ ДИСЦИПЛІНИ</b>  |  |                                |  |
| Комп'ютерне обладнання, програмне забезпечення JDK 1.8.xx, IDE Eclipse, мережа Інтернет ауд. 132, 216, 221   |  |                                |  |
| <b>ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ ДИСЦИПЛІНИ</b>  |  |                                |  |
| <p>1. Computer Science: An Interdisciplinary Approach 1st Edition by Robert Sedgewick (Author), Kevin Wayne (Author), Hardcover: 1168 pages, Publisher: Addison-Wesley Professional; 1 edition (June 25, 2016) <a href="http://dl.dut.edu.ua/course/view.php?id=1373">http://dl.dut.edu.ua/course/view.php?id=1373</a></p> <p>2. Intro to Java Programming, Comprehensive Version (10th Edition) 10th Edition, by Daniel Liang, 1344 pages, Publisher: Pearson; 10 edition , 2014 <a href="http://dl.dut.edu.ua/course/view.php?id=1373">http://dl.dut.edu.ua/course/view.php?id=1373</a></p> <p>3. Boyarsky Jeanne, Selikoff Scott. OCA: Oracle Certified Associate Java SE8 Programmer I Study Guide: Exam 1Z0-808. – John Wiley &amp; Sons, Inc., Indianapolis, 2015.- 435 p.</p> <p>4. Java 8 API <a href="https://docs.oracle.com/javase/8/docs/api/">https://docs.oracle.com/javase/8/docs/api/</a></p>  |  |                                |  |
| <b>ПОЛІТИКА КУРСУ («ПРАВИЛА ГРИ»)</b>  |  |                                |  |
| <ul style="list-style-type: none"> <li>• Курс передбачає роботу в колективі, самостійну роботу.</li> <li>• Середовище в аудиторії є дружнім, творчим, відкритим до взаємодопомоги та конструктивної критики.</li> <li>• Освоєння дисципліни передбачає обов'язкове відвідування лекцій і практичних занять, а також обов'язково самостійну роботу.</li> <li>• Самостійна робота включає в себе теоретичне вивчення питань, що стосуються тем лекційних занять, їх поглиблена проробка за рекомендованою літературою. Також самостійна робота передбачає обов'язкове самостійне виконання практичних вправ (написання програм) згідно завдань, заданих на практичних або лекційних заняттях.</li> <li>• Усі завдання, передбачені програмою, мають бути виконані у встановлений термін.</li> <li>• Якщо студент відсутній з поважної причини, він презентує виконання завдання під час самостійної підготовки та консультації викладача.</li> <li>• Під час роботи над завданнями не допустимо порушення академічної доброчесності: при використанні Інтернет ресурсів та інших джерел інформації студент повинен вказати джерело, використане в ході виконання завдання(коментар у заголовку програмного коду). У разі виявлення факту плагіату студент отримує за завдання 0 балів.</li> <li>• Студент, який спізнився, вважається таким, що пропустив заняття з неповажної причини з виставленням 0 балів за заняття, і при цьому має право бути присутнім на занятті.</li> <li>• За використання смартфонів (телефонів, планшетів) і комп'ютерних засобів без дозволу викладача, порушення дисципліни студент видаляється з заняття, за заняття отримує 0 балів.</li> </ul> |  |                                |  |
| <b>КРИТЕРІЇ ТА МЕТОДИ ОЦІНЮВАННЯ</b>   |  |                                |  |
| Умовою допуску до підсумкового контролю є набрання студентом 30 балів у сукупності за всіма темами дисципліни  |  |                                |  |
| Форми контролю   | Види навчальної роботи   | Оцінювання                     |  |
| <b>ПОТОЧНИЙ КОНТРОЛЬ</b>   | <i>Активність на заняттях, у т.ч.:</i>   |                                |  |
|  | <ul style="list-style-type: none"> <li>• присутність на заняттях (при пропусках занять з поважних причин допускається відпрацювання пройденого матеріалу)</li> </ul> | за кожне відвідування 0,5 бала |  |

|  |  |  |   |
|--|--|--|---|
|  | <ul style="list-style-type: none"> <li>• участь у експрес-опитуванні</li> </ul>  | за кожну правильну відповідь 0,25 бала           |   |
|  | <ul style="list-style-type: none"> <li>• доповідь з презентацією за тематикою самостійного вивчення дисципліни (оцінка залежить від повноти розкриття теми, якості інформації, самостійності та креативності матеріалу, якості презентації і доповіді), підготовка реферату</li> </ul>   | за кожну презентацію (реферат) максимум 15 балів |   |
|  | <ul style="list-style-type: none"> <li>• усне опитування, тестування, рішення практичних задач</li> </ul>  | за кожну правильну відповідь 0,5 бала            |   |
|  | <ul style="list-style-type: none"> <li>• участь у навчальній дискусії, обговоренні ситуаційного завдання</li> </ul>  | за кожну правильну відповідь 3 бала              |   |
| <b>РУБІЖНЕ<br/>ОЦІНЮВАННЯ<br/>(МОДУЛЬНИЙ<br/>КОНТРОЛЬ)</b> | Контроль № 1 (тестування, 1 семестр) Теми 1-3  | максимальна оцінка – 15 балів                    |   |
|  | Контроль № 2 (тестування, 1 семестр) Теми 4-7  | максимальна оцінка – 15 балів                    |   |
|  | Контроль № 3 (тестування, 1 семестр) Теми 8-16   | максимальна оцінка – 15 балів                    |   |
|  | Контроль № 4 (тестування, 2 семестр) Тема 19-20  | максимальна оцінка – 28 балів                    |   |
|  | Контроль № 5 (тестування, 2 семестр) Тема 21-23  | максимальна оцінка – 28 балів                    |   |
|  | Контроль № 6 (тестування, 2 семестр) Тема 24-27  | максимальна оцінка – 28 балів                    |   |
| <b>Додаткова<br/>оцінка</b>                                | Участь у наукових конференціях, підготовка наукових публікацій, участь у Всеукраїнських та Міжнародних конкурсах наукових студентських робіт за спеціальністю, створення кейсів тощо.  | максимальна оцінка – 30 балів                    |   |
|  | Отримання сертифікату про проходження online-курсів (або очних) із програмування на мові Java, згідно тематики курсу   | максимальна оцінка – 20 балів                    |   |
|  | Отримання сертифікату Oracle Java SE8 Programmer I (II)  | максимальна оцінка – 50 балів (за кожний)        |   |
| <b>ПІДСУМКОВЕ<br/>ОЦІНЮВАННЯ<br/>Іспит (1 сем.)</b>        | Метою заходу є контроль сформованості практичних навичок та професійних компетентностей, необхідних для виконання професійних обов'язків.<br>Іспит проходить у письмовій формі. Під час виставлення загальної оцінки за дисципліну, враховуються також бали за складені модульні контролю (№1-3), відвідування занять та активність на практичних заняттях і інші види досягнень (див. додаткова оцінка) | максимальна оцінка – 40 балів                    |   |
| <b>Залік (2 сем.)</b>                                      | Залік відбувається шляхом виставлення оцінки згідно суми балів за складені модульні контролю (№4-6), відвідування занять та активність на практичних заняттях і інші види досягнень (див. додаткова оцінка)  | максимальна оцінка – 100 балів                   |   |
| <b>Курсова робота<br/>(2 сем.)</b>                         | Курсова робота оцінюється згідно критеріїв якості укладання та функціонування програмного коду (макс. 40 балів), відповідності вимогам завдання (35 балів) та відповіді на запитання під час усного захисту (25 балів).  | максимальна оцінка – 100 балів                   |   |
| <b>ПІДСУМКОВА ОЦІНКА ЗА ДИСЦИПЛІНУ</b>                     |  |  |   |
| <b>бали</b>  | <b>Критерії оцінювання</b>   | <b>Рівень компетентності</b>                     | <b>Оцінка /запис в<br/>екзаменаційній<br/>відомості</b> |
| 1-34   | Студент повністю не виконав вимог робочої програми навчальної дисципліни.<br>Його знання на підсумкових етапах навчання є фрагментарними.  | Незадовільний<br>Студент не підготовлений до     | Незадовільно 3<br>обов'язковим                          |

|  |   |   |  |
|--|---|---|--|
|  | Студент не допущений до здачі заліку.   | самостійного вирішення задач, які окреслює мета та завдання дисципліни  | повторним вивченням / Не допущений (F) В залікову книжку не пропоставляється |
| <b>ПІДСУМКОВА ОЦІНКА ЗА ДИСЦИПЛІНУ</b> |   |   |  |
| <b>бали</b>                            | <b>Критерії оцінювання</b>  | <b>Рівень компетентності</b>  | <b>Оцінка /запис в<br/>екзаменаційній<br/>відомості</b>                      |
| <b>90-100</b>                          | Студент демонструє повні й міцні знання навчального матеріалу в обсязі, що відповідає робочій програмі дисципліни, правильно й обґрунтовано приймає необхідні рішення в різних нестандартних ситуаціях.<br>Вміє реалізувати теоретичні положення дисципліни в практичних розрахунках, аналізувати та співставляти дані об'єктів діяльності фахівця на основі набутих з даної та суміжних дисциплін знань та умінь.<br>Знає сучасні технології та методи розрахунків з даної дисципліни. За час навчання при проведенні практичних занять, при виконанні індивідуальних / контрольних завдань проявив вміння самостійно вирішувати поставлені завдання, активно включатись в дискусії, може відстоювати власну позицію в питаннях та рішеннях, що розглядаються. Зменшення 100-бальної оцінки може бути пов'язане з недостатнім розкриттям питань, що стосується дисципліни, яка вивчається, але виходить за рамки об'єму матеріалу, передбаченого робочою програмою, або студент проявляє невпевненість в тлумаченні теоретичних положень чи складних практичних завдань. | Високий<br>Повністю забезпечує вимоги до знань, умінь і навичок, що викладені в робочій програмі дисципліни. Власні пропозиції студента в оцінках і вирішенні практичних задач підвищує його вміння використовувати знання, які він отримав при вивченні інших дисциплін, а також знання, набуті при самостійному поглибленому вивченні питань, що відносяться до дисципліни, яка вивчається. | Відмінно /<br>Зараховано (A)   |
| <b>82-89</b>                           | Студент демонструє гарні знання, добре володіє матеріалом, що відповідає робочій програмі дисципліни, робить на їх основі аналіз можливих ситуацій та вміє застосовувати теоретичні положення при вирішенні практичних задач, але допускає окремі неточності.<br>Вміє самостійно виправляти допущені помилки, кількість яких є незначною.<br>Знає сучасні технології та методи розрахунків з даної дисципліни. За час навчання при проведенні практичних занять, при виконанні індивідуальних / контрольних завдань та поясненні прийнятих рішень, дає вичерпні пояснення.  | Достатній<br>Забезпечує студенту самостійне вирішення основних практичних задач в умовах, коли вихідні дані в них змінюються порівняно з прикладами, що розглянуті при вивченні дисципліни  | Добре /<br>Зараховано (B)  |
| <b>75-81</b>                           | Студент в загальному добре володіє матеріалом, знає основні положення матеріалу, що відповідає робочій програмі дисципліни, робить на їх основі аналіз можливих ситуацій та вміє застосовувати при вирішенні типових практичних завдань, але допускає окремі неточності. Вміє пояснити основні положення виконаних завдань та дати правильні відповіді при зміні результату при заданій зміні вихідних параметрів. Помилки у відповідях/ рішеннях/ розрахунках не є системними. Знає характеристики основних положень, що   | Достатній<br>Конкретний рівень, за вивченим матеріалом робочої програми дисципліни.<br>Додаткові питання про можливість використання теоретичних положень   | Добре /<br>Зараховано (C)  |



|       |  |   |   |
|-------|--|---|---|
|       | мають визначальне значення при проведенні практичних занять, при виконанні індивідуальних / контрольних завдань та поясненні прийнятих рішень, в межах дисципліни, що вивчається.  | для практичного використання викликають утруднення.   |   |
| 64-74 | Студент засвоїв основний теоретичний матеріал, передбачений робочою програмою дисципліни, та розуміє постанову стандартних практичних завдань, має пропозиції щодо напрямку їх вирішень. Розуміє основні положення, що є визначальними в курсі, може вирішувати подібні завдання тим, що розглядалися з викладачем, але допускає значну кількість неточностей і грубих помилок, які може усувати за допомогою викладача.   | Середній<br>Забезпечує достатньо надійний рівень відтворення основних положень дисципліни                           | Задовільно /<br>Зараховано (D)  |
| 60-63 | Студент має певні знання, передбачені в робочій програмі дисципліни, володіє основними положеннями, що вивчаються на рівні, який визначається як мінімально допустимий. З використанням основних теоретичних положень, студент з труднощами пояснює правила вирішення практичних/розрахункових завдань дисципліни. Виконання практичних / індивідуальних / контрольних завдань значно формалізовано: є відповідність алгоритму, але відсутнє глибоке розуміння роботи та взаємозв'язків з іншими дисциплінами. | Середній<br>Є мінімально допустимим у всіх складових навчальної програми з дисципліни                               | Задовільно /<br>Зараховано (E)  |
| 35-59 | Студент може відтворити окремі фрагменти з курсу.<br>Незважаючи на те, що програму навчальної дисципліни студент виконав, працював він пасивно, його відповіді під час практичних робіт в більшості є невірними, необґрунтованими.<br>Цілісність розуміння матеріалу з дисципліни у студента відсутні.   | Низький<br>Не забезпечує практичної реалізації задач, що формуються при вивченні дисципліни                         | Незадовільно з<br>можливістю<br>повторного<br>складання) / Не<br>зараховано (FX) В<br>залікову книжку не<br>представляється |
| 1-34  | Студент повністю не виконав вимог робочої програми навчальної дисципліни.<br>Його знання на підсумкових етапах навчання є фрагментарними.<br>Студент не допущений до здачі заліку.   | Незадовільний<br>Студент не підготовлений до самостійного вирішення задач, які окреслює мета та завдання дисципліни | Незадовільно з<br>обов'язковим<br>повторним<br>вивченням / Не<br>допущений (F) В<br>залікову книжку не<br>представляється   |