Міністерство освіти і науки України
Національний університет «Чернігівська політехніка»
Навчально-науковий інститут електронних та інформаційних технологій
Кафедра *інформаційних та комп'ютерних систем*

"**ЗАТВЕРДЖУЮ**"
Завідувач кафедри

В.М.Базилевич

"_____"_____2020 р.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

ОБ'ЄКТНО ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ (ОК21)

Освітня програма «*Інженерія програмного забезпечення*»

Рівень вищої освіти – *перший (бакалаврський)*

Спеціальність *121 – Інженерія програмного забезпечення*

Мова навчання: *українська*

Статус дисципліни: *обов'язкова*

Форма навчан.	Рік навч.	Сем.	Розподіл годин				Разом	За тиждень		ІНДЗ	Контр.
			Всього ауд.	Лек	Лаб.	СРС		Ауд.	СРС		
Денна	2	3	40	26	14	110	150	2.5	7.9	РГР	Е
	2	4	30	16	14	90	120	1.9	5.6	КП	Е

Чернігів – 2020 рік

Робоча програма *Об'єктно орієнтоване програмування*
(назва навчальної дисципліни)
для здобувачів вищої освіти галузі знань *12 – Інформаційні технології* спеціальності *121 – Інженерія програмного забезпечення*

Розробник робочої навчальної програми:
к.т.н., доцент кафедри інформаційних та комп'ютерних систем НУ «Чернігівська політехніка»

_____ *П.Г.Бивойно)*
(підпис) (прізвище та ініціали)

Робочу програму обговорено на засіданні кафедри *інформаційних та комп'ютерних систем*

Протокол від "27" *серпня 2020* року № *1*

Завідувач кафедри *інформаційних та комп'ютерних систем*

_____ (*В.М.Базилевич)*

УЗГОДЖЕНО:
Завідувач кафедри *ІТ та ПІ* _____ (*І.В.Білоус*)
(підпис) (прізвище та ініціали)

Abstract

ESIEIT/SE ОК21 Object-oriented Programming

2020/2021 Sem. 1(3), 2020/2021 Sem. 2(4)

Course Description

"Object-oriented programming" is discipline of university's obligatory cycle of educational and professional programs of higher education in the specialty 121 - "Software Engineering". This subject is as a part of programming training together with discipline " Fundamentals of programming " and " System programming " which studied in the previous years.

The **purpose** of presenting this discipline is to promote high professional qualities of future professional, mastering the latest techniques and software environment for object-oriented programming. The basis of discipline is learning by key concepts of OOP and their implementation in multilingual linguistic basis. The discipline gives special attention to practical training in the development of Java applications based on object-oriented methodology, particularly when working together in a team that meets of the specialist qualification characteristics requirements.

Contents: class, object, method, encapsulation, inheritance, polymorphism, Java, swing, interface, collections, streams, threads, events.

1 Опис навчальної дисципліни

<table>
<tr><th rowspan="2">Найменування показників</th><th rowspan="2">Галузь знань, напрям підготовки, освітньо-кваліфікаційний рівень</th><th colspan="2">Характеристика навчальної дисципліни</th></tr>
<tr><th colspan="2">денна форма навчання</th></tr>
<tr><td rowspan="2">Кількість кредитів – 9</td><td>Галузь знань
12 – Інформаційні технології</td><td colspan="2" rowspan="2">Нормативна</td></tr>
<tr><td></td></tr>
<tr><td>Модулів (семестрів) – 2</td><td rowspan="5">Спеціальність: 121 - Інженерія програмного забезпечення</td><td colspan="2">Рік підготовки:</td></tr>
<tr><td>Змістових модулів – 6</td><td>2-й</td><td>2-й</td></tr>
<tr><td>Індивідуальні науково-дослідні завдання – розрахунково-графічна робота, курсовий проект</td><td colspan="2">Семестри</td></tr>
<tr><td rowspan="2">Загальна кількість годин – 270</td><td>3-й</td><td>4-й</td></tr>
<tr><td colspan="2">Лекції</td></tr>
<tr><td rowspan="11">Тижневих годин:
аудиторних –
3 сем.–2.5; 4 сем. -1.9;
самостійної та індивідуальної роботи студента –
3 сем. – 6.9;
4 сем. – 5.7.</td><td rowspan="11">Освітньо-кваліфікаційний рівень: бакалавр</td><td>26 год.</td><td>16 год.</td></tr>
<tr><td colspan="2">Практичні, семінарські</td></tr>
<tr><td></td><td></td></tr>
<tr><td colspan="2">Лабораторні</td></tr>
<tr><td>14 год.</td><td>14 год.</td></tr>
<tr><td colspan="2">Самостійна робота</td></tr>
<tr><td>110 год.</td><td>90 год.</td></tr>
<tr><td colspan="2">Індивідуальні завдання:</td></tr>
<tr><td>РГР</td><td>КП</td></tr>
<tr><td colspan="2">Вид контролю:</td></tr>
<tr><td>Екзамен</td><td>Екзамен, КП</td></tr>
</table>

Примітка.
Співвідношення кількості годин аудиторних занять до самостійної і індивідуальної роботи становить:

у 3-ому семестрі 1:2,75;
у 4-ому семестрі 1:3.
Застосовані скорочення:
ООП – об’єктно-орієнтоване програмування.
RTTI – динамічна ідентифікація типів даних.
VMT – таблиця віртуальних методів.

Передумовою для вивчення дисципліни *“Об’єктно-орієнтоване програмування*” є успішне засвоєння дисциплін *“Основи програмування”*, *“Системне програмування”*, *“Операційні системи”*, та здобуті такі результати навчання, як поняття алгоритму, вміння складати програми мовою С, знання персонального комп’ютера.

Набуті під час вивчення дисципліни “*Об’єктно-орієнтоване програмування*” знання є базовими для вивчення таких дисциплін, як “*Імітаційне моделювання"*, *”Java і C# технології прикладного програмування"*. Набуті знання та вміння застосовуються також у виконанні курсових проектів в подальших семестрах та випускної бакалаврської роботи.

Обов’язковою умовою викладання дисципліни є проведення лабораторного практикуму із застосуванням персональних комп’ютерів та спеціалізованого інтегрованого середовища розробки програм, наприклад, Eclipse.

2 Мета навчальної дисципліни

Метою викладання навчальної дисципліни *“Об’єктно-орієнтоване програмування*” є закріплення та розвиток загальних та фахових компетентностей бакалавра в галузі знань *12 – Інформаційні технології*.

Зокрема, це такі загальні компетентності, як:

– ЗК1. Здатність до абстрактного мислення, аналізу та синтезу;
– ЗК2. Здатність застосовувати знання у практичних ситуаціях;
– ЗК3. Здатність спілкуватися державною мовою як усно, так і письмово;
– ЗК5. Здатність вчитися і оволодівати сучасними знаннями;
– ЗК6. Здатність до пошуку, оброблення та аналізу інформації з різних джерел;
– ЗК7. Здатність працювати в команді;
– ЗК31. Здатність працювати в міжнародному контексті.

Також такі фахові компетенції:

– ФК15. Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення
– ФК17. Здатність розробляти архітектури, модулі та компоненти програмних систем;
– ФК19. Здатність дотримуватися специфікацій, стандартів, правил і рекомендацій в професійній галузі при реалізації процесів життєвого циклу;
– ФК24. Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя;
– ФК27. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення;
– ФК28. Здатність до алгоритмічного та логічного мислення.

3 Очікувані результати навчання з дисципліни

Навчальна дисципліна "*Об'єктно-орієнтоване програмування*" має допомогти сформувати наступні програмні результати навчання:

– ПР01. Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки;

– ПР05 Знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізу та математичного моделювання для розробки програмного забезпечення;

– ПР06. Уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення;

– ПР07. Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення;

– ПР08. Вміти розробляти людино-машинний інтерфейс;

– ПР13. Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань;

– ПР15. Мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення;

– ПР16. Мати навички командної розробки, погодження, оформлення і 1 випуску всіх видів програмної документації;

– ПР23. Вміти документувати та презентувати результати розробки програмного забезпечення.

4 Критерії оцінювання результатів навчання

Відповідно до «Положення про поточне та підсумкове оцінювання знань здобувачів вищої освіти Національного університету «Чернігівська політехніка», погодженого вченою радою НУ «Чернігівська політехніка» (протокол № 6 від 31.08.2020 р.) та затвердженого наказом ректора НУ «Чернігівська політехніка» від 31.08.2020 р. №26., діючого в університеті екзаменаційна оцінка має дві складових:

- результат роботи у семестрі – вага 60%.
- результат отриманий безпосередньо на іспиті – вага 40%.

До іспиту допускаються студенти, що виконали усі заплановані на семестр завдання з підсумковою оцінкою не менше 60 балів (за 100 – бальною шкалою оцінювання). Окрім того, відповідно до «Положення…», такі студенти, за бажанням, можуть отримати оцінки «достатньо» та «задовільно» без іспиту.

Студенти, що мають підсумкову оцінку за семестр менше ніж 35 балів (за 100 – бальною шкалою оцінювання) потребують повторного вивчення дисципліни.

З тими студентами, які до проведення підсумкового семестрового контролю не встигли виконати всі обов'язкові види робіт та мають підсумкову оцінку за семестр від 35 до 59 балів (за 100 – бальною шкалою оцінювання), проводяться додаткові індивідуальні заняття, за результатами яких визначається рівень засвоєння матеріалу, та чи необхідне повторне вивчення дисципліни.

Дисципліну можна вважати такою, що засвоєна, якщо студент:

1) **знає**:

– базові поняття об’єктно-орієнтованого програмування (модуль, об’єкт, клас, інтерфейс, абстрактний клас);

– основні властивості об’єкта (стан, поведінка та унікальність);

– призначення та структуру класу;

– поняття про успадкування, поліморфізм та інкапсуляцію;

– клас Object, основні методи класу Object;

– основні контейнерні класи, огляд;

– основні класи вводу-виводу, огляд;

– основні класи для організації потоків;

– засоби обробки виключень;

2) **вміє:**

– проводити об’єктно-орієнтований аналіз предметної області і визначати абстракції з мінімальними необхідними властивостями;

– використовувати можливості середовища розробки для створення та подальшого запуску простого проекту;

– написати на Java специфікацію класу, який успадковує інший клас чи інтерфейс;

5 Засоби діагностики результатів навчання

Засобами оцінювання результатів навчання з дисципліни у1-му семестрі є:

– співбесіда із студентом по результатам лабораторних робіт;

– письмові тестові контрольні;

– РГР;

– іспит.

Засобом оцінювання результатів навчання з дисципліни у 2-му семестрі є:

– співбесіда із студентом по результатам лабораторних робіт;

– письмові тестові контрольні;

– іспит;

– курсовий проект.

6 Програма навчальної дисципліни

Семестр 1

Змістовий модуль 1. Швидкий старт

Тема 1. *Вступ*

Предмет та завдання курсу “Об’єктно-орієнтоване програмування”. Структура навчального курсу. Навчально-методична література з дисципліни. Складність програмного забезпечення та засоби боротьби з нею. Історія розвитку ООП. Чисте і «не чисте ООП». Короткі відомості про поняття об’єкт та клас та особливості окремих мов об’єктно-орієнтованого програмування.

Java – як зразок зваженої реалізації об’єктно-орієнтованого підходу до

програмування. Чому не C++? Історія виникнення та розвитку Java. Платформа Java. Середовище компіляції та середовище виконання Java.

Тема 2. ***Загальні відомості про мову Java***

Алфавіт мови. Ідентифікатори. Домовленості щодо створення імен. Види коментарів. Базові типи. Операції над базовими типами. Оператори. Масиви.

Тема 3. ***Загальна характеристика середовища розробки Eclipse***

Поняття проекту, пакету, класу. Стандартні панелі. Пісочниця. Автогенерація коду. Автодоповнення коду. Навігація.

Тема 4. ***Використання стандартної бібліотеки класів Java***

Основи роботи з класами. Методи класів та методи об’єктів. Створення об’єктів.

Утилітні класи System, Math, Arrays. Класи String, StringBilder, DecimalFormat. Класи оболонки.

Тема 5. ***Огляд бібліотеки Swing***

Класи бібліотеки Swing. Панелі, компоненти, менеджери компоновки, події компонентів.

Змістовий модуль 2. Основні концепції ООП

Тема 6. ***Поняття класу***

Класи, як засіб опису абстракцій реальних сутностей.

Ім’я класу та вирішення проблеми унікальності імен.

Вміст файлу класу. Оголошення пакету. Оголошення імпорту. Заголовок класу. Тіло класу. Класи файлу класу.

Вміст тіла класу.

Оголошення та ініціалізація полів. Статичні поля. Блоки ініціалізації.

Методи. Локальні змінні методів. Змінна this.

Конструктори. Конструктор super(). Життєвий цикл об’єкту. Створення та знищення об’єктів.

Різновиди класів за структурою – класи утиліти, класи фабрики об’єктів, класи структури даних.

Абстрактні класи та інтерфейси. Функціональні інтерфейси та дефолтні методи.

Внутрішні, локальні та анонімні класи.

Тема 7. ***Спадкування***

Визначення поняття спадкування. Переваги і недоліки використання механізм спадкування. Принцип підстановки LSP, «правильні» і «неправильні» форми спадкування. Особливості використання множинного спадкування.

Перевантаження методів та полів.

Особливості використання понять тип і клас.

Тема 8. ***Поліморфізм***

Визначення поняття поліморфізм. Антагонізм поліморфізму і типізації. Шляхи реалізації поліморфізму в Java. Поняття узагальненого коду. Поліморфні

змінні та методи. Переваги. Псевдо поліморфізм (перевантаження).

Тема 9. ***Інкапсуляція***

Визначення поняття інкапсуляції. Класичне та розширене поняття інкапсуляції. Специфікатори доступу. Переваги інкапсуляції.

Змістовий модуль 3/1. Реалізація специфічних технологій програмування

Тема 10. ***Технології обробку виняткових ситуацій***

Поняття про виняткову ситуацію і об'єкт-виняток. Викид винятку. Обробка винятку.

Тема 11. ***Технології створення та обробки подій***

Проблема взаємозв'язків між джерелом події та її слухачем. Шаблон проектування Observer. Механізм реалізації події на боці джерела та її обробки на боці слухача. Стандартні класи та інтерфейси для реалізації подій.

Тема 12. ***Робота з файлами в Java***

Огляд бібліотеки вводу виводу Потокові класи: форматне введення-виведення, маніпулятори, обмін інформацією з файлами.

Тема 13. ***Серіалізація об'єктів***

Поняття структури об'єкту та проблеми збереження інформації про нього. Класи для реалізації серіалізації. Приклад використання у лабораторній роботі.

Тема 14. ***Підсумки***

Підсумки першого семестру. Короткий огляд пройденого матеріалу.

Семестр 2

Змістовий модуль 3/2. Реалізація специфічних технологій програмування

Тема 15. ***Вступ***

Завдання на семестр. Огляд тем минулого семестру. Тест на збереження знань.

Тема 16. ***Технології передачі специфічних операцій користувачів у методи***

Постановка проблеми. Огляд шляхів вирішення проблеми у інших мовах програмування. Методика вирішення проблеми у мові Java. Дії розробника узагальненого методу. Дії користувача узагальненого методу.

Тема 17. ***Лямбда функції***

Огляд функціональних інтерфейсів. Створення та використання лямбда функцій

Тема 18. ***Технології послідовної обробки груп об'єктів***

Поняття про ітератор. Інтерфейси Enumeration, Iterator. Приклад реалізації ітератора.

Змістовний модуль 4. Контейнери

Тема 19. ***Прості колекції Java***

Огляд контейнерної бібліотеки Java. Інтерфейс Collections та List. Огляд та порівняльний аналіз класів, що реалізують інтерфейс List.

Поняття про генеріки у Java. Переваги та недоліки їх використання. Приклади використання генериків у стандартних класах Java.

Аутобоксінг, анбоксінг

Тема 20. ***Організація черг в Java***

Інтерфейси Queue та Deque. Огляд та порівняльний аналіз класів, що реалізують ці інтерфейси.

Тема 21. ***Організація множин в Java***

Інтерфейси Set, SortedSet, NavigableSet. Огляд та порівняльний аналіз класів, що реалізують ці інтерфейси.

Тема 22. ***Асоціативні масиви в Java***

Інтерфейси Map, SortedMap, NavigableMap. Огляд та порівняльний аналіз класів, що реалізують ці інтерфейси.

Тема 23. ***Проблеми обробки колекцій у багатопоточних застосуваннях***

Створення потокобезпечних колекцій методами класу Collections.

Класи колекцій пакету java.util.concurrent

Змістовний модуль 5. Потоки в Java

Тема 24. ***Основи багатопоточного програмування в Java***

Поняття процесу і потоку. Способи створення потоків. Методи управління виконанням потоку. Засоби керування взаємодією потоків.

Проблема сумісного використання даних. Синхронізація. Рівні безпечності потоків.

Тема 25. ***Бібліотека класів пакету java.util.concurrent***

Непрямі методи створення потоків. Пули потоків. Таймери. Черга подій графічного інтерфейсу. Повернення результатів із потоків. Інтерфейси Callable та Future.

Синхронізатори. Класи Semaphore, CountDownLatch, CycleBarrier, Phaser, Exchanger,

Тема 26. ***Бібліотека класів пакету java.util.concurrent.atomic***

Проблемі сумісної обробки даних декількома потоками. Поняття visibility, reordering, happened-before, специфікатор volatile.

Неблокуюча синхронізація на основі операції CAS.

Класи AtomicLong, AtomicDouble, LongAdder, DoubleAdder.

Змістовний модуль 6 Графіка

Тема 27. ***Базові поняття графіки Java***

Поняття графічного контексту. Методи рисування класів Graphics та Graphics2d. Методи трансформації зображень класу Graphics2d.

Клас java.awt.Color. Методи створення кольорів та простори кольорів.

Тема 28. ***Додаткові можливості роботи з графікою Java***

Клас BasicStroke та його використання

Інтерфейс Shape. Клас GeneralPath та його методи.

Афінні перетворення. Клас BuferedImage і робота із зображеннями.

Тема 29. ***Підсумки***

Огляд тем курсу. Перспективи використання ООП для вирішення наукових і практичних задач.

7 Структура навчальної дисципліни

Назви змістових модулів і тем		Кількість годин для денної/заочної форми навчання							
		Всього		У тому числі					
				Лек.		Лаб.		С.р.	
1	2	3	4	5	6	9	10	11	12
Семестр 3									
Змістовий модуль 1. Швидкий старт									
1	Вступ	4		2		1		1	
2	Загальні відомості про мову Java	7		2		1		4	
3	Огляд середовища розробки Eclipse	11		2		1		8	
4	Використання стандартної бібліотеки класів Java	10		1		1		8	
5	Огляд бібліотеки Swing	20		1		1		18	
	Разом за змістовим модулем 1	**52**		**8**		**5**		**39**	
Змістовий модуль 2. Основні концепції ООП									
6	Поняття об’єкту та класу	12		2		1		9	
7	Спадкування, форми спадкування	10		2		1		7	
8	Поліморфізм. Різновиди та необхідні передумови ООП поліморфізму. Узагальнений код.	20		2		1		17	
9	Модульність та інкапсуляція абстракцій	10		2		1		7	
	Разом за змістовим модулем 2	**52**		**8**		**4**		**40**	
Змістовий модуль 3/1. Реалізація специфічних технологій програмування у Java									
10	Технології обробки винятків	10		2		1		7	
11	Технології створення та обробки подій	10		2		2		6	
12	Робота з файлами в Java	10		2		1		7	
13	Серіалізація об’єктів	10		2		1		7	
14	Підсумки	6		2				4	
	Разом за змістовим модулем 3/1	**46**		**10**		**5**		**31**	
	Усього годин за семестр 3	**150**		**26**		**14**		**110**	

1	2	3	4	5	6	9	10	11	12
Семестр 4									
Змістовий модуль 3/2. Реалізація специфічних технологій програмування у Java									
15	Вступ. Завдання на семестр. Огляд тем минулого семестру	8		1		1		6	
16	Технології передачі операцій користувачів у методи	10		1		1		8	
17	Лямбда функції	9		1		1		7	
18	Технології послідовної обробки груп об’єктів.	10		1		2		7	
	Разом за змістовим модулем 3/2	**37**		**4**		**5**		**28**	
Змістовий модуль 4. Контейнери									
19	Прості колекції Java	7		1				6	
20	Організація черг в Java	8		1		1		6	
21	Організація множин в Java	8		1		1		6	
22	Асоціативні масиви в Java	9		1		2		6	
23	Проблеми обробки колекцій у багатопоточних застосуваннях	7		1				6	
	Разом за змістовим модулем 4	**39**		**5**		**4**		**30**	
Змістовий модуль 5. Потоки в Java									
24	Основи багатопоточного програмування в Java	10		1		1		8	
25	Бібліотека класів пакету java.util.concurrent	8		1		1		6	
26	Бібліотека класів пакету java.util.concurrent.atomic	9		2		1		6	
	Разом за змістовим модулем 5	**27**		**4**		**3**		**20**	
Змістовий модуль 6. Графіка у Java									
27	Базові поняття графіки Java	8		1		1		6	
28	Додаткові можливості роботи з графікою Java	8		1		1		6	
29	Підсумки	1		1					
	Разом за змістовим модулем 6	**17**		**3**		**2**		**12**	
	Усього годин за семестр 4	**120**		**16**		**14**		**90**	
	Усього годин за дисципліну	**270**		**42**		**28**		**200**	

8 Теми лабораторних занять

№ з/п	Назва теми	Кількість годин (д/з)
	Семестр 1	
1	Вступ	2
2	Знайомство з базовими типами та операторами мови “Java”	2
3	Знайомство з бібліотекою JRE	2
4	Об’єктна модель Java	2
5	Знайомство з бібліотекою Swing	2
6	Реалізація механізму обробки подій в Java	2
7	Потоки вводу-виведення і серіалізація об’єктів	2
	Разом за семестр 1	**14**
	Семестр 2	
8	Використання та реалізація ітераторів	2
9	Проведення експериментів з узагальненими методами, що дозволяють використання специфічних операцій користувача	2
10	Колекції Java	2
11	Карти відображення Java	2
12	Дослідження засобів створення потоків даних в Java	2
13	Організація взаємодії потоків, що виконуються паралельно	2
14	Робота з графікою	4
Разом за семестр2		**14**
Разом		

9 Самостійна робота

№ з/п	Назва теми	Кількість годин (д/з)
	Семестр 1	
1	Робота з конспектом лекцій, методичними вказівками, основною та додатковою літературою, підготовка до тестів	48
2	Підготовка до лабораторних занять	32
3	Виконання розрахунково-графічної роботи (контрольної роботи)	30
Разом за семестр 1		**110**
	Семестр 2	
1	Робота з конспектом лекцій, методичними вказівками, основною та додатковою літературою, підготовка до тестів	16
2	Підготовка до лабораторних занять	26
3	Виконання курсового проекту	48
Разом за семестр 2		**90**
Разом		**200**

10 Індивідуальні завдання

Робочим планом передбачено виконання індивідуальних завдань у вигляді розрахунково-графічної роботи у третьому семестрі та курсового проекту у четвертому семестрі. Вимоги до розрахунково-графічної роботи містяться у методичних вказівках до виконання РГР [14.3]. Виконавши РГР, студент має представити пояснювальну записку. Вимоги до курсового проекту містяться у методичних вказівках до виконання проекту [14.4]. Виконавши проект, студент має представити пояснювальну записку.

11 Методи контролю

Поточний контроль проводиться шляхом спілкування із студентами під час співбесід по темам лабораторних занять та тестів [14.1].

Семестровий контроль за результатами вивчення дисципліни у першому семестрі проводиться в останній атестаційний тиждень семестру. Бали за семестр обчислюються за результатами лабораторних робіт та тестових контрольних, як середнє арифметичне усіх результатів за 100-бальною шкалою. Отримане середнє значення множимо на 0.6, що і буде результатом за семестр. Оцінка виставляється у відповідну графу екзаменаційної відомості.

На іспиті студент готує відповіді на три питання білету і після цього спілкується з викладачем. Кожне питання оцінюється за 12-бальною шкільною системою. Ще до чотирьох балів студент може отримати, відповідаючи на додаткові питання. Таким чином студент може отримати до 40 балів.

Засобом оцінювання результатів навчання з дисципліни у 2-му семестрі є курсовий проект. Оцінка за проект виставляється як середнє арифметичне двох складових за 100 бальною шкалою:

– оцінки керівника проекту, що враховує своєчасність завершення проекту, його складність, якість коду та оформлення;

– оцінки комісії за презентацію результатів виконаних завдань та досліджень. Студент має зробити коротке повідомлення про тему роботи і шляхи вирішення завдань, які були поставлені у технічному завданні. Окрім того студент має дати відповіді на практичні і теоретичні питання.

12 Розподіл балів, які отримують студенти

Поточний контроль за результатами лабораторних робіт та тестів

		Кількість балів	
Усього		**0…**	**100**
1	Підготовленість до лабораторної роботи.	0…	10
2	Самостійність виконання лабораторної роботи.	0…	25
3	Своєчасність виконання лабораторної роботи.	0…	15
4	Повнота і якість оформлення звіту.	0…	6
5	Теоретичне питання	0…	22
6	Практичне завдання	0…	22

Тести		**0…**	**100**

Оцінка за виконання проекту

Вид роботи	Форма контролю	Кількість балів
Програмна частина	1. Відповідність умовам завдання 2. Відповідність вимогам стандартів	0… 20 0… 10
Пояснювальна записка	1. Обґрунтованість технічних рішень 2. Посилання на першоджерела 3. Відповідність оформлення вимогам 4. Своєчасність здачі	0… 20 0… 5 0… 15 0… 10
Захист курсового проекту	Самостійність виконання (відповіді на запитання)	0… 20
Разом		**0… 100**

Підсумковий контроль

Іспит	Кількість балів
1 Теоретичне питання із практичним завданням по темі питання	0… 12
2 Теоретичне питання із практичним завданням по темі питання	0… 12
3 Практичне завдання із практичним завданням по темі питання	0… 12
4 Додаткові питання	0… 4
Оцінка за іспит	**0… 40**

Шкала оцінювання: національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	
		для екзамену, курсової роботи	для заліку
90 – 100	**A**	відмінно	зараховано
82-89	**B**	добре	
74-81	**C**		
64-73	**D**	задовільно	
60-63	**E**		
35-59	**FX**	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
0-34	**F**	незадовільно з обов’язковим повторним вивченням дисципліни	не зараховано з обов’язковим повторним вивченням дисципліни

13 Інструменти, обладнання та програмне забезпечення, використання яких передбачає навчальна дисципліна

Лекційний матеріал подається у вигляді презентацій за допомогою медіа-проектора. Під час лекцій аналізуються проблемні ситуації, підтримується зворотний зв'язок з аудиторією шляхом формулювання запитань і стислих відповідей з обох сторін.

Особливістю виконання лабораторних робіт є застосування програмного забезпечення, а саме – Java 8 та IDE Eclipse.

14 Методичне забезпечення

1. Об’єктно орієнтоване програмування на Java. Конспект лекцій з дисципліни «Об’єктно орієнтоване програмування» для студентів спеціальностей 121 – «Інженерія програмного забезпечення», 123 – “Комп’ютерна інженерія”. /Укл.: Бивойно П.Г. – Чернігів: ЧНТУ, 2019. – 136 с.
2. Об’єктно орієнтоване програмування на Java. Методичні вказівки до лабораторного практикуму та самостійної роботи з дисципліни «Об’єктно орієнтоване програмування» для студентів спеціальності 121 – “Інженерія програмного забезпечення”. /Укл.: Бивойно П.Г., Бивойно Т.П. – Чернігів: ЧНТУ, 2018. – 108 с.
3. Організація взаємодії об’єктів, що працюють паралельно. Методичні вказівки до виконання курсового проекту з дисципліни «Об’єктно-орієнтоване програмування» для студентів напряму підготовки 6.050103 – „Програмна інженерія”. /Укл.: Бивойно П.Г., Бивойно Т.П. – Чернігів: ЧДНУ, 2015. – 50 с.
4. Реалізація поліморфної моделі дерева для багаторівневої структури даних. Методичні вказівки до виконання розрахунково-графічної роботи з дисципліни «Об’єктно-орієнтоване програмування» для студентів напрямів підготовки 6.050103 – „Програмна інженерія”. /Укл.: Бивойно П.Г., Бивойно Т.П. – Чернігів: ЧНТУ, 2017. – 60 с.

15 Рекомендована література

Базова

1. Казимир В.В. Об’єктно-орієнтоване програмування: навчальний посібник : рекомендовано МОН України/ В.В.Казимир К.: Слово, 2008.
2. Simon Kendal. Object oriented programming using Java. Ventus Publishing ApS-, 2009. – 209 с.
3. Програмування мовою Java. Олексій Васильєв. Видавництво: Навчальна книга - Богдан, 2020. -696 с.
4. Копитко М.Ф., Іванків К.С. Основи програмування мовою Java: Тексти лекцій. − Львів: Видавничий центр ЛНУ ім. Івана Франка, 2002.− 83 с.
5. Shildt Herbert. Java Complete guide. Eleventh edition. — McGraw-Hill Education, 2018. — 1248 p.
6. Object-Oriented Analysis and Design with Applications (3rd Edition) by Grady Booch. Addison-Wesley Professional, 2007

Допоміжна

1. Bloch Joshua. Effective Java. Third Edition. — Addison-Wesley, 2018. — 413 p.
2. Goetz Brain. Java concurrency in practice. Addison Wesley, 2010.
3. Gerbert Schildt. Java: The Complete Reference, Tenth Edition (Complete Reference Series) 10th Edition – McGraw-Hill, 2017.

16 Інформаційні ресурси

7. Повний відеокурс з програмування на Java. Український програміст 2019. https://programist.com.ua/video/java-programming/
1. Вікіпідручник «Освоюємо Java». https://uk.wikibooks.org/wiki/ Освоюємо Java
2. https://www.youtube.com/watch?v=TpxGzbn2_x4&list=PLyxk-1FCKqockmP-fXZmHQ7UlYP3qvZRa&index=1
3. https://eln.stu.cn.ua/login/index.php курс ООП ІПЗ