Міністерство освіти і науки України

Національний університет водного господарства та природокористування
Навчально-науковий інститут автоматики, кібернетики та обчислювальної техніки
Комп'ютерних технологій та економічної кібернетики

«ЗАТВЕРДЖУЮ»
Проректор з науково-педагогічної, методичної та виховної роботи
________________ О.А. Лагоднюк
«______»_______________20__ р.

04-05-73

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
Program of the Discipline

Крос-платформне програмування
Cross-platform programming

спеціальність	126 «Інформаційні системи та технології»
specialt	126 "Information systems and technologies"

Рівне 2019

Робоча програма навчальної дисципліни «Крос-платформне програмування» для здобувачів вищої освіти першого (бакалаврського) рівня, які навчаються за освітньо-професійною програмою «Інформаційні системи і технології» за спеціальністю 126 «Інформаційні системи та технології». Рівне: НУВГП, 2019 р. 16 с.

Розробник: Т.С. Шроль, канд. пед. наук, доцент кафедри комп’ютерних технологій та економічної кібернетики

Робочу програму схвалену на засіданні кафедри комп’ютерних технологій та економічної кібернетики

Протокол № 13 від 1.04. 2020 р.

Завідувач кафедри комп’ютерних технологій та економічної кібернетики _________________ П.М. Грицюк
(підпис)

«____» ____________ 20__ р.

Керівник групи забезпечення спеціальності 126 «Інформаційні системи та технології» _______________ О.М. Гладка
(підпис)

Схвалено науково-методичною радою з якості ННІ автоматики, кібернетики та обчислювальної техніки

Протокол № 8 від 29.04. 2020 р.

Голова науково-методичної ради з якості ННІАКОТ _________________ П.О. Тадеєв
(підпис)

«___» ____________ 20__ р.

Вступ

Робоча програма з начальної дисципліни «Крос-платформне програмування» складена відповідно до освітньо-професійної програми спеціальності 126 «Інформаційні системи та технології» та стандарту вищої освіти за вказаною спеціальністю для першого (бакалаврського) рівня вищої освіти (затверджений наказом МОН України № 1380 від 12.12.2018).

Предметом вивчення навчальної дисципліни є сучасні методи та технології розробки крос-платформних комп'ютерних застосунків, технології крос-платформного програмування, технології людино-комп'ютерної взаємодії.

Вивченню дисципліни передує отримання компетентностей з таких дисциплін, як «Програмування», «Сучасні та спеціалізовані мови програмування», «Методи обчислень», «Web-технології та Web-дизайн», «Організація баз даних та знань» та інших.

Компетентності, які набувають здобувачі вищої освіти протягом вивчення навчальної дисципліни «Крос-платформне програмування» включають здатності:

- здатність застосовувати знання та розуміння предметної області у практичних ситуаціях, виявляти, ставити та вирішувати проблеми;
- вчитися і оволодівати сучасними знаннями, адаптуватися та діяти в новій ситуації;
- застосовувати базові знання сучасних методів проектування, розроблення та аналізу алгоритмів, уявлення про сучасні принципи людино-машинної взаємодії та засоби розробки людиномашинного інтерфейсу, використання розподілених інформаційних систем, технологій хмарних обчислень та крос-платформених мобільних додатків;
- розуміння всіх аспектів з моделювання інформаційних систем та сучасних інформаційних технологій, методів та комп'ютерних засобів обробки, зберігання і представлення інформації;
- до системного мислення, опанування методами аналізу предметної області в прикладних сферах для виявлення та оцінки наявних потреб інформатизації, оцінювання вихідних даних, моделювання та прогнозування, передбачення ризиків та аспектів безпеки, що можуть виникати стосовно розробленого ПЗ;
- проектувати та оцінювати програмне забезпечення, вміння вибирати оптимальний набір засобів операційної системи та інструментальних програмних засобів для розробки розподілених додатків, стратегії використання хмарних технологій;
- застосовувати методології та технології моделювання у процесі дослідження, проектування та експлуатації інформаційних систем, продуктів, сервісів інформаційних технологій, інших об'єктів професійної діяльності.

Ці компетентності допоможуть успішно виконувати майбутні професійні обов'язки.

Анотація

Дисципліна «Крос-платформне програмування» є теоретичною та практичною основою сукупності знань та вмінь, що формують профіль спеціаліста в області розробки крос-платформного програмного забезпечення. Зокрема, дисципліна покликана сформувати у студентів компетентності, необхідні для ефективного використання мови Java при розробці крос-платформного прикладного програмного забезпечення, веб-застосунків, розв'язування практичних обчислювальних задач на комп'ютері.

Викладання навчальної дисципліни «Крос-платформне програмування» забезпечить такі **результати навчання**:

- застосовувати розуміння принципів структурного програмування, сучасних процедурно-орієнтованих мов, основних структур даних для розробки складних програмних систем;
- володіти сучасними технологіями автоматизації проектування складних об'єктів і систем, продуктів і сервісів інформаційних технологій, сучасними парадигмами та мовами програмування;
- розуміти теорію побудови та володіти сучасними методами проектування, розробки та використання розподілених інформаційних систем (архітектура, структура і основні задачі створення, концепція апаратних і програмних рішень), методологією прийняття оптимальних рішень щодо складу програмного забезпечення, алгоритмів, процедур і операцій;
- застосовувати сучасні технології та інструментальні засоби розробки програмних систем на всіх етапах життєвого циклу;
- пояснити різницю між різними парадигмами програмування, охарактеризувати види програмування, здійснювати класифікацію методів розроблення інформаційних систем;
- демонструвати поєднання різних методів проектування, програмування та створення сучасних систем обробки інформації, обчислювальних систем різного призначення.

Ключові слова: інструментальні середовища програмування, Eclipse, IntelliJ IDEA, Java 8, JDK, Java SDK, JVM, Apache Tomcat, MySQL, JDBC, веб-застосунки, сервлет, Rest API, об'єктно-орієнтоване програмування, колекції.

Abstract

The discipline "Cross-platform programming" is a theoretical and practical basis of a set of knowledge and skills that form the profile of a specialist in the field of cross-platform software development. In particular, the discipline is designed to form in students the competencies necessary for the effective use of Java in the development of cross-platform application software, web applications, solving practical computational problems on a computer.

Teaching the discipline "Cross-platform programming" will provide the following learning outcomes:

- apply an understanding of the principles of structural programming, modern procedural-oriented languages, basic data structures for the development of complex software systems;
- have modern technologies for automating the design of complex objects and systems, products and services of information technology, modern paradigms and programming languages;
- understand the theory of construction and have modern methods of design, development and use of distributed information systems (architecture, structure and main tasks of creation, the concept of hardware and software decisions), methodology for making optimal decisions about software, algorithms, procedures and operations;
- apply modern technologies and tools for software systems development at all stages of the life cycle;
- explain the difference between different paradigms of programming, describe the types of programming, classify methods of developing information systems;
- demonstrate a combination of different methods of design, programming and creation of modern information processing systems, computer systems for different purposes.

Keywords: programming tools, Eclipse, IntelliJ IDEA, Java 8, JDK, Java SDK, JVM, Apache Tomcat, MySQL, JDBC, web applications, servlet, Rest API, object-oriented programming, collections.

1. Опис навчальної дисципліни

<table>
<tr><td rowspan="2">Найменування показників</td><td rowspan="2">Галузь знань,
спеціальність,
рівень вищої освіти</td><td colspan="2">Характеристика навчальної дисципліни</td></tr>
<tr><td>денна форма навчання</td><td>заочна форма навчання</td></tr>
<tr><td>Кількість кредитів – 4,5</td><td>Галузь знань
12 «Інформаційні технології»</td><td colspan="2">Обов’язкова</td></tr>
<tr><td>Модулів – 2</td><td rowspan="5">Спеціальність
126 «Інформаційні системи та технології»</td><td colspan="2">Рік підготовки:</td></tr>
<tr><td rowspan="2">Змістових модулів – 4</td><td>2-й</td><td>-й</td></tr>
<tr><td colspan="2">Семестр</td></tr>
<tr><td rowspan="2">Загальна кількість годин –135</td><td>3-й</td><td>-й</td></tr>
<tr><td colspan="2">Лекції</td></tr>
<tr><td rowspan="11">Тижневих годин
для *денної форми* навчання:
аудиторних – 4 год.
самостійної роботи студента – 6 год.</td><td rowspan="11">Рівень вищої освіти:
перший (бакалаврський)</td><td>20 год.</td><td>- год.</td></tr>
<tr><td colspan="2">Практичні, семінарські</td></tr>
<tr><td>- год.</td><td>- год.</td></tr>
<tr><td colspan="2">Лабораторні</td></tr>
<tr><td>28 год.</td><td>- год.</td></tr>
<tr><td colspan="2">Самостійна робота</td></tr>
<tr><td>87 год.</td><td>- год.</td></tr>
<tr><td colspan="2">Індивідуальні завдання:
- год.</td></tr>
<tr><td colspan="2">Вид контролю:</td></tr>
<tr><td>*екзамен*</td><td></td></tr>
</table>

Примітка: Співвідношення кількості годин аудиторних занять до самостійної роботи становить для денної форми навчання – 35% до 65%.

2. Мета та завдання дисципліни

Мета дисципліни: є формування у студентів теоретичних знань та практичних навичок використання об'єктно-орієнтованої методології програмування для створення сучасних кросплатформенних застосунків; 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.

Завдання дисципліни є:

- набуття теоретичних знань з основ програмування мовою Java, реалізації основних принципів об'єктно-орієнтованого програмування мовою Java;
- формування здатностей застосовувати мову програмування Java та її особливостей для реалізації алгоритмів розв'язування різнотипних завдань;
- засвоєння підходів до створення крос-платформних застосунків;
- формування здатностей проєктувати та розробляти кросп-платформне програмне забезпечення мовою Java із застосуванням структурного, об'єктно-орієнтованого, функціонального з відповідними моделями, методами та алгоритмами обчислень, структурами даних (колекціями) і механізмами управління;
- набуття вмінь використовувати стандартні інтерфейси та класи для створення відповідно користувацьких графічних інтерфейсів та віконних крос-платформних застосунків;
- формування розуміння клієнт-серверної архітектури на Rest API та здатностей щодо розробки веб-застосунків;
- сприяння розвитку усвідомлення студентами цінності передового досвіду й використання його в майбутній професійній діяльності;
- розвиток навиків роботи в команді, здатності до самоосвіти і самовдосконалення щодо розроблення та використання технології Javaв освітньому процесі закладів професійної (професійно-технічної) освіти та бізнес-комунікаціях тощо.

3. Програма навчальної дисципліни

Змістовий модуль 1. Основні відомості про розробку і побудову крос-платформних програм. Програмування на мові Java

Тема 1. Методології і технології розробки крос-платформних застосунків. Місце Java у сучасній IT-індустрії.

Методології і технології розробки крос-платформних програмних продуктів для різних пристроїв на різних платформах. характеристиками нативних та кросплатформних середовищ для розробки програмних додатків для різних платформ. Java Virtual Machine та байт-код. Особливості та різновиди Java. Стандартний інструментарій Java Development Kit. Середовища розробки для Java. Мова програмування Java. Перша програма на Java. Перша програма в IntelliJ IDEA. Перша програма в NetBeans. Перша програма в Eclipse.

Тема 2. Основи програмування мовою Java.

Структура програми. Змінні і константи. типи даних. Консольний введення / виведення в Java. Арифметичні операції. Побітові операції. Умовні вирази. Операції присвоювання і пріоритет операцій. Перетворення базових типів даних. Масиви. Умовні конструкції. Цикли. Функції (методи). Параметри функцій (методів). Оператор return. Результат функції (методу). Перевантаження функцій. Рекурсивні функції. Введення в обробку винятків (try...catch...finally).

Тема 3. Об'єктно-орієнтоване програмування в контексті Java. Класи і об'єкти. Пакети. Наслідування. Абстрактні класи. Інтерфейси. Лямда-вирази.

Класи і об'єкти. Пакети. Модифікатори доступу і інкапсуляція. Статичні члени і модифікатор static. Об'єкти як параметри функцій (методів). Внутрішні і вкладені класи. Наслідування. Абстрактні класи. Ієрархія наслідування і перетворення типів. Інтерфейси. Інтерфейси в механізмі зворотного виклику. Введення в лямбда-вирази. Лямбда як параметри і результати функцій (методів). Вбудовані функціональні інтерфейси. Перерахування enum. Клас Object і його методи. Узагальнення (Generics). Обмеження узагальнень. Наслідування і узагальнення. Довідкові типи і клонування об'єктів. Математичні обчислення і клас Math. Великі числа BigInteger і BigDecimal. Робота з датами. LocalDate.

Тема 4. Обробка винятків. Колекції. Модулі. Створення графічного інтерфейсу засобами Java

Обробка винятків. Оператор throws. Класи винятків. Створення своїх класів винятків.

Колекції. Типи колекцій. Інтерфейс Collection. Клас ArrayList і інтерфейс List. Черги і клас ArrayDeque. Клас LinkedList. Інтерфейс Set і клас HashSet. Інтерфейси SortedSet і NavigableSet та клас TreeSet. Інтерфейси Comparable і Comporator. Сортування. Інтерфейс Map і клас HashMap. Інтерфейси SortedMap і NavigableMap та клас TreeMap. Ітератори.

Модулі. Створення модуля. Залежні модулі. Взаємодія між модулями.

Створення графічного інтерфейсу мовою Java. Бібліотеки SWING, AWT

Тема 5. Потоки введення-виведення. Робота з рядками. Робота з файлами.

Введення в рядки. клас String. Основні операції з рядками. Класи StringBuffer і StringBuilder. Регулярні вирази.

Потоки введення-виведення. Читання і запис файлів. FileInputStream і FileOutputStream. Закриття потоків. Класи ByteArrayInputStream і ByteArrayOutputStream. Буферізовані потоки BufferedInputStream і BufferedOutputStream. Форматоване виведення (PrintStream і PrintWriter).

Класи DataOutputStream і DataInputStream. Читання і запис текстових файлів. Буферизація символьних потоків. BufferedReader і BufferedWriter. Серіалізация об'єктів. Клас File: робота з файлами і каталогами. Робота з ZIP-архівами. Клас Console.

Тема 6. Багатопотокове програмування. Stream API.

Клас Thread. Створення та виконання потоків. Завершення і переривання потоку. Синхронізація потоків. Оператор synchronized. Взаємодія потоків. Методи wait і notify. Семафори. Обмін між потоками. Клас Exchanger. Клас Phaser. Блокування. Клас ReentrantLock. Умови в блокуваннях.

Введення в Stream API. Створення потоку даних. Фільтрація, перебір елементів і відображення. Сортування. Отримання підпотоків і об'єднання потоків. Методи skip і limit. Операції відомості. Метод reduce. Тип Optional. Метод collect. Групування. Паралельні потоки. Паралельні операції над масивами.

Змістовий модуль 2. Сучасні підходи створення крос-платформних застосунків на Java

Тема 7. Базові поняття роботи з JDBC API.

Бібліотека JDBC. Драйвери Connector/J, з'єднання і запити. СУБД MySQL. Просте з'єднання і простий запит. Метадані. Підготовлені запити і процедури для зберігання. Транзакції. Точки відкату. Data Access Object: рівень методу, рівень класу, рівень логіки. Поняття ORM. Специфікація JPA.

Тема 8. Основи архітектури клієнт-сервер. Rest API.

REST-сервіси. Принципи REST. HTTP-методи, HTTP-request, HTTP-response, HTTP-headers. Rest Controller. Анотації. Архітектура застосунків на Rest.

Тема 9. Технології розробки веб-застосунків. Сервлети

Встановлення сервера Apache Tomcat. Структура Apache Tomcat. Інтеграція Eclipse в Tomcat. Сервлет. Життєвий цикл сервлета. Сервлети в Eclipse. Отримання даних в сервлет. Отримання даних з рядка запиту. Передача масивів. Отримання даних з форм.

Тема 10. Технології розробки веб-застосунків. JSP. MVC.

Призначення технології JSP. Обробка JSP-сторінок. Елементи JSP. JavaBeans і JSP. JSP, EL і JSTP. Розробка застосувань. Технологія Model-View Controller.

Тема 11. Робота зі Spring Framework. Spring IoC.

Вступ в Spring Framework. Структура фреймворка. Архітектура програми. Співвідношення об'єктів. Spring IoC. Поняття IoC. Beans. Конфігурація з підтримкою анотацій. Scope бінов. Життєвий цикл біна. Події. Локалізація. Профілі конфігурації.

Тема 12. Робота зі Spring Framework. Spring MVC.

Принципи роботи Spring MVC. Конфігурація Spring Web MVC. Введення в Spring MVC. Патерни MVC. Контролери (controller). Вигляд (view). Анотації. Локалізація. Збирачі даних. Перевірка даних.

4. Структура навчальної дисципліни

Назви змістових модулів і тем	Кількість годин					
	Денна форма навчання					
	усього	у тому числі				
		л	п	лаб	інд	с. р.
1	2	3	4	5	6	7
Модуль 1						
Змістовий модуль 1. Основні відомості про розробку і побудову крос-платформних програм. Програмування на мові Java						
Тема 1. Методології і технології розробки крос-платформних застосунків. Місце Java у сучасній IT-індустрії.	4	1		1		2
Тема 2. Основи програмування мовою Java.	6	1		1		4
Тема 3. Об'єктно-орієнтоване програмування в контексті Java. Класи і об’єкти. Пакети. Наслідування. Абстрактні класи. Інтерфейси. Лямда-вирази	12	2		2		8
Тема 4. Обробка винятків. Колекції. Модулі. Створення графічного інтерфейсу засобами Java.	12	2		2		8
Тема 5. Потоки введення-виведення. Робота з рядками. Робота з файлами	11	1		2		8
Тема 6. Багатопотокове програмування. Stream API.	11	1		2		8
Всього за змістовий модуль 1:	**56**	**8**		**10**		**38**
Змістовий модуль 2. Сучасні підходи створення крос-платформних застосунків на Java						
Тема 7. Базові поняття роботи з JDBC API.	16	2		4		10
Тема 8. Основи архітектури клієнт-сервер. Rest API.	9	2		2		5
Тема 9. Технології розробки веб-застосунків. Сервлети.	16	2		4		10
Тема 10. Технології розробки веб-застосунків. JSP. MVC.	18	2		4		12
Тема 11. Робота зі Spring Framework. Spring IoC.	10	2		2		6
Тема 12. Робота зі Spring Framework. Spring MVC	10	2		2		6
Всього за змістовий модуль 2:	**79**	**12**		**18**		**49**
Усього годин:	**135**	**20**		**28**		**81**

5. Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
1.	Встановлення та налаштування JDK 8. Встановлення та налаштування IDE Eclipse. Написання програм лінійної, розгалуженої та циклічної структури мовою Java. Використання масивів у програмах. Написання функцій.	2
2.	Об'єктно-орієнтоване програмування в контексті Java. Класи і об’єкти. Пакети. Наслідування. Абстрактні класи. Інтерфейси. Лямда-вирази	2
3.	Обробка винятків. Колекції. Модулі. Створення графічного інтерфейсу засобами Java.	2
4.	Потоки введення-виведення. Робота з рядками. Робота з файлами	2
5.	Багатопотокове програмування. Stream API.	2
6.	Основи роботи з JDBC API. MySQL при програмуванні на мові Java.	4
7.	Проектування та особливості розробки Rest-сервісу	2
8.	Розробка веб-застосунків. Сервлети (клас Servlet)	4
9.	Розробка веб-застосунків (JSP, MVC).	4
10.	Розробка веб-застосунку засобами Spring Framework: Spring IoC	2
11.	Розробка веб-застосунку засобами Spring Framework: Spring MVC	2
Всього:		**28**

6. Самостійна робота
Розподіл навчального часу на вивчення дисципліни «Кросп-платформне програмування»

Число кредитів ECTC	Загальний обсяг дисципліни	Розподіл часу		Частка самостійної роботи, в %
		Аудиторні заняття	Самостійна робота	
4,5	135	48	87	65

Розподіл самостійної роботи студента

Опрацювання лекційного матеріалу	0,5*20=	10 годин
Підготовка до лабораторних занять	1*28=	28 годин
Підготовка до контрольних заходів	7*2=	14 годин
Всього		52 годин
Резерв		35 годин

Завдання для самостійної роботи (35 год)

№ з/п	Назва теми	Кількість годин
1.	Клас Arrays. Методи сортування та пошуку елементів в масиві.	2
2.	Регулярні вирази. Класи Pattern і Matcher	3
3.	Створення графічного інтерфейсу засобами Java.	4
4.	Форматоване виведення: PrintStream і PrintWriter Класи DataOutputStream і DataInputStream. Архівування в Java. Робота з JAR та ZIP-архівами	3,5
5.	Паралельне виконання програм, кооперація потоків. Concurrency. Пакет java.util.concurrent. ExecutorService. Заплановане виконання потоків. Stream API методи	3,5
6.	Поняття ORM. Специфікація JPA.	4
7.	Принципи та обмеження RESTful	1
8.	Сервлети. Валідація форм.	4
9.	JavaBeans і JSP. JSP, EL і JSTP	6
10.	Spring IoC. Scope бінов. Життєвий цикл біна. Події. Локалізація. Профілі конфігурації.	2
11.	Spring MVC . Збирачі даних. Перевірка даних.	2
Всього:		**35**

Підсумком самостійної роботи над вивченням дисципліни «Крос-платформне програмування» є складання конспекту за темами, вказаними у завданнях для самостійної роботи. Загальний обсяг конспекту визначається з умови повноти та якості викладеного матеріалу.

Конспект оформлюється на стандартному папері формату А4 або в зошиті. Конспект може бути рукописним або друкованим і виконується українською мовою. В останньому випадку електронний варіант конспекту розміщується на Google Диску студента із відкритим доступом до нього викладачу.

Перевірка конспекту з самостійної роботи відбувається у терміни, спільно обумовлені студентом і викладачем.

7. Методи навчання

Науковими та методичними основами викладання дисципліни «Крос-платформне програмування» є системний підхід. Методологія вивчення дисципліни й оцінювання результатів навчання студентів побудована на підходах індивідуального та диференціального навчання.

Використовуються наступні методи навчання:

1) проблемно-пошукові (евристичні бесіди, навчальні дискусії, лабораторні роботи, «перевернутий клас», «мозковий штурм»);

2) творчо-репродуктивні (ділові ігри, імітація професійної діяльності, аналіз ситуацій, варіативні завдання);

3) дослідницькі (індивідуальні навчально-дослідні завдання).

Зокрема:

1. Лекції проводяться з використанням технічних засобів навчання і супроводжуються демонстрацією за допомогою проектора елементів лекційного матеріалу.

2. Лабораторні роботи проводяться із використанням ПК та відповідного програмного забезпечення: система керування навчання Moodle, Microsoft Office, Java 8, IDE для написання програм на Java (Eclipse, IntelliJ IDEA та ін.), MySQL, Apache Tomcat, Connector/J та ін.

3. Проведення контрольних тестувань.

8. Методи контролю

Оцінювання навчальних досягнень студентів за усіма видами навчальних робіт проводиться за *поточним* та *підсумковим* контролями**.** Поточний контроль знань студентів з навчальної дисципліни проводиться у вигляді модульного комп'ютерного незалежного тестування. Контрольні завдання за змістовим модулем включають тестові питання трьох рівнів складності.

Контроль самостійної роботи проводиться: з лекційного матеріалу – шляхом перевірки конспектів; з лабораторних занять – з допомогою перевірки виконаних завдань, шляхом проведення тестів та захистів лабораторних робіт.

Усі контрольні заходи включено до 100-бальної шкали оцінювання. Підсумковий семестровий контроль знань відбувається на екзамені у вигляді незалежного комп'ютерного тестування.

Основними критеріями, що характеризують рівень компетентності студента при оцінюванні результатів поточного та підсумкового контролів з навчальної дисципліни «Крос-платформне програмування» є:

- виконання всіх видів навчальної роботи, що передбачені робочою програмою навчальної дисципліни;
- глибина і характер знань навчального матеріалу за змістом навчальної дисципліни;
- характер відповідей на поставлені питання (чіткість, лаконічність, логічність, послідовність тощо);
- обґрунтування вибору методу для розв'язання тих чи інших задач;
- рівень вміння аналізувати та захищати одержані результати.

Критерії оцінювання результатів поточної роботи (завдань, що виконуються на лабораторних заняттях, результати самостійної роботи студентів) проводиться у % від кількості балів, виділених на завдання, із заокругленням до цілого числа:

0% – завдання не виконано;

40% – завдання виконано частково та містить суттєві помилки методичного або розрахункового характеру;

60% – завдання виконано повністю, але містить суттєві помилки у розрахунках або в методиці;

80% – завдання виконано повністю і вчасно, проте містить окремі несуттєві недоліки (розмірності, висновки, оформлення тощо);

100% – завдання виконано правильно, вчасно і без зауважень.

9. Критерії та шкала оцінювання програмних результатів навчання

Розподіл балів, які отримують здобувачі

Поточне тестування та самостійна роботи												Підсумковий тест (екзамен)		Сума
Змістовий модуль 1						Змістовий модуль 2						40		
Т1	Т2	Т3	Т4	Т5	Т6	Т7	Т8	Т9	Т10	Т11	Т12	МК1	МК2	100
4	4	4	4	4	4	4	4	8	10	5	5	20	20	

Т1, Т2...Т12 – теми змістових модулів.

Шкала оцінювання

Сума балів за всі види навчальної діяльності	Оцінка за національною шкалою
	для екзамену
90–100	відмінно
82–89	добре
74–81	
64–73	задовільно
60–63	
35–59	незадовільно з можливістю повторного складання
0–34	незадовільно з обов’язковим повторним вивченням дисципліни

10. Методичне забезпечення

1. Опорний конспект лекцій (в електронному вигляді) та лабораторних по всіх темах курсу, у тому числі – для самостійного вивчення.
2. Пакети тестових завдань по кожній темі і в цілому по всьому курсу дисципліни (навчальна платформа Moodle).
3. Програмне забезпечення: система керування навчання Moodle, Microsoft Office, Java 8, IDE для написання програм на Java (Eclipse, IntelliJ IDEA та ін.), MySQL, Apache Tomcat, Connector/J та ін.

11. Рекомендована література

Основна

1. Блинов И.Н., Романчик В. С. Java. Методы программирования : уч.-мет. пособие / И.Н. Блинов, В.С. Романчик. – Минск : издательство «Четыре четверти», 2013. – 896 с.
2. Гаврилов А.В. Программирование на Java. Конспект лекций / А.В. Гаврилов, С.В. Клименков, А.Е. Харитонова, Е.А. Цопа. – СПб: Университет ИТМО, 2015. – 126 с.
3. Динеш Раджпут. Spring. Все паттерны проектирования / Раджпут Динеш. – СПб.: Питер, 2019. – 320 с.
4. Докука О., Лозинский И. Практика реактивного программирования в Spring 5 / О. Докука, И. Лозинский. – М.: ДМК Пресс, 2019. – 508 с.
5. Кейденхед Роджерс. Java за 24 часа, 8-е издание.: Пер. с англ. – СПб.: 000 "Диалектика", 2019. – 480 с.
6. Прохоренок Н. А. Основы Java / Н. А. Прохоренок. – СПб.: БХВ-Петербург, 2017. — 704 с.
7. Седжвик Р., Уэйн К. Computer Science: основы программирования на Java, ООП, алгоритмы и структуры данных / Р. Седжвик, К. Уэйн. – СПб.: Питер, 2018. – 1072 с.
8. Фримен Э. Паттерны проектирование / Эрик Фримен, Элизабет Фримен – СПб: «Питер», 2013. – 656 с.
9. Хорстманн К. С. Java SE8. Базовый курс / Кей С. Хорстманн - М: «Вильямс», 2016. – 464 с.
10. Хорстманн Кей С. Java SE 8. Базовый курс. / Кей С. Хорстманн : Пер. с англ. – М. : ООО "И .Д. Вильямс", 2015. – 464 с.
11. Шилдт Г. Полное руководство 8-е издание. Исчерпывающее описание языка Java. — М.: Издательский дом «Вильямс», 2013. — 1101 с.
12. Шилдт, Герберт. Java 8. Полное руководство; 9-е изд.: Пер. с англ. - М. : ООО "И.Д. Вильяме", 2015. – 1376 с.
13. Эккель Б. Философия Java. 4-е полное изд. – СПб.: Питер,2015. – 1168 с.

Допоміжна

14. Виссер Дж. Разработка обслуживанмых програм на языке Java / Дж. Виссер; пер. с англ. Р.Н. Рагимова. – М.: ДМК Пресс, 2017. – 182 с.
15. Гуськова, Ольга Ивановна. Объектно ориентированное программирование в Java : учебное пособие / О. И. Гуськова. – Москва: МПГУ, 2018. – 240 с.
16. Дашнер С. Изучаем Java EE. Современное программирование для больших предприятий / С. Дашнер. – СПб.: Питер, 2018. – 384 с.
17. Мурат Йенер, Алекс Фидом. Java EE. Паттерны проектирования для профессионалов. - СПб.: Питер, 2016. – 240 с.
18. Эванс Бенджамин Дж., Флэнаган Дэвид. Java. Справочник разработчика / Бенджамин Дж. Эванс, Дэвид Флэнаган; 7-е изд. : Пер. с англ. – СПб. "Диалектика", 2019. - 592 с.

19. Java Concurrency на практике / Гетц Брайан, Пайерлс Тим, Блох Джошуа, Боубер Джозеф, Холмс Дэвид, Ли Даг. – СПб.: Питер, 2020. – 464 с.
20. Ganesh S. G. Oracle Certified Professional Java SE 7 Programmer Exams 1Z0-804 and 1Z0-805: A Comprehensive OCPJP 7 Certification Guide (Expert's Voice in Java) 1st ed. Edition / S G Ganesh, Tushar Sharma - Apress, 2013. – 656 pages
21. Sierra K. OCA/OCP Java SE 7 Programmer I & II Study Guide (Exams 1Z0-803 & 1Z0-804) (Certification Press) 1st Edition / Kathy Sierra, Bert Bates - McGraw-Hill Education; 1 edition, 2014. – 888 pages

12. Інформаційні ресурси

1. Наукова бібліотека НУВГП (м. Рівне, вул. Олекси Новака, 75) / [Електронний ресурс]. – Режим доступу: http://nuwm.edu.ua/naukova-biblioteka
2. Цифровий репозиторій НУВГП. – Режим доступу: http://ep3.nuwm.edu.ua/
3. Java Object Oriented (https://www.tutorialspoint.com)
4. Електронний навчальний курс «Програмування мовою Java». – Режим доступу: https://programing.moodlecloud.com/
5. Java SE 8 Documentation:
https://docs.oracle.com/javase/tutorial/tutorialLearningPaths.html
– https://docs.oracle.com/javase/8/docs/api/index.html
– https://docs.oracle.com/javase/8/docs/index.html
6. Java Tutorials:
– http://download.oracle.com/javase/tutorial/index.html
– https://www.tutorialspoint.com/java/index.htm
– http://proglang.su/java
– https://metanit.com/java/tutorial/
7. The Java Tutorial – URL: https://docs.oracle.com/javase/tutorial/
8. JDBC Tutorial – URL: https://www.tutorialspoint.com/jdbc/index.htm
9. Core Java Tutorial – URL: http://www.javatpoint.com/java-tutorial
10. Sun Microsystems, Inc. JDK 8 Documentation – URL: https://docs.oracle.com/javase/8/docs/
11. Форум Java програмістів – URL: https://javatalks.ru/
12. Global IT community – URL: https://dzone.com/
13. Колективний блог у сфері IT – URL: https://habrahabr.ru
14. Solutions for Java Developers – URL: http://www.javaworld.com/
15. Largest java community for programmers – URL: http://stackoverflow.com/