1. МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ
«ФАХОВИЙ КОЛЕДЖ ПРОМИСЛОВОЇ АВТОМАТИКИ
ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
ОДЕСЬКОГО НАЦІОНАЛЬНОГО ТЕХНОЛОГІЧНОГО УНІВЕРСИТЕТУ»

Циклова комісія комп'ютерних наук та інженерії програмного забезпечення

ЗАТВЕРДЖУЮ
директор ФКПАІТ ОНТУ
підписано Ольга ЄПУР
30.08.2023 року

ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ

обов'язкова

Освітньо-професійна програма Комп'ютерні науки

Код та найменування спеціальності 122 «Комп'ютерні науки»

Шифр та найменування галузі знань 12 «Інформаційні технології»

Мова навчання українська

2023

Розроблено та забезпечується: цикловою комісією Комп'ютерних наук та інженерії програмного забезпечення ВСП «Фаховий коледж промислової автоматики та інформаційних технологій Одеського національного технологічного університету»

Розробники:

Наталія ШВЕЦЬ, викладач вищої кваліфікаційної категорії ФКПАІТ ОНТУ

Розглянуто та схвалено на засіданні циклової комісії Комп'ютерних наук та інженерії програмного забезпечення

Протокол №01 від 28.08.2023 р.

Голова циклової комісії *підписано* Тетяна КОСТИРЕНКО
(підпис) (Власне ім'я, ПРІЗВИЩЕ)

Гарант освітньо-професійної програми *підписано* Наталія Швець
(підпис) (Власне ім'я, ПРІЗВИЩЕ)

Розглянуто та схвалено Методичною радою ФКПАІТ ОНТУ

Протокол №1 від 29.08.2023 р.

Голова Методичної ради ФКПАІТ ОНТУ *підписано* Вікторія ОКСАНІЧЕНКО
(підпис)

1. Пояснювальна записка

Вступ

Програма вивчення навчальної дисципліни «Об'єктно-орієнтоване програмування» складена відповідно до Стандарту фахової передвищої освіти із спеціальності 122 «Комп'ютерні науки» та освітньо-професійної програми «Комп'ютерні науки» підготовки фахових молодших бакалаврів.

Предмет вивчення

Предметом вивчення дисципліни «Об'єктно-орієнтоване програмування» є методологія об'єктно-орієнтованого аналізу, проєктування та програмування із реалізацією мовою Java.

Міждисциплінарні зв'язки

Попередні – «Основи програмування та алгоритмічні мови», «Основи програмної інженерії», послідовні — «Бази даних», «Конструювання програмного забезпечення», «Розробка клієнт-серверних застосунків».

Мета та завдання навчальної дисципліни

Метою викладання дисципліни «Об'єктно-орієнтоване програмування» є формування у здобувачів освіти системи теоретичних і практичних знань, вмінь і навичок використання об'єктно-орієнтованої технології програмування, ефективного використання її в процесі розробки інформаційних систем і комплексів.

Основними завданнями вивчення дисципліни «Об'єктно-орієнтоване програмування» є:

1. Вивчення основних принципів і понять, покладених в основу об'єктно-орієнтованого програмування:
   - — ознайомлення здобувачів освіти з базовими поняттями, такими як клас, елементи класу, об'єкти класу;
   - — розуміння основних принципів об'єктно-орієнтованого програмування: інкапсуляція, поліморфізм, успадкування і абстракція;
   - — надання знань про систему введення-виведення інформації, інтерфейси, анотації і узагальнення.
2. Навчання підходам щодо вирішення проблем в роботі програм:
   - — отримання практичних навичок роботи з відладчиком середовища розробки програм для тестуваня і відладки роботи програмного забезпечення;
   - — ознайомлення з класифікацією помилкових ситуацій;
   - — надання знань про способи передбачення і обробки виняткових ситуацій.
3. Надання знань про засоби зберігання інформації, введеної користувачем:
   - — ознайомлення з застосуванням файлів даних;
   - — вивчення інфраструктури колекцій.
4. Робота з програмним забезпеченням:

— проєктування та розробка програмного забезпечення із застосуванням об’єктно-орієнтованої парадигми програмування;
— застосування інструментальних засобів та інтегрованих середовищ для створення програмного забезпечення.

5. Використання засобів розв'язання задач:
— здатність застосовувати знання у практичних ситуаціях.
— самостійне створення програмних рішень для конкретних сценаріїв.

6. Акцент на критичному мисленні та оптимізації:
— розвиток навичок готувати і приймати рішення, виробляти у оцінку нових ідей, управляти процесами, проектами;
— здатність вибирати оптимальні підходи та алгоритми для вирішення конкретних задач.

7. Підготовка до роботи в команді:
— використання веб-сервісів для спільної розробки програмного забезпечення;
— розвиток навичок комунікації.

Ці завдання спрямовані на те, щоб здобувачі освіти отримали досвід у застосуванні об’єктно-орієнтованого підходу в програмуванні, зрозуміли його принципи і можливості та могли успішно застосовувати їх для створення програмних систем та їх супроводження;.

Компетентності та результати навчання

У результаті вивчення навчальної дисципліни «*Об’єктно-орієнтоване програмування*» здобувач освіти отримує наступні програмні компетентності та програмні результати навчання, які визначені в Стандарті фахової передвищої освіти із спеціальності 121 Інженерія програмного забезпечення (https://mon.gov.ua/storage/app/media/Fakhova%20peredvyshcha%20osvita/Zatverdzheni.standarty/2021/09/21/121-inzh.prohr.zabezp.21.09.docx) та освітньо-професійній програмі «Інженерія програмного забезпечення» (https://dev-kpa.fakel.com.ua/storage/uploads/4t0YvRV8MBZ1IXWV9i190ZBGF5H7rglXYysWLzuH.pdf) підготовки фахових молодших бакалаврів.

Загальні компетентності:

ЗК3. Здатність до абстрактного мислення, аналізу та синтезу.
ЗК4. Здатність застосовувати знання у практичних ситуаціях.
ЗК8. Здатність вчитися і оволодівати сучасними знаннями.

Спеціальні (фахові, предметні) компетентності:

СК2. Здатність використовувати теоретичні та фундаментальні знан-ня в галузі комп’ютерних наук та інформаційних технологій для ви-рішення різноманітних проблем.

СК3. Здатність розробляти, аналізувати та застосовувати ефективні алгоритми для розв'язання конкретних професійних задач залежно від предметного середовища.

СК4. Здатність здійснювати проєктування та розробку програмного забезпечення**.**

Програмні результати навчання:

РН04. Застосовувати сучасні методи математичного та комп'ютерного моделювання та будувати ефективні алгоритми для чисельного дослідження та розв'язання прикладних задач.

РН05. Пояснювати основні методи та технології об'єктно-орієнтованого та компонентного програмування.

РН17. Володіти навичками написання коду з використанням мов програмування/розмітки, визначення та маніпулювання даними.

2. Інформаційний обсяг навчальної дисципліни

2.1 Тематичний план

Таблиця 2.1 – Тематичний план навчальної дисципліни

№	Назва змістових модулів і тем
Змістовий модуль 1: Основи мови Java SE Основні принципи об'єктно-орієнтованого програмування. Успадкування. інтерфейси. Обробка виняткових ситуацій. Автоупакування/авторозпакування. Анотації	
Тема 1	Загальна характеристика мови. Основні принципи об'єктно-орієнтованого програмування
Тема 2	Основи введення/виведення даних
Тема 3	Класи та об'єкти
Тема 4	Клас String
Тема 5	Робота з файлами даних
Тема 6	Статичні дані та методи
Тема 7	Успадкування. Обмеження доступу до елементів класу
Тема 8	Динамічне зв'язування. Поліморфізм. Абстрактні класи
Тема 9	Інтерфейси. Спадкування інтерфейсів. Функціональні інтерфейси. Методи за замовчуванням
Тема 10	Обробка виняткових ситуацій
Тема 11	Перелічення. Автоупаковка. Анотації
Змістовий модуль 2: Узагальнення. Програмування графічного інтерфейсу користувача засобами JavaFX. Каркас колекцій Collections Framework	
Тема 12	Узагальнення: поняття, призначення. Параметри типів. Узагальнені класи. Обмеження типів. Використання метасимвольних аргументів
Тема 13	Узагальнені методи. Узагальнені інтерфейси
Тема 14	Лямбда-вирази
Тема 15	Основні поняття JavaFX. Скелет JavaFX-застосунку. Основи обробки

	подій в JavaFX
Тема 16	Колекції Collections Framework. Основні положення. Інтерфейси колекцій
Тема 17	Класи колекцій. Клас ArrayList. Доступ до колекцій через ітератор. Компаратори
Тема 18	Алгоритми колекцій. Масиви. Клас Arrays
Разом: 150 годин	

2.2 Зміст дисципліни

Змістовий модуль 1: Основи мови Java SE Основні принципи об'єктно-орієнтованого програмування

Тема 1. Загальна характеристика мови. Основні принципи об'єктно-орієнтованого програмування.

Особливості мови Java. Крос-платформне програмування. Поняття: транслятор, компілятор, транслюючий інтерпретатор, байт-код, інкапсуляція, абстракція, поліморфізм, успадкування

Тема 2. Основи введення/виведення даних.

Організація введення/виведення даних в мові Java. Потоки: байтові, символьні. Ієрархія класів потоків. Клас Scanner

Тема 3. Класи та об'єкти.

Об'єкти як дані типу клас. Склад класу. Синтаксис опису класу. посилання this

Тема 4. Клас String

Застосування класу String для роботи з рядками. Основні конструктори та методи класу.

Тема 5. Робота з файлами даних.

Ієрархії класів для роботи з файлами даних. Основні конструктори. Режими відкриття файлів даних на читання, запис, додавання інформації в кінець файлу

Тема 6. Статичні дані та методи.

Призначення, особливості статичних членів класу. Статичні даних і статичниі методи, особливості їх застосування. Статичний блок

Тема 7. Успадкування. Обмеження доступу до елементів класу.

Успадкування – засіб для побудови ієрархії класів. Права доступу до членів класу та їх особливості

Тема 8. Динамічне зв'язування. Поліморфізм. Абстрактні класи.

Відмінність статичного та динамічного зв'язування. Механізм поліморфізму. Особливості абстрактних класів.

Тема 9. Інтерфейси. Спадкування інтерфейсів. Функціональні інтерфейси. Методи за замовчуванням.

Інтерфейс як програмна конструкція. Призначення, склад інтерфейсів. Особливості функціональних інтерфейсів.

Тема 10. Обробка виняткових ситуацій.

Підходи щодо обробки виняткових ситуацій в мові Java. Оператори try, catch, finally.Бібліотечні класи виняткових ситуацій

Тема 11. Перелічення. Автоупаковка. Анотації.

Тип даних enum як особовий вид класу. Призначення автоупакування та авторозпакування. Призначення та використання анотацій.

Змістовий модуль 2: Узагальнення. Програмування графічного інтерфейсу користувача засобами JavaFX. Каркас колекцій Collections Framework

Тема 12. Узагальнення: поняття, призначення. Параметри типів. Узагальнені класи. Обмеження типів. Використання метасимвольних аргументів.

Призначення узагальнень. Синтаксис оголошення. Робота з параметризованими типами. Обмежені типи. Метасимвольні аргументи в методах.

Тема 13. Узагальнені методи. Узагальнені інтерфейси.

Узагальнені методи та інтерфейси: синтаксис, використання

Тема 14. Лямбда-вирази.

Лямбда-вирази: призначення, синтаксис, особливості використання.

Тема 15. Основні поняття JavaFX. Скелет JavaFX-застосунку. Основи обробки подій в JavaFX.

Розвиток механізму підтримки графічного інтерфейсу користувача. Бібліотека JavaFX. Обробка подій в Java

Тема 16. Колекції Collections Framework. Основні положення. Інтерфейси колекцій.

Призначення колекцій. Основні інтерфейси колекцій та їх методи .

Тема 17. Класи колекцій. Клас ArrayList. Доступ до колекцій через ітератор. Компаратори

Основні класи колекцій та їх призначення. Клас ArrayList – таблиця методів. Ітератори. створення ітераторів

Тема 18. Алгоритми колекцій. Масиви. Клас Arrays.

Основні алгоритми колекцій. Зв’язок між масивами та колекціями.

2.3 Перелік практичних робіт з дисципліни

Таблиця 2.2 – Тематичний план практичних робіт

№ з/п	Тема практичної роботи
1.	Розробка лінійних програм
2.	Програмування обчислювальних процесів з розгалуженням
3.	Циклічні конструкції мови Java
4.	Програми, які реалізують розгалуження. Оператор switch
5.	Розробка програм, які застосовують оператори переходу. Оператор continue, continue-мітка
6.	Розробка програм, які застосовують оператори переходу. Оператори

	break, break-мітка
7.	Розробка програм, які застосовують одновимірні масиви
8.	Класи та об'єкти. Робота з файлами даних
9.	Успадкування
10.	Поліморфізм
11.	Абстрактні класи
12.	Спадкоємство інтерфейсів. Реалізація в одному класі декількох інтерфейсів. Змінні інтерфейсу
13.	Пакети. Права доступу
14.	Обробка виняткових ситуацій. Принцип обробки
15.	Узагальнення
16.	Розробка програм з графічним інтерфейсом
17.	Реалізація меню
18.	Робота зі шрифтами
19.	Обробка подій
20.	Використання колекцій для збереження і управління набором даних
21.	Алгоритми колекцій. Робота з масивами за допомогою класа Arrays
22.	Форматоване виведення даних
23.	Робота з часом
	Всього: 50 годин

2.4 Кількість годин разом за програмою дисципліни

Розподіл дисципліни у годинах									
Курс	**I**		**II**		**III**		**IV**		**Всього**
Семестр	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	
Повний обсяг часу						150			150
Аудиторні заняття, годин						85			85
із них (кількість годин):									
лекції						35			35
практичні						50			50
Самостійна робота						65			65

3. Інструменти, обладнання та програмне забезпечення, використання яких передбачає дисципліна

1 Пакет розробника програм мовою Java. //Oracle: [сайт] https://www.oracle.com/cis/java/technologies/downloads/
2 Інтегроване середовище розробки IntelliJ Idea. // Jet Brains: [сайт]/ https://www.jetbrains.com/idea/download/
3 Інтегроване середовище розробки NetBeans. //NetBeans: [сайт] https://netbeans.apache.org/front/main/download/
4 Інтегроване середовище розробки Eclipse: //Eclipse Foundation [сайт] https://www.eclipse.org/downloads/
5 Бібліотека JavaFX SDK. //Gluon: [сайт] https://gluonhq.com/products/javafx/

4. Інформаційні ресурси

Базові (основні):

1 Васильєв О. М. Програмування мовою Java / О. М. Васильєв - М. Тернопіль: Навчальна книга - Богдан, 2020. - 696 с.
2 Кунгурцев О. Б. Основи програмування на мові Java. Середовище Net Beans: навчальний посібник для студентів вищих навчальних закладів / О. Б. Кунгурцев; за ред. Т.В. Ковалюк. - Одеса, 2016.-183 с.

Додаткові:

1 Java The Complete Reference, 9th Edition, Herbert Schildt..: Україна книга Language: English, 2017. — 1376 с.
2 Core Java Volume I--Fundamentals, 11th Edition/ K/ Horstman/Україна книга. Language: English, 2019. — 864 с.

5. Форма підсумкового контролю

Екзамен (VI семестр), поточний контроль.

6. Засоби діагностики результатів навчання

Перевірка та оцінювання знань здобувачів освіти може проводитись кількома методами:

1. Оцінювання знань здобувача освіти під час практичних занять.
2. Виконання індивідуальних навчально-дослідних завдань.
3. Захист практичних робіт.
4. Тестування.
5. Проведення поточно-модульного контролю.
6. Проведення екзамену.

6.1 Питання для самоконтролю

1. Загальна характеристика мови. Основні принципи об’єктно-орієнтованого програмування.:
    a. Що означає поняття «кросплатформеність»? Які бувають типи кросплатформеності?
    b. Що таке віртуальна машина? На які категорії поділяють віртуальні машини?
    c. Які основні принципи покладені в основу об’єктно-орієнтованого програмування? У чому полягає їхня сутність?
2. Основи введення/виведення даних:
    a. В основу введення-виведення інформації в сучасних мовах програмування покладені класи потоків. Що таке потік і які типи потоків використовуються в мові Java для реалізації введення/виведення?
    b. В чому полягає призначення класу Scanner і як його застосовувати для введення інформації?
    c. Які ієрархії класів призначені для забезпечення введення/виведення?
3. Класи та об’єкти:
    a. Що таке класовий тип даних? Яка структура класу?
    b. Що таке конструктор класу? Які бувають типи конструкторів?
    c. В чому полягає відмінність між поняттями «перевантаження методів» та «перевизначення методів»?
4. Клас String:
    a. Поясніть різницю між типом даних string в мові C і типом String в мові Java ?
    b. Класи String і StringBuffer призначені для роботи з рядками. В чому особливість в їх застосуванні?
    c. Як порівнювати рядки?
5. Робота з файлами даних:
    a. Які ієрархії класів забезпечують роботу з файлами даних в мові Java?
    b. Як відкриття файл даних на читання, запис, додавання інформації в кінець файлу?
    c. Які переваги надає оператор «try з ресурсами» при роботі з файлами даних?
6. Статичні дані та статичні методи класу:
    a. З якою метою додаються статичні елементи до складу класу?
    b. В чому полягає особливість роботи із статичними методами?
    c. Для чого призначений статичний блок класу?

7. *Обробка виняткових ситуацій*:
    a. Яка ситуація в роботі програми є винятковою? Як класифікують виняткові ситуації в мові Java?
    b. Як виконується обробка винятків?
    c. Як створити власний клас виняткової ситуації?
8. *Успадкування. Обмеження доступу до елементів класу*:
    a. Як реалізовано успадкування в мові Java?
    b. Поясніть особливості специфікаторів доступу «за замовчуванням», private, protected и public?
    c. Як впливають на доступ до членів класу специфікатори доступу «за замовчуванням», private, protected и public в залежності від їх розташування (різні пакети, класи-родичі, незалежні класи) ?
9. Узагальнення:
    a. Для чого призначені узагальнення, як вони оголошуються і використовуються?
    b. Як оголошуються і використовуються узагальнені методи та інтерфейси?
    c. Яку роль відіграють класи-оболонки при роботі з узагальненнями?
10. Каркас колекцій Collections Framework:
    a. Для чого призначена інфраструктура Collections Framework і на чому вона побудована?
    b. В чому переваги застосування колекцій для зберігання однотипних даних перед масивами і файлами даних?
    c. Що таке алгоритми колекцій?