

Чернівецький національний університет імені Юрія Федьковича

Інститут фізико-технічних та комп'ютерних наук

Кафедра програмного забезпечення комп'ютерних систем



**“ЗАТВЕРДЖУЮ”**

Директор Інституту фізико-технічних та комп'ютерних наук

/Ангельський О.В./

“31” серпня 2021 року

**РОБОЧА ПРОГРАМА**  
навчальної дисципліни

**JAVA-ТЕХНОЛОГІЇ**

**вибіркова**

Освітня програма Інформаційні системи та технології

Спеціальність 126 Інформаційні системи та технології

Галузь знань 12 Інформаційні технології

Рівень вищої освіти перший (бакалаврський)

Інститут фізико-технічних та комп'ютерних наук

Мова навчання українська

Чернівці 2021 рік

Робоча програма навчальної дисципліни «Java-технології» складена відповідно до освітньо-професійної програми першого (бакалаврського) рівня вищої освіти: «Інформаційні системи та технології», спеціальність 126 Інформаційні системи та технології, галузь знань 12 Інформаційні технології, затвердженої Вченою радою Чернівецького національного університету імені Юрія Федьковича (Протокол № 7 від « 31 » серпня 2020 року).

Розробник: Прохоров Георгій Валерійович, доцент кафедри програмного забезпечення комп'ютерних систем, кандидат фізико-математичних наук


Робоча програма затверджена на засіданні кафедри *програмного забезпечення комп'ютерних систем*

Протокол № 1 від « 26 » серпня 2021 року

Завідувач кафедри  Остапов С.Е.  
(підпис)

Схвалено методичною радою Інституту фізико-технічних та комп'ютерних наук

Протокол № 1 від « 31 » серпня 2021 року

Голова методичної ради ІФТКН  Струк Я.М.  
(підпис)

© Прохоров Г.В., 2021 рік  
© ЧНУ, 2021 рік

## 1. Мета навчальної дисципліни

Дисципліна «Java-технології» є частиною освітньо-професійної програми підготовки здобувачів в галузі знань 12 «Інформаційні технології» спеціальності 126 Інформаційні системи та технології.

Навчальна дисципліна призначена для засвоєння необхідних знань із основ об'єктно-орієнтованого програмування, отримання навичок використання класів, механізмів наслідування, інкапсуляції та поліморфізму; отримання практичних навичок самостійно будувати програми середнього рівня складності з використанням структурно-модульного та об'єктно-орієнтованого методів програмування; застосування здобутих знань та підходів для розв'язання практичних задач різного рівня складності.

Основними завданнями курсу «Java-технології» є оволодіння кожним студентом процесом проектування задачі і її реалізації як системи взаємодіючих класів, що діють завдяки інкапсуляції, успадкуванню та поліморфізму, а також іншим засобам мови програмування Java.

## 2. Результати навчання.

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

- концепції та принципи організації програмного коду;
- рівні абстракції даних при проектуванні програмного забезпечення (ПЗ);
- основи роботи багатопоточних систем (multi threading);
- вимоги до організації об'єктної моделі і методи її забезпечення;
- загальні принципи побудови об'єктно-орієнтованої моделі ПЗ і принципи її функціонування;
- принципи організації структур збереження даних і методи доступу до даних;
- відмінності між об'єктно-орієнтованим і функціональним програмуванням;
- етапи проектування об'єктно-орієнтованої моделі;
- принципи взаємодії об'єктів, які виконані на мові високого рівня Java;
- процес створення програм на мові Java;
- основні операції при роботі з рядками та масивами;
- арифметичні операції;
- операції порівняння;
- цикли в мові Java;
- створення класів у Java;
- компоненти класу, модифікатори компонент класу;
- елементи GUI-інтерфейсу;
- аплети в мові Java;
- програмування сокетів;
- робота з базою даних в Java.
- сутність понять IoC та DI;
- можливості фреймворку Spring;
- призначення фреймворку Spring Boot;
- застосування технології Thymeleaf в Spring Boot;
- використання технології FreeMarker та Groovy в Spring Boot;
- переваги використання технології Mustache;
- особливості використання Spring JDBC;

**вміти:**

- формулювати вимоги до об'єктно-орієнтованої моделі і забезпечувати її властивості;
  - розробляти об'єктно-орієнтовану модель у процесі технічного проектування на основі оціночних варіантів програмного забезпечення, вимог користувачів, аналізу технічних, економічних, функціональних, сервісних характеристик ПЗ, використовуючи науково-технічну, довідкову інформацію;
  - розробляти логічну структуру об'єктної моделі ПЗ у процесі технічного проектування за допомогою методу нормалізації відношень, використовуючи рівні абстракції даних, вимоги технічного завдання;
  - розробляти класи і зв'язок між ними в умовах технічного проектування за допомогою відповідного технічного і програмного забезпечення;
  - створювати віконне та консольне ПЗ;
  - створювати окремі компоненти ПЗ;
  - створити програму на крос-платформній мові програмування Java будь-якої складності;
  - реалізувати вирішення поставленої задачі за допомогою об'єктно-орієнтованого програмування;
  - створити сокетне з'єднання на протоколах TCP та UDP;
  - спроектувати зручний інтерфейс користувача;
  - реалізувати роботу програми з базою даних;
  - встановлювати та підключати фреймворк Spring Boot;
  - проводити початкові налаштування проекту в Spring Boot;
  - користуватись різними типами анотацій для налаштування класів в проектах;
  - працювати в Spring MVC;
  - користуватись можливостями Maven;
  - підключати бази даних за допомогою Spring JDBC;
  - комбінувати можливості Spring Boot та AngularJS.
- Компетентності, що забезпечує вивчення дисципліни:
- КЗ 1. Здатність до абстрактного мислення, аналізу та синтезу.
  - КЗ 2. Здатність застосовувати знання у практичних ситуаціях.
  - КС 3. Здатність до проектування, розробки, налагодження та вдосконалення системного, комунікаційного та програмноапаратного забезпечення інформаційних систем та технологій, Інтернету речей (IoT), комп'ютерно-інтегрованих систем та системної мережної структури, управління ними.

Результати у вигляді програмних результатів навчання:

ПРН 3. Використовувати базові знання інформатики й сучасних інформаційних систем та технологій, навички програмування, технології безпечної роботи в комп'ютерних мережах, методи створення баз даних та інтернет-ресурсів, технології розроблення алгоритмів і комп'ютерних програм мовами високого рівня із застосуванням об'єктно-орієнтованого програмування для розв'язання задач проектування і використання інформаційних систем та технологій.

### 3. Опис навчальної дисципліни

#### 3.1. Загальна інформація

Форма навчання	Рік підготовки	Семестр	Кількість		Кількість годин					Вид підсумкового контролю	
			кредитів	годин	лекції	практичні	лабораторні	самостійна робота	індивідуальні завдання		Курсова робота
Денна	1	5,6	8	240	60		60	102		18	Іспит

#### 3.2. Структура змісту навчальної дисципліни

Назви змістових модулів і тем	Кількість годин												
	денна форма						заочна форма						
	усього	у тому числі					усього	у тому числі					
1	2	л	п	лаб	інд	с.р.	8	л	п	лаб	інд	с.р.	13
<b>Змістовий модуль 1. Основи синтаксису</b>													
Тема 1.1. Вступ	5	1		1		3							
Тема 1.2. Середовище розробки джава-додатків Intelij IDEA	5	1		1		3							
Тема 1.3. Основний синтаксис мови Java.	8	2		2		4							
Тема 1.4. Умовні та циклічні оператори	8	2		2		4							
Тема 1.5. Робота з рядками String	8	2		2		4							
Тема 1.6. Робота з масивами Array	8	2		2		4							
Тема 1.7. Java Code Convention	8	2		2		4							
Тема 1.8. Функціональне програмування, Streams, Lambdas	8	2		2		4							
<i>Разом за ЗМ1</i>	<b>58</b>	<b>14</b>		<b>14</b>		<b>30</b>							
<b>Змістовий модуль 2. Об'єкто-орієнтоване програмування</b>													
Тема 2.1. Абстракція	4	1		1		2							
Тема 2.2. Інкапсуляція. Encapsulation	4	1		1		2							
Тема 2.3. Успадкування Inheritance	4	1		1		2							
Тема 2.4. Динамічне зв'язування. Поліморфізм	4	1		1		2							
Тема 2.5. Композиція та агрегація	4	1		1		2							
Тема 2.6. Принципи	8	2		2		4							

SOLID													
Тема 2.7. Колекції, List	7	2		2		3							
Тема 2.8. Колекції, Map	7	2		2		3							
Тема 2.9. Клас LocalDateTime	5	1		1		3							
Тема 2.10. Система вводу\виводу NIO	7	2		2		3							
Тема 2.11. Робота з базами даних, JDBC	8	2		2		4							
<i>Разом за ЗМ2</i>	<b>62</b>	<b>16</b>		<b>16</b>		<b>30</b>							
<b>Змістовий модуль 3. Базовий проект SpringBoot</b>													
Тема 1.1. Вступ	5	1		1		3							
Тема 1.2. Середовище розробки джава- додатків Intelij IDEA Ultimate Edition	5	1		1		3							
Тема 1.3. Основні положення фреймворку SpringBoot	8	2		2		4							
Тема 1.4. Архітектура проекту SpringBoot	8	2		2		4							
Тема 1.5. Побудова рівня MODEL	8	2		2		4							
Тема 1.6. Побудова рівня CONTROLLER	8	2		2		4							
Тема 1.7. Побудова рівня SERVICE	8	2		2		4							
Тема 1.8. Технологія REST API	8	2		2		4							
<i>Разом за ЗМ3</i>	<b>58</b>	<b>14</b>		<b>14</b>		<b>30</b>							
<b>Змістовий модуль 4. Різновиди використання</b>													
Тема 2.1. Зовнішній REST- клієнт POSTMAN	4	1		1		2							
Тема 2.2. Налаштування з'єднання з базою даних	4	1		1		2							
Тема 2.3. ORM- моделювання	4	1		1		2							
Тема 2.4. Побудова рівня REPOSITORY	4	1		1		2							
Тема 2.5. Тестування SpringBoot проекту.	4	1		1		2							
Тема 2.6. Побудова рівня ADMIN PANEL.	8	2		2		4							
Тема 2.7. Побудова рівня WEB- CONTROLLER	7	2		2		3							
Тема 2.8. Деплой проекту на зовнішній сервер	7	2		2		3							
Тема 2.9. Використання хмарних баз даних AWS.	5	1		1		3							

Тема 2.10. Одночасне використання різних баз даних.	7	2	2	3						
Тема 2.11. Створення вихідної технічної документації.	8	2	2	4						
<i>Разом за ЗМ4</i>	<b>62</b>	<b>16</b>	<b>16</b>	<b>30</b>						
<b>Усього годин</b>	<b>240</b>	<b>60</b>	<b>60</b>	<b>120</b>						

### 3.2.1. Теми семінарських або практичних, або лабораторних занять

№	Назва теми
1	Технологія виконання простої програми в середовищі JDK та Intelij IDEA.
2	Разработка власних та використання готових пакетів класів
3	Робота з рядками. Методи класу String.
4	Масиви та цикли. Умовний оператор.
5	Програмування обчислення значення інтегралу.
6	Побудова простого класу.
7	Використання успадкування методів батьківського класу.
8	Імплементация інтерфейсів
9	Перевірка класів на дотримання принципів SOLID.
10	Колекція List. Streams, Lamndas
11	Абстракція предметної області. Створення архітектури проекту.
12	Використання класу LocalDateTime
13	Робота з класом NIO. Зчитування файлу та запис у файл
14	Паралелізм. Використання потоків. Синхронізація.
15	Технологія створення базового проекту SpringBoot в середовищі JDK.
16	Технологія створення базового проекту SpringBoot в середовищі Intelij IDEA ULTIMATE.
17	Розробка архітектури проекту
18	Моделювання предметної області.
19	Створення рівня REST CONTROLLER.
20	Створення рівня WEB CONTROLLER.
21	Створення рівня SERVICE.
22	Імплементация CRUD інтерфейсу.
23	Створення ORM-оболонки.
24	Налаштування зв'язку з базою даних.
25	Створення рівня REPOSITORY.
26	Зовнішнє тестування REST API. Postman.
27	Внутрішнє тестування проекту. Junit, Mockito.
28	Створення WEB-інтерфейсу. Адмін-панель. Apache Freemarker.
29	Деплой проекту на зовнішній сервер.
30	Одночасна робота з кількома базами різних типів.

### 3.2.2. Зміст завдань для самостійної роботи

№	Назва теми
1	Типи даних Java.
2	Оператори циклу

3	Порівняльні оператори
4	Середовище розробки NetBeans
5	Середовище розробки Intelij IDEA
6	Середовище розробки Eclipse
7	Одно та двовимірні масиви.
8	Стандарт Java Code Convention.
9	Створення об'єктів класу . Патерн Factory
10	Створення об'єктів класу . Патерн Builder
11	Створення об'єктів класу . Патерн Singleton
12	Функціональне програмування. Predicate.
13	Імперативне програмування.
14	Зв'язок з базою даних MySQL.
15	Фреймворк Spring.
16	Spring MVC
17	SpringBoot
18	REST API technology.
19	SOAP technology.
20	Середовище розробки Eclipse
21	Generic CRUD inteface.
22	Стандарт SpringBoot Code Convention.
23	Під'єднання до бази даних PostgreSQL.
24	DAO та REPOSITORY технології.
25	Сценарії тестування Postman.
26	Документування SWAGGER.
27	Веб-інтерфейс MUSTASHE.
28	Хмарна база даних Google Cloud.

### 3.2.3. Зміст завдань для індивідуальної роботи

- 1) Аналітичне порівняння методів агрегації.
- 2) Порівняльний аналіз імперативного та функціонального програмування.
- 3) Порівняльний аналіз сортування методів **Arrays.sort** та **Bubble Sort**
- 4) Порівняльний аналіз сортування методів Arrays.sort та Selection Sort
- 5) Порівняльний аналіз сортування методів Arrays.sort та Insertion Sort
- 6) Порівняльний аналіз сортування методів Arrays.sort та Shuttle Sort
- 7) Порівняльний аналіз сортування методів Arrays.sort та Shell Sort
- 8) Порівняльний аналіз сортування методів Arrays.sort та Merge Sort
- 9) Порівняльний аналіз сортування методів Arrays.sort та Radix Sort
- 10) Порівняльний аналіз сортування методів Arrays.sort та Coun Sort
- 11) Фреймворк Spring. Spring MVC. SpringBoot.
- 12) REST API technology.
- 13) SOAP technology.
- 14) Generic CRUD inteface.
- 15) Стандарт SpringBoot Code Convention.
- 16) Під'єднання до бази даних PostgreSQL.
- 17) DAO та REPOSITORY технології.
- 18) Сценарії тестування Postman.
- 19) Документування SWAGGER.
- 20) Веб-інтерфейс MUSTASHE.
- 21) Хмарна база даних Google Cloud.



#### 4. Система контролю та оцінювання

Формами поточного контролю при вивченні курсу є:

- Усна відповідь студентів під час опитування на лекціях;
- Захист лабораторних робіт;
- Тестування з використанням платформи Moodle;
- Написання та захист рефератів.

Формами підсумкового контролю є Іспит;

Засобами оцінювання є:

- Модульні та поточні контрольні роботи;
- Тестувальна система на платформі Moodle;
- Реферати з тематики курсу;
- Виконання та захист лабораторних робіт.

#### Критерії оцінювання результатів навчання з навчальної дисципліни

Критерієм успішного проходження здобувачем освіти підсумкового оцінювання може бути досягнення ним мінімальних порогових рівнів оцінок за кожним запланованим результатом навчання навчальної дисципліни.

Мінімальний пороговий рівень оцінки варто визначати за допомогою якісних критеріїв і трансформувати його в мінімальну позитивну оцінку використовуваної числової (рейтингової) шкали).

#### Шкала оцінювання: національна та ЄКТС

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	
		для екзамену, курсового проекту (роботи), практики	для заліку
90 – 100	<b>A</b>	відмінно	зараховано
80 – 89	<b>B</b>	добре	
70 – 79	<b>C</b>		
60 – 69	<b>D</b>	задовільно	
50 – 59	<b>E</b>		
35 – 49	<b>FX</b>	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
0 – 34	<b>F</b>	незадовільно з обов'язковим повторним вивченням дисципліни	не зараховано з обов'язковим повторним вивченням дисципліни

#### Розподіл балів, які отримують студенти 5 семестр

Поточне тестування та самостійна робота																			Підсумковий тест (екзамен)	Сума
Змістовий модуль 1								Змістовий модуль 2												
T1	T2	T3	T4	T5	T6	T7	T8	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11		
2	4	2	4	4	5	4	5	2	4	4	2	4	4	4	4	4	4	4	30	100

## 6 семестр

Поточне тестування та самостійна робота																			Підсумковий тест (екзамен)	Сума	
Змістовий модуль 3								Змістовий модуль 4													
T1	T2	T3	T4	T5	T6	T7	T8	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11			
2	4	2	4	4	5	4	5	2	4	4	2	4	4	4	4	4	4	4	4	30	100

### 5. Рекомендована література

Сторінка курсу в Moodle <https://moodle.chnu.edu.ua/course/view.php?id=529>

#### 5.1. Базова (основна)

1. Васильєв О. Програмування мовою Java. Тернопіль: Навч. кн. – Богдан, 2020. 696с.
2. Nagos T. Beginning IntelliJ IDEA: Integrated Development Environment for Java Programming. Apress, 2022. 248p.
3. Horstmann C.S. Core Java, Vol. II-Advanced Feature, 12-th edition. Pearson, 2022. 944p.
4. Lowe D. Java All-in-One For Dummies, 7-th edition. Wiley, 2023. 912p.
5. Schildt H. Java: A Beginner's Guide, Ninth Edition, Ninth edition. McGraw Hill, 2022. 752p.
6. Weidig B. A Functional Approach to Java: Augmenting Object-Oriented Java Code with Functional Principles. O'Reilly Media, 2023. 275p.
7. Галкін О.В., Катеринич Л.О., Шкільняк О.С. Програмування на Java 8: Навчальний посібник для студентів факультету комп'ютерних наук та кібернетики. – Київ: ЛОГОС, 2017. – 186 с.
8. Кадомський К.К., Ніколюк П.К. Java. Теорія і практика : Навч. посіб. - Вінниця: Донну, 2019. 197 с.

#### 5.2. Допоміжна

1. Schildt H. Java: The Complete Reference. Twelfth Edition - NY: McGraw Hill Professional, 2021. - 1573 p.
2. І. Копитко М.Ф., Іванків К.С. Основи програмування мовою Java: Тексти лекцій. – Львів: Вид. центр ЛНУ ім. Івана Франка, 2002. 83 с.
3. Cheng F. Exploring Java 9. Apress, 2018. 179p.
4. Cosmina I. Java for Absolute Beginners. Apress, 2018. 623p.
5. Horstmann C.S. Core Java for the Impatient, Second Edition. Addison-Wesley Professional, 2018. 567p.
6. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. Introduction to the algorithms. – The MIT Press, 2001. – 1180p.

### 8. Інформаційні ресурси

1. [https://courses.prometheus.org.ua/courses/EPAM/JAVA101/2016\\_T2/about](https://courses.prometheus.org.ua/courses/EPAM/JAVA101/2016_T2/about)
2. <https://www.coursera.org/learn/java-object-oriented-programming>
3. <https://www.coursera.org/specializations/java-object-oriented>
4. <https://www.coursera.org/learn/java-introduction>
5. <https://www.coursera.org/specializations/core-java>
6. <https://www.coursera.org/specializations/object-oriented-programming>
7. <https://www.coursera.org/specializations/java-programming>