

Міністерство освіти і науки України
Український державний університет імені Михайла Драгоманова

«ЗАТВЕРДЖЕНО»

на засіданні Вченої ради
УДУ імені Михайла Драгоманова
Протокол № 7 від

15 лютого 2024 р.

Проректор з навчально-
методичної роботи,
професор Роман Вернидуб



ПРОГРАМА
навчальної дисципліни

ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ

Освітньо-професійна програма	<i>Середня освіта (Інформатика)</i>
Галузь знань	<i>01 Освіта / Педагогіка</i>
Спеціальність	<i>014 Середня освіта</i>
Предметна спеціальність	<i>014.09 Інформатика</i>
Рівень вищої освіти	<i>перший (бакалаврський)</i>

Шифр за навчальним планом *ПП 2.11*

Київ – 2024

*Розробники
програми:*

Ігор Твердохліб – доцент кафедри інформаційних технологій і програмування Українського державного університету імені Михайла Драгоманова, кандидат педагогічних наук, доцент

Сергій Оніщенко – аспірант III року навчання спеціальності 014.09 «Середня освіта (Інформатика)» Українського державного університету імені Михайла Драгоманова

Рецензенти:

Петро Малезик – доцент кафедри комп'ютерної та програмної інженерії УДУ імені Михайла Драгоманова, доктор педагогічних наук, доцент

Тарас Сіткар – доцент кафедри комп'ютерних технологій Тернопільського національного педагогічного університету імені Володимира Гнатюка, кандидат педагогічних наук, доцент

Обговорено та рекомендовано до видання на засіданні Науково-методичної ради Українського державного університету імені Михайла Драгоманова «27» лютого 2024 року, протокол № 6.

Голова НМР Університету



Роман Вернидуб

(Ім'я, прізвище)

I. Пояснювальна записка

Програма нормативної навчальної дисципліни «*Об'єктно-орієнтоване програмування*» складена відповідно до освітньо-професійної програми «*Середня освіта (Інформатика)*» підготовки фахівців освітнього рівня бакалавр, галузі знань «01 Освіта / Педагогіка», спеціальності «014.09 Середня освіта (Інформатика)».

Предметом вивчення навчальної дисципліни «*Об'єктно-орієнтоване програмування*» є засоби об'єктно-орієнтованого програмування мови Java.

Міждисциплінарні зв'язки. Дисципліна «*Об'єктно-орієнтоване програмування*» розрахована на студентів, які вивчили дисципліни «Основи сучасної інформатики», «Алгоритми і структури даних», «Логічні основи інформатики», «Програмування».

Вивчення дисципліни забезпечує необхідний рівень знань для опанування наступних курсів циклу програмної інженерії: «Програмування графічного інтерфейсу», «Створення додатків для мобільних пристроїв» «Кросплатформне програмування».

Мета і завдання навчальної дисципліни.

Метою вивчення дисципліни «*Об'єктно-орієнтоване програмування*» є формування у студентів, які навчаються за освітньо-професійною програмою «Середня освіта (Інформатика)», знань засобів мови програмування Java для розв'язування задач довільної складності з використанням методології, що базується на об'єктно-орієнтованій парадигмі.

Для досягнення мети вивчення дисципліни «*Об'єктно-орієнтоване програмування*» потрібно розв'язати такі **завдання**:

- ✓ розкрити місце і значення парадигми об'єктно-орієнтованого програмування в загальній і професійній освіті;
- ✓ з'ясувати психолого-педагогічні аспекти засвоєння предмету, взаємозв'язки курсу з іншими навчальними курсами циклу програмної інженерії;
- ✓ сприяти розвитку алгоритмічного мислення, навичок системного підходу до вивчення програмування;
- ✓ розширити знання студентів про сучасні методології та парадигми програмування;
- ✓ сформувати у студентів знання, навички та уміння аналізувати зміст поставлених задач, створювати об'єктну модель задачі, формалізувати вхідні та вихідні дані, обирати доцільні структури даних, будувати відповідні ієрархії класів;
- ✓ сформувати у студентів знання, навички та уміння оцінювати ефективність алгоритмів, створювати алгоритми для розв'язування задач за допомогою комп'ютера.

Основні програмні результати навчання і компетентності згідно з вимогами освітньо-професійної програми:

Результати навчання	Компетентності
<p>ПРН2. Знати методи розробки та дослідження алгоритмів розв'язування задач з інформатики, знати методики оцінювання ефективності алгоритмів; володіти мовами програмування різних видів, розуміти їх переваги для розв'язання базових задач інформатики.</p> <p>ПРН23. Здатність використовувати ІКТ для подання, редагування, збереження та перетворення текстових, числових, графічних, звукових та відео даних на уроці, в позакласній і позашкільній роботі; виважено добирати й застосовувати відповідне програмне забезпечення для розв'язування прикладних задач.</p> <p>ПРН24. Здатність розробляти алгоритми розв'язування задач з інформатики, аналізувати складність та ефективність алгоритмів; реалізовувати алгоритми мовами програмування.</p> <p>ПРН42. Володіння основами професійної мовленнєвої культури.</p> <p>ПРН45. Презентувати, обговорювати та захищати власні погляди в усній, письмовій формах та за допомогою ІКТ.</p>	<p>ЗК3. Здатність вчитися та оволодівати сучасними знаннями.</p> <p>ЗК4. Здатність до пошуку, опрацювання, аналізу та критичного оцінювання даних з різних джерел.</p> <p>ЗК11. Здатність грамотно спілкуватися державною мовою як усно, так і письмово.</p> <p>ФК4. Здатність застосовувати теоретичні знання, демонструвати розуміння наукових фактів, концепцій, теорій, принципів і методів, що стосуються інформатики та ІКТ, при вирішенні професійних завдань.</p> <p>ФК7. Здатність оволодівати навичками використання сучасних ІКТ і застосовувати їх в практичній та професійній діяльності.</p>

II. Орієнтовний тематичний план

Дисципліна «Об'єктно-орієнтоване програмування» за навчальним планом підготовки бакалавра належить до циклу професійної підготовки (нормативна частина), на вивчення якої відводиться 3 кредити ЄКТС (90 навчальних годин). Значну кількість годин відведено на самостійну навчально-пізнавальну діяльність студентів, що полягає у підготовці до аудиторних занять, виконанні завдань, що пропонуються на лекційних та лабораторних заняттях, виконанні індивідуальних завдань, підготовці до модульного контролю, тощо.

<i>Назви модулів і тем</i>	Кількість годин (денна форма навчання)				Кількість годин (заочна форма навчання)			
	Аудиторні	Лекції	Лабораторні	СРС	Аудиторні	Лекції	Лабораторні	СРС
<i>Всього</i>	46	16	30	44	12	6	6	78
<i>Модуль 1. Об'єктно-орієнтоване програмування мовою Java.</i>								
Тема 1. Огляд засобів для об'єктно-орієнтованого програмування.	4	2	2	2				6
Тема 2. Огляд структурних засобів мови.	6	2	4	6				12
Тема 3. Класи і об'єкти.	6	2	4	6	1	1		11
Тема 4. Успадкування і поліморфізм.	6	2	4	6	3	1	2	9
Тема 5. Пакети і інтерфейси.	6	2	4	6	1	1		11
Тема 6. Винятки.	6	2	4	6	1	1		11
Тема 7. Багатопоточне програмування.	6	2	4	6	3	1	2	9
Тема 8. Узагальнення і лямбда-вирази.	6	2	4	6	3	1	2	9
<i>Всього за модулем 1:</i>	46	16	30	44	12	6	6	78

На лекційних заняттях розглядаються основи об'єктно-орієнтованого програмування, зокрема формалізація вхідних та вихідних даних поставленої задачі, побудова об'єктної моделі задачі, опис та використання класів даних, принципи об'єктно-орієнтованого програмування, елементи багатопоточного програмування; систематизуються та узагальнюються знання, навички та уміння, набуті при вивченні суміжних дисциплін.

На лабораторних заняттях приділяється значна увага практичним навичкам опису та створення ієрархії класів, використання принципів об'єктно-орієнтованого програмування. Метою практичних занять є розвиток у студентів алгоритмічного мислення, навичок самостійного використання набутих знань, навичок та умінь і забезпечення засвоєння основних понять навчальної дисципліни.

Вивчення навчальної дисципліни супроводжується використанням запропонованої навчально-методичної літератури та активним використанням сучасних інформаційних технологій і мережі «Інтернет».

III. Зміст навчальної дисципліни за модулями і темами

Зміст курсу «Об'єктно-орієнтоване програмування» подано у вигляді модуля, до якого наведено тематику занять та анотації до них.

Модуль 1. Об'єктно-орієнтоване програмування мовою Java.

Тема 1. ОГЛЯД ЗАСОБІВ ДЛЯ ОБ'ЄКТНО-ОРІЄНТОВАНОГО ПРОГРАМУВАННЯ.

Історія створення мови програмування Java. Призначення та основні характеристики мови. Інструментальні засоби програмування. Принципи об'єктно-орієнтованого програмування. Структура проєкта, програми. Особливості компіляції програми.

Тема 2. ОГЛЯД СТРУКТУРНИХ ЗАСОБІВ МОВИ.

Прості типи даних: цілі, дійсні, символні, логічні. Особливості літералів. Операції: надання значення, арифметичні, побітові, відношення, логічні. Інструкції управління порядком виконання програми: слідування – оператор-вираз, оператор об'єднання; розгалуження – умовного виконання, вибору; повторення: з лічильником, з передумовою, з післяумовою, з переліком. Передавання управління – виклик методу, вихід з методу (перехід до точки його виклику), вихід з програми, вихід з інструкцій повторення або вибору, продовження ітерації повторення. Засоби консольного введення/виведення. Масиви.

Тема 3. КЛАСИ І ОБ'ЄКТИ.

Поняття об'єкта. Поля, методи. Поняття класу. Синтаксис оголошення класів. Оголошення об'єктів. Синтаксис оголошення методу. Параметри методу. Аналіз передавання аргументів. Конструктори. Перевантаження конструкторів. Управління доступом. Вкладені і внутрішні класи. Використання аргументів командного рядка. Опрацювання довільної кількості аргументів. Рядковий клас.

Тема 4. УСПАДКУВАННЯ І ПОЛІМОРФІЗМ.

Основи успадкування. Поняття батьківського класу і класу нащадка. Створення ієрархії класів. Перевизначення методів. Динамічна диспетчеризація методів. Використання абстрактних класів.

Тема 5. ПАКЕТИ І ІНТЕРФЕЙСИ.

Поняття пакета. Доступ до членів класу. Імпорт пакетів. Поняття інтерфейсу. Визначення і реалізація інтерфейсу. Вкладені інтерфейси. Застосування інтерфейсів. Стандартні, статичні і закриті методи інтерфейсу.

Тема 6. ВИНЯТКИ.

Поняття винятку. Основи опрацювання винятків. Типи винятків. Вбудовані винятки. Створення власних винятків. Застосування винятків.

Тема 7. БАГАТОПОТОЧНЕ ПРОГРАМУВАННЯ.

Потокова модель: пріоритети потоків, синхронізація, обмін повідомленнями. Головний потік. Створення потоку, множини потоків. Взаємодія між потоками. Зупинка, поновлення і завершення потоків. Отримання стану потоку. Використання багатопоточності.

Тема 8. УЗАГАЛЬНЕННЯ І ЛЯМБДА-ВИРАЗИ.

Параметризовані класи. Узагальнені методи, конструктори, інтерфейси. Ієрархія узагальнених класів. Поняття лямбда-виразу. Функціональні інтерфейси. Блокові лямбда-вирази. Передавання лямбда-виразів як аргументи. Лямбда-вирази і винятки. Посилання на методи.

IV. Засоби діагностики успішності навчання

За результатами роботи на лабораторних заняттях, виконання завдань для самостійного опрацювання, підготовки та виступу з доповіддю на заняттях, модульних тестів, захисту результатів лабораторних робіт, студенти накопичують певну кількість балів, відповідно до якої відбувається оцінювання якості їх навчальних досягнень.

Побудова програми за блочно-модульною схемою спрямована на максимальну індивідуалізацію процесу навчання. Структура програми дібрана так, щоб надати студентам можливість навчатись в індивідуальному темпі та орієнтуватись на певні рівні вимог щодо засвоєння навчального матеріалу.

На лекційних заняттях розглядаються фундаментальні теоретичні питання; систематизуються та узагальнюються знання, навички й уміння, набуті під час вивчення суміжних дисциплін.

На лабораторних заняттях студенти знайомляться із сучасними програмними і апаратними засобами, набувають уміння і навички роботи із ними. Одним із основних завдань під час проведення лабораторного практикуму є набуття умінь та навичок розв'язувати задачі за допомогою комп'ютера. Метою проведення лабораторних занять є розвиток у студентів навичок самостійного використання набутих знань, навичок та умінь і забезпечення засвоєння основних понять навчальної дисципліни.

Викладання навчального курсу супроводжується з використанням системи управління навчальними ресурсами MOODLE, де розміщені навчально-методичні та дидактичні матеріали до курсу: список навчально-методичної літератури; конспекти (відеозаписи) лекцій, завдання до лабораторних та самостійних робіт; тестові завдання до теоретичного матеріалу, модульного та підсумкового тестування; доступ до спеціального програмного забезпечення; посилання на ресурси неформальної освіти та ін..

На консультаціях зі студентами передбачається з'ясування і обговорення проблемних питань, що стосуються виконання самостійних завдань до лекційних і лабораторних занять, незрозумілих студентами теоретичних питань тощо. Реалізація вищезгаданих вимог забезпечує один з головних напрямків професійної підготовки сучасного фахівця і дозволяє йому активно використовувати сучасні цифрові технології в професійній діяльності.

Контроль знань студентів здійснюється за модульно-рейтинговою системою. Навчальна діяльність студентів протягом семестру оцінюються за 100-бальною системою відповідно до рейтингового регламенту факультету.

Рейтинг студента з дисципліни складається з балів, які зараховуються за виконання та захист лабораторних.

Основним засобом діагностики успішності навчання упродовж семестру є лабораторна робота, яку студент самостійно виконує і захищає на лабораторному занятті.

V. Форма підсумкового контролю успішності навчання – залік

Відмітка „зараховано” виставляється студенту за умови набору більше 60 рейтингових балів, а саме:

- ✓ регулярного відвідування лекційних і лабораторних занять або їх негайного відпрацювання, своєчасного складання усіх видів поточного контролю з позитивними результатами;
- ✓ поглибленні набутих знань у процесі самостійної роботи;
- ✓ засвоєнні змісту навчального курсу в обсязі, передбаченому галузевим стандартом вищої освіти.

Якщо студент з поважних причин, що підтверджено документально, був відсутній на заняттях, він має право на одне перескладання з можливістю отримання максимальної кількості балів. Термін перескладання визначається викладачем.

Якщо впродовж семестру студент пропустив значну кількість занять, не має оцінок за виконання модулів, у відповідних графах „Відомості обліку успішності КМСОНП” виставляються „0”, у графі «залік» виставляється «не зараховано».

VI. Інформаційні джерела для вивчення курсу

Основні:

1. Васильєв О.М. Програмування мовою Java. Тернопіль: Навчальна книга – Богдан, 2019. 696 с.
2. Галкін О.В., Катеринич Л.О., Шкільняк О.С. Програмування на Java 8: Навчальний посібник для студентів факультету комп'ютерних наук та кібернетики. К.: ЛОГОС, 2017. 186 с.
3. Блинов И.Н., Романчик В.С. Java from EPAM: учеб.-метод. пособие. Минск: Четыре четверти, 2020. 560 с.
4. Schildt Н. Java: The Complete Reference, Twelfth Edition, Fully updated for Java SE 17. McGraw-Hill Companies, 2021. 1280 Pages.

Додаткові:

5. Урма Р., Фуско М., Майкрофт А. Современный язык Java. Лямбда-выражения, потоки и функциональное программирование. СПб.: Питер, 2020. 592 с.
6. Блох Д. Java: эффективное программирование, 3-е изд.: Пер. с англ. СПб.: ООО "Диалектика", 2019. 464 с.
7. Гетц Б., Пайерлс Т., Блох Д., Боубер Д, Холмс Д., Ли Д.. Java. Concurrency на практике. СПб.: Питер, 2020. 464 с.
8. Прохоренок Н. А. JavaFX. СПб.: БХВ-Петербург, 2020. 768 с.
9. Эванс Б., Флэнаган Д. Java. Справочник разработчика, 7-е изд. : Пер. с англ. СПб. "Диалектика", 2019. 592 с.
10. Sedgewick R., Wayne K. Introduction to Programming in Java: An Interdisciplinary Approach. Addison-Wesley. 2007

Інтернет ресурси:

11. Документація Java. URL: <https://docs.oracle.com/en/java/index.html>
12. Посібник з програмування Java. URL: <https://metanit.com/java/tutorial/>.
13. Java Tutorial for Beginners. URL: <https://vertex-academy.com/tutorials/en/java-tutorial-beginners/>