**ЗАТВЕРДЖЕНО**

Вченою радою факультету

інформаційно-комп’ютерних технологій

\_\_ \_\_\_\_\_\_\_ 202\_\_ р.,

протокол № \_\_

Голова Вченої ради

\_\_\_\_\_\_\_\_ Тетяна НІКІТЧУК

**РОБОЧА ПРОГРАМА**

**вибіркової навчальної дисципліни фахової підготовки**

**«Розробка програмного забезпечення мовою програмування Java»**

Схвалено на засіданні кафедри інженерії програмного забезпечення

\_\_\_ \_\_\_\_\_\_\_\_\_\_ 20\_\_ р.,

протокол № \_\_\_

Завідувач кафедри

\_\_\_\_\_\_\_ Тетяна ВАКАЛЮК

Гарант освітньо-професійної програми

\_\_\_\_\_\_\_ Андрій МОРОЗОВ

Розробник: асистент кафедри інженерії програмного забеспечення Владислав ПІОНТКІВСЬКИЙ

Житомир

2024 – 2025 н.р.

Робоча програма навчальної дисципліни «Розробка програмного забезпечення мовою програмування Java» для здобувачів вищої освіти освітнього ступеня «бакалавр» затверджена Вченою радою факультету інформаційно комп’ютерних технологій від \_\_\_ \_\_\_\_\_\_\_\_\_\_\_ 202\_\_ р., протокол № \_\_\_.

**1. Опис навчальної дисципліни**

|  |  |  |
| --- | --- | --- |
| Найменування показників | Характеристика навчальної дисципліни | |
| денна форма навчання | заочна форма навчання |
| Кількість кредитів 4 | Вибіркова | |
| Модулів – 3 | Лекції | |
| 32 год. | – |
| Змістових модулів – 3 | Практичні | |
| – | – |
| Загальна кількість годин – 120 | Лабораторні | |
| 32 год. | – |
| Тижневих годин для денної форми навчання:  аудиторних – 4  самостійної роботи – 3,5 | Самостійна робота | |
| \_56 год. | – |
| Вид контролю: залік | |

Частка аудиторних занять і частка самостійної та індивідуальної роботи у загальному обсязі годин з навчальної дисципліни становить:

для денної форми навчання – 53 % аудиторних занять, 47 % самостійної та індивідуальної роботи;

**2.** **Мета та завдання навчальної дисципліни**

**Метою** дисципліни є вивчення та застосування сучасних парадигм програмування, зокрема мовою програмування Java, під час програмної реалізації прикладних завдань в області інформаційних систем і технологій та у майбутній професійній діяльності. Дисципліна «Розробка програмного забезпечення мовою програмування Java» є теоретичною та практичною основою сукупності знань та вмінь, що формують профіль спеціаліста в області розробки програмного забезпечення.

**Завдання дисципліни:** вивчення інструментів, основних концепцій, різних стилів програмування на мові Java. Особлива увага приділяється об’єктно-орієнтованому програмуванню, колекціям для виконання різних операцій над об’єктами, багатопотоковому програмуванню. Розглядаються особливості Java 8 та відлагодження програми із використанням спеціалізованих IDE, інструменти збірки та тестування додатків, взаємодія з базою даних, розробка веб-інтерфейсу.

**Предмет дисципліни:** розробка програмного забезпечення мовою програмування Java.

**У результаті** вивчення навчальної дисципліни **студент повинен**

**знати:**

* Синтаксис мови Java
* Основні типи даних та структури управління
* Роботу з винятками та механізми їх обробки, потоки вводу-виводу
* Багатопоточність та принципи синхронізації потоків
* Бібліотеки та фреймворки для роботи з базами даних, створення веб-додатків

**вміти:**

* Писати та компілювати Java код
* Створювати та використовувати класи та об’єкти
* Реалізовувати принципи ООП у програмах
* Обробляти винятки для забезпечення стабільності програм, читати та записувати дані у файл
* Писати багатопоточні програми та керувати потоками
* Використовувати бібліотеки та фреймворки для написання програм мовою Java

**3. Програма навчальної дисципліни**

**Змістовий модуль 1. Основи програмування мовою Java. Об'єктноорієнтоване програмування в контексті Java. Узагальнені типи. Робота з рядками. Регулярні вирази. Date API. Колекції.**

**Тема 1. Вступ до Java SE. Основи мови програмування Java.**

Огляд найпоширеніших IDE для написання програм мовою Java та їх встановлення (Eclipse, IntelliJ IDEA). Структура програми. Поняття Packages. Метод main. Базові типи даних і літерали. Класи-обгортки примітивних типів даних. Змінні та оператори. Вирази. Лінійній програми. Консольне введення та виведення даних. Оператор умовного переходу if-else. Оператор множинного вибору switch. Цикли. Оператор циклу з передумовою (while) та післяумовою (do-while). Оператор циклу з параметром for. Масиви. Оголошення та ініціалізація. Особливості роботи з елементами масиву. Оператор циклу foreach. Клас Arrays. Функції в Java. Параметри функцій. Параметри змінної довжини. Оператор Return. Результат функції. Рекурсивні функції.

**Тема 2. Об'єктно-орієнтоване програмування в контексті Java.**

Ідеологія та основні принципи ООП. Поняття класу і об’єкту. Створення об’єкту. Поля та методи класу. Конструктори. Модифікатори доступу та інкапсуляція. Робота з об’єктами. Методи get та set. Об’єкти, як параметри методів. Метод toString(). Наслідування. Композиція, агрегація. Поняття this і super. Порівняння об'єктів instanceof. Поліморфізм. Перевизначення (override) та перевантаження (overload). Enum тип. Абстрактні класи. Інтерфейси. Дефолтні та статичні методи інтерфейсу. Plain Old Java Object. Внутрішні класи. Зв'язок зовнішнього класу з внутрішнім, і навпаки. Вкладені класи. Локальні та анонімні класи. Приклади використання.

**Тема 3. Узагальнення (Generics). Клас Object. Порівняння об'єктів.**

Клас Object та його методи. Параметризовані класи. Узагальнення (Generics). Наслідування узагальнень. Інтерфейси Comparator, Comparable.

**Тема 4. Робота з рядками. Регулярні вирази. Date API.**

Введення в рядки. Клас String. Методи для маніпуляцій над рядками. Класи StringBuffer и StringBuilder. Регулярні вирази. Класи Pattern і Matcher. Date API.

**Тема 5. Колекції.**

Java Collections Framework. Основні типи колекцій. Основні інтерфейси Collections API. Інтерфейс Collection. Тип колекції «Список». Інтерфейс List та його реалізація: класи ArrayList, LinkedList. Ітерування елементів колекції (інтерфейси Iterator, ListIterator). Методи equals, hashCode. Тип колекції «Множина». Інтерфейс Set та його реалізація класи HashSet, LinkedHashSet Інтерфейси SortedSet, NavigableSet. Клас TreeSet. Тип колекції «Черга». Інтерфейс Queue та його реалізації. Колекції типу «Асоціативний масив (ключ:значення)». Інтерфейс Map і його реалізація: класи TreeMap, LinkedHashMap та HashMap.

**Змістовий модуль 2. Обробка виключних ситуацій. Потоки вводу-виводу. Робота з файлами. Багатопоточне програмування. Java 8. Reflection API. Анотації. Java + XML та JSON**

**Тема 6. Обробка виключних ситуацій. Потоки вводу-виводу. Робота з файлами.**

Ієрархія виключень і помилок. Способи обробки помилок.Блоки trycatch– finally. Обробка виняткових ситуацій. Обробка декількох винятків. Декларація throws. Створення власних винятків. Байтові потоки введення і виведення. FileInputStream і FileOutputStream. Закриття потоків. Класи ByteArrayInputStream і ByteArrayOutputStream. Буферизація потоків байтів: BufferedInputStream і BufferedOutputStream Форматоване виведення: PrintStream і PrintWriter Класи DataOutputStream і DataInputStream. Читання і запис текстових файлів. Буферизація символьних потоків. BufferedReader і BufferedWriter. Серіалізація об'єктів. Інтерфейс Serializable. Клас File. Робота з файлами і каталогами. Архівування. Робота з JAR та ZIP-архівами.

**Тема 7. Багатопоточне програмування.**

Створення потоків із використанням класу Threads. Створення потоків із використанням інтерфейсу Runnable. Синхронізация потоків. Оператор synchronized. Демони потоки. Взаємодія потоків. Методи wait, notify, notifyAll()

**Тема 8. Java 8.**

Лямбда-вирази. Лямбда як параметри і результати методів. Вбудовані функціональні інтерфейси. Stream API. Характеристика Streams. Створення Streams. Анатомія Stream pipeline. Stream API методи.

**Тема 9. Reflection API. Анотації.**

Поняття рефлексії. Reflection API: його можливлсті та використання. Анотації: Їх можливості та використання. Створення власних анотацій.

**Тема 10. Java + XML та JSON.**

XML в Java, маршалинг, демаршалинг. XML парсери. Парсинг JSON, парсер Jackson.

**Змістовий модуль 3. Java і бази даних. Інструменти збірки. Тестування. Вступ до Spring framework. Spring Boot, Spring MVC, Spring Data.**

**Тема 11. Java і бази даних.**

Конфігурація додатку для роботи з базою даних та взаємодія з нею в процесі роботи. JDBC. DriverManager. Statements (Statement / PreparedStatement / CollableStatement). ResultSet. Connection Pool. DAO.

**Тема 12. Інструменти збірки.**

Знайомство та особливості роботи з інструментами збірки: Maven та Gradle.

**Тема 13. Тестування**

Знайомство та особливості роботи з інструментами тестування. Юніт тестування. Framework Junit. Інтеграційне тестування: Mockito framework.

**Тема 14. Вступ до Spring framework**

Знайомство та особливості роботи з фреймворком Spring: Spring IoC container, Spring Bean, життєвий цикл Spring Bean, Spring Context.

**Тема 15. Spring Boot, Spring MVC, Spring Data.**

Spring Boot: автоконфігурація, концепція стартерів. Spring MVC: знайомство з патерном MVC, особливості роботи анотацій @Controller, @RestController, @RequestMapping, @ResponseBody та інші, обробка виключних ситуацій. Spring Data: репозиторії, запити та обробка їх результатів.

**4. Структура (тематичний план) навчальної дисципліни**

| Змістові модулі і теми | Кількість годин | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| денна форма | | | | заочна форма | | | |
| усього | лекції | лабораторні | самостійна робота | усього | лекції | лабораторні | самостійна робота |
| **МОДУЛЬ 1** | | | | | | | | |
| **Змістовий модуль 1. Основи програмування мовою Java. Об'єктно-орієнтоване програмування в контексті Java. Узагальнення (Generics). Робота з рядками. Регулярні вирази. Date API. Колекції.** | | | | | | | | |
| Тема 1. Вступ до Java SE. Особливості синтаксису мови програмування Java. | 8 | 2 | 2 | 4 | – | – | – | – |
| Тема 2. Об'єктно-орієнтоване програмування в контексті Java. | 12 | 4 | 4 | 4 | – | – | – | – |
| Тема 3. Узагальнення (Generics). Порівняння об'єктів. | 8 | 2 | 2 | 4 |  |  |  |  |
| Тема 4. Робота з рядками. Регулярні вирази. Date API. | 8 | 2 | 2 | 4 |  |  |  |  |
| Тема 5. Колекції | 8 | 2 | 2 | 4 |  |  |  |  |
| ***Разом за змістовий модуль 1*** | 44 | 12 | 12 | 20 | – | – | – | – |
| **Змістовий модуль 2. Обробка виключних ситуацій. Потоки вводу-виводу. Робота з файлами. Багатопоточне програмування. Java 8. Reflection API. Анотації. Java + XML та JSON.** | | | | | | | | |
| Тема 6. Обробка виключних ситуацій. Потоки вводу-виводу. Робота з файлами. | 8 | 2 | 2 | 4 | – | – | – | – |
| Тема 7. Багатопоточне програмування | 8 | 2 | 2 | 4 | – | – | – | – |
| Тема 8. Java 8 | 8 | 2 | 2 | 4 |  |  |  |  |
| Тема 9. Reflection API. Анотації | 6 | 2 | 2 | 2 |  |  |  |  |
| Тема 10. Java + XML та JSON | 8 | 2 | 2 | 4 |  |  |  |  |
| ***Разом за змістовий модуль 2*** | 38 | 10 | 10 | 18 | – | – | – | – |
| **Змістовий модуль 3. Java і бази даних. Інструменти збірки. Тестування. Вступ до Spring framework. Spring Boot, Spring MVC, Spring Data.** | | | | | | | | |
| Тема 11. Java і бази даних | 8 | 2 | 2 | 4 |  |  |  |  |
| Тема 12. Інструменти збірки | 6 | 2 | 2 | 2 |  |  |  |  |
| Тема 13. Тестування | 8 | 2 | 2 | 4 |  |  |  |  |
| Тема 14. Вступ до Spring framework | 8 | 2 | 2 | 4 |  |  |  |  |
| Тема 15. Spring Boot, Spring MVC, Spring Data. | 8 | 2 | 2 | 4 |  |  |  |  |
| ***Разом за змістовий модуль 3*** | 38 | 12 | 12 | 18 | – | – | – | – |
| **ВСЬОГО** | 120 | 32 | 32 | 56 | – | – | – | – |

**5. Теми лабораторних занять**

| № з/п | Назва теми | Кількість годин | |
| --- | --- | --- | --- |
| денна форма | заочна форма |
| **МОДУЛЬ 1** | | | |
| **Змістовий модуль 1. Основи програмування мовою Java. Об'єктно-орієнтоване програмування в контексті Java. Узагальнення (Generics). Робота з рядками. Регулярні вирази. Date API. Колекції.** | | | |
| 1 | Тема 1. Встановлення та налаштування JDK 8. Встановлення та налаштування IDE IntelliJ IDEA. Написання простих програм мовою Java (лінійних, з розгалуженням, з циклами, з використанням функцій та масивів). Тип Enum. | 2 | – |
| 2 | Тема 2. Створення структури класу заданої предметної області. Визначення полів об’єкта та відповідних методів для роботи з ним. Визначення конструктора/конструкторів. Створення екземпляру класу. Визначення методів get(), set(), toString().Створення абстрактних класів та їх використання. Створення інтерфейсів та їх використання. Реалізація наслідування у класах. Реалізація поліморфізму у класах. Використання внутрішніх та вкладених класів. | 4 | – |
| 3 | Тема 3. Використання узагальнень. Порівняння об'єктів. | 2 | – |
| 4 | Тема 4. Введення в рядки. Клас String. Методи для маніпуляцій над рядками. Елементи роботи з регулярними виразами та Date API. | 2 | – |
| 5 | Тема 5. Колекції в Java. Створення списку об’єктів. Виконання дій над елементами списку. Ітерування елементів колекції та їх порівняння. Виконання дій над елементами списку. Створення множини об’єктів. Дій над елементами множини. Використання асоціативних масивів. | 2 | – |
| **Змістовий модуль 2. Обробка виключних ситуацій. Потоки вводу-виводу. Робота з файлами. Багатопоточне програмування. Java 8. Reflection API. Анотації. Java + XML та JSON.** | | | |
| 6 | Тема 6. Опрацювання виняткових ситуацій у програмах. Генерування винятків у коді користувача. Робота з байтовими та текстовими потоками введення/виведення. Серіалізація об'єктів. Інтерфейс Serializable. Введення/виведення об'єктів у файл. | 2 | – |
| 7 | Тема 7. Створення та використання потоків в Java. Використання оператора synchronized, методів wait, notify, notifyAll() для реалізації багатопотоковості в програмах. | 2 | – |
| 8 | Тема 8. Робота з лямбда виразами, функціональним інтерфейсом та Stream API. | 2 | – |
| 9 | Тема 9. Робота з Reflection API. Створення власних анотацій. | 2 | – |
| 10 | Тема 10. Парсинг XML та JSON. | 2 | – |
| **Змістовий модуль 3. Java і бази даних. Інструменти збірки. Тестування. Вступ до Spring framework. Spring Boot, Spring MVC, Spring Data.** | | | |
| 11 | Тема 11. Реалізація зв’язку та взаємодії додатку з базою даних. | 2 | – |
| 12 | Тема 12. Конфігурація та робота з інструментом збірки Maven | 2 | – |
| 13 | Тема 13. Написання простих юніт та інтеграційних тестів. | 2 | – |
| 14 | Тема 14. Вступ до Spring framework. | 2 | – |
| 15 | Тема 15. Spring Boot, Spring MVC, Spring Data. | 2 | – |
| **РАЗОМ** | | **32** | – |

**6. Завдання для самостійної роботи**

| № з/п | Назва теми | Кількість годин | |
| --- | --- | --- | --- |
| денна форма | заочна форма |
| **МОДУЛЬ 1** | | | |
| **Змістовий модуль 1. Основи програмування мовою Java. Об'єктно-орієнтоване програмування в контексті Java. Узагальнення (Generics). Робота з рядками. Регулярні вирази. Date API. Колекції.** | | | |
| 1 | Java Collection API | 8 | – |
| **Змістовий модуль 2. Обробка виключних ситуацій. Потоки вводу-виводу. Робота з файлами. Багатопоточне програмування. Java 8. Reflection API. Анотації. Java + XML та JSON.** | | | |
| 2 | Анотації Java | 12 | – |
| 3 | Gson | 12 |  |
| **Змістовий модуль 3. Java і бази даних. Інструменти збірки. Тестування. Вступ до Spring framework. Spring Boot, Spring MVC, Spring Data.** | | | |
| 4 | Gradle | 12 | – |
| 5 | Spring Security, Spring AOP | 12 |  |
| **РАЗОМ** | | **56** | – |

**7. Індивідуальні самостійні завдання**

Індивідуальні самостійні завдання базуються на виконанні та захисті лабораторних робіт.

**8. Методи навчання**

Під час викладання навчальної дисципліни використовуються наступні методи навчання.

* Вербальні методи (лекція, пояснення)
* Практичні методи (проведення дослідів, експериментів, виконання різних видів вправ, практичних завдань, кейсів)
* Дискусійний метод
* Методи самостійної роботи (анотування опрацьованого матеріалу, вирішення задач, проведення розрахунків, написання есе, підготовка доповідей, написання наукових статей)

**9. Методи контролю**

Перевірка досягнення результатів навчання здійснюється з використанням наступних методів.

* Усне опитування, участь у дискусії, відповіді на проблемні запитання
* Перевірка виконання домашніх завдань, практичних завдань, вправ, кейсів
* Перевірка виконання та захист лабораторних робіт
* Експрес-тестування
* Самооцінювання та взаємооцінювання
* Залік

**10. Оцінювання результатів навчання здобувачів вищої освіти**

Оцінювання результатів навчання здобувачів вищої освіти з навчальної дисципліни здійснюється відповідно до Положення про оцінювання результатів навчання здобувачів вищої освіти у Державному університеті «Житомирська політехніка» та розподілу балів, що наведений нижче.

Система оцінювання результатів навчання здобувачів вищої освіти з навчальної дисципліни включає поточний та підсумковий контроль.

Поточний контроль проводиться для оцінювання рівня засвоєння знань, формування умінь і навичок здобувачів вищої освіти впродовж вивчення ними матеріалу модуля (змістових модулів) навчальної дисципліни. Поточний контроль здійснюється під час проведення навчальних занять.

Підсумковий контроль проводиться для підсумкового оцінювання результатів навчання здобувачів вищої освіти з навчальної дисципліни. Підсумковий контроль здійснюється після завершення вивчення навчальної дисципліни. Підсумковий контроль проводиться у формі заліку. Процедура складання заліку визначена у Положенні про організацію освітнього процесу у Державному університеті «Житомирська політехніка».

**Розподіл балів з навчальної дисципліни**

| Види робіт здобувача вищої освіти | Кількість балів за семестр | |
| --- | --- | --- |
| денна форма | заочна форма |
| Виконання завдань поточного контролю | 100 | – |
| **Підсумкова семестрова оцінка** | **100** | – |

**Розподіл балів за виконання завдань поточного контролю**

| Види робіт здобувача вищої освіти | Кількість балів за семестр | |
| --- | --- | --- |
| денна форма | заочна форма |
| Виконання завдань під час навчальних занять | 60 | – |
| Виконання та захист індивідуальних самостійних завдань | 40 | – |
| Виконання науково-дослідної роботи та інших видів робіт (**додаткові – заохочувальні бали**):  1. Участь у студентських предметних олімпіадах, Всеукраїнському конкурсі студентських наукових робіт, грантах, науково-дослідних проектах  2. Підготовка наукових статей, тез доповідей наукових конференцій  3. Інші види робіт (наводиться перелік видів робіт) | – | – |
| **Разом за виконання завдань поточного контролю** | **100** | – |

**Розподіл балів за виконання завдань під час навчальних занять**

| Види робіт здобувача вищої освіти1 | Кількість балів за семестр | |
| --- | --- | --- |
| денна форма | заочна форма |
| Виконання та захист лабораторних робіт | 80 | – |
| Виконання тестових завдань | 20 | – |
| **Разом за виконання завдань під час навчальних занять** | **100** | – |

З метою застосування цілих чисел для оцінювання результатів роботи здобувачів під час навчальних занять може використовуватися 100-бальна шкала оцінювання щодо кожного окремо виду робіт. Розрахунок загальної кількості балів, які здобувач може набрати за результатами роботи під час навчальних занять протягом семестру, проводиться за формулою:

РНЗ = ∑(Рi × ВКi) × КНЗ, (1)

де РНЗ – загальна кількість балів, набраних здобувачем за виконання завдань під час навчальних занять за семестр;

Рi – кількість набраних здобувачем балів за семестр за виконання і-го виду робіт під час навчальних занять (за 100-бальною шкалою);

ВКi – ваговий коефіцієнт за виконання і-го виду робіт під час навчальних занять. Значення вагових коефіцієнтів розраховуються шляхом ділення кількості балів, яка передбачена за виконання окремого виду робіт під час навчальних занять, на сумарну кількість балів за виконання усіх видів робіт під час навчальних занять за семестр;

КНЗ – коригувальний коефіцієнт, який визначається шляхом ділення кількості балів, що передбачена за виконання завдань під час навчальних занять за семестр, на 100 балів.

Якщо здобувач вищої освіти набрав за поточний контроль 60 балів або більше, він може погодити дану оцінку в електронному кабінеті і вона стане семестровою оцінкою за вивчення навчальної дисципліни.

Якщо здобувач вищої освіти під час вивчення навчальної дисципліни набрав 60 балів або більше і бажає покращити свій результат успішності, він проходить процедуру підсумкового контролю у формі заліку. За складання заліку здобувач вищої освіти може набрати 100 балів. Семестрова оцінка з навчальної дисципліни формується за результатами підсумкового контролю.

Здобувач вищої освіти допускається до процедури підсумкового контролю у формі заліку, якщо за виконання завдань поточного контролю набрав 50 балів або більше.

Якщо здобувач вищої освіти за результатами поточного контролю набрав 35−49 балів, він отримує право за власною заявою опанувати окремі теми (змістові модулі) навчальної дисципліни понад обсяги, встановлені навчальним планом освітньої програми. Вивчення окремих складових навчальної дисципліни понад обсяги, встановлені навчальним планом освітньої програми, здійснюється у вільний від занять здобувача вищої освіти час.

Якщо здобувач вищої освіти за результатами поточного контролю набрав від 0 до 34 балів (включно), він вважається таким, що не виконав вимоги робочої програми навчальної дисципліни та має академічну заборгованість. Здобувач вищої освіти отримує право за власною заявою опанувати навчальну дисципліну у наступному семестрі понад обсяги, встановлені навчальним планом освітньої програми.

Процедура надання додаткових освітніх послуг здобувачу вищої освіти з метою вивчення навчального матеріалу дисципліни понад обсяги, встановлені навчальним планом освітньої програми, визначена у Положенні про надання додаткових освітніх послуг здобувачам вищої освіти в Державному університеті «Житомирська політехніка».

**Визнання результатів навчання, набутих у неформальній та/або інформальній освіті**

Визнання результатів навчання, набутих у неформальній та/або інформальній освіті в рамках окремих тем навчальної дисципліни, здійснюється викладачем за зверненням здобувача вищої освіти та представленням документів, які підтверджують результати навчання (сертифікати, свідоцтва, скріншоти тощо). Рішення про визнання та оцінка за відповідну частину освітнього компонента приймається викладачем за результатами співбесіди зі здобувачем вищої освіти.

Визнання результатів навчання, набутих у неформальній та/або інформальній освіті в рамках цілого освітнього компонента, здійснюється за процедурою, яка визначена у Положенні про організацію освітнього процесу у Державному університеті «Житомирська політехніка».

**Шкала оцінювання**

|  |  |  |
| --- | --- | --- |
| Шкала ЄКТС | Національна шкала | 100-бальна шкала |
| A | Зараховано | 90-100 |
| B | Зараховано | 82-89 |
| C | 74-81 |
| D | Зараховано | 64-73 |
| E | 60-63 |
| FX | Не зараховано | 35-59 |
| F | Не зараховано | 0-34 |

**11. Глосарій**

|  |  |  |
| --- | --- | --- |
| № з/п | Термін державною мовою | Відповідник англійською мовою |
| 1 | Інтегроване середовище розробки | IDE |
| 2 | Колекції (Collections) | Collections |
| 3 | Узагальнення (Generics) | Generics |
| 4 | Розширювана мова розмітки | XML |
| 5 | Запис об’єктів JavaScript | JSON |
| 6 | Прикладний програмний інтерфейс | API |
| 7 | Рефлексія (Reflection) | Reflection |
| 8 | Перерахування | Enum |

**Рекомендована література**

***Основна література***

1. Бейтс Б., Сьєрра К. Книга Head First Java. Легкий для сприйняття довідник. 2022.

2. Kennedy D. S., Putten M. Learn Java with Projects: A concise practical guide to learning everything a Java professional really needs to know. Packt Publishing, 2023. 598 с.

3. Bloch J. Effective Java: Programming Language Guide (Java Series). Addison-Wesley, 2001. 252 с.

***Допоміжна література***

1. Eckel B. Thinking in Java. 4-те вид. Pearson, 2006. 1150 с. ISBN 978‐0131872486.
2. Fowler M., Beck K., Brant J. Refactoring: Improving the Design of Existing Code. Addison-Wesley Professional, 1999. 431 с.
3. Gosling J. The Java Language Specification, 3rd Edition. Addison Wesley, 2005. 684 с.

**13. Інформаційні ресурси в Інтернеті**

1. Java. Офіційна документація. — Режим доступу: https://docs.oracle.com/en/java/. — Дата звернення: 01.03.2024.