

Робоча програма з дисципліни Об'єктно-орієнтоване програмування для підготовки фахових молодших бакалаврів за освітньо-професійною програмою для IV курсу Обслуговування комп'ютерних систем і мереж спеціальності 123 Комп'ютерна інженерія.

РОЗРОБНИКИ ПРОГРАМИ: Олена Ленченко – спеціаліст вищої категорії, старший викладач

Робочу програму схвалено на засіданні циклової комісії комп'ютерно-інтегрованих технологій

Протокол №1 від «27» серпня 2021р.

Голова циклової комісії  Людмила ГЛУШКО

Розглянуто і рекомендовано до затвердження навчально-методичною радою коледжу

Протокол № 1 від «30» серпня 2021р.

Голова НМР  Аліна МАРКОВА

Зміст

Пояснювальна записка	4
Навчально-тематичний план дисципліни	8
Календарно-тематичний план дисципліни	8
Теми і плани лекційних занять	11
Теми і плани лабораторних занять	16
Теми для самостійної роботи студентів	23
Методи активізації навчального процесу	26
Система поточного і підсумкового контролю знань	26
Критерії оцінювання результатів навчання студентів	30
Рекомендована література	31
ДОДАТКИ	

Пояснювальна записка

Актуальність даної програми сьогодні.

Об'єктно-орієнтована технологія створення програмного забезпечення була задумана та розроблена як інструмент подолання складності користування. Вона успадкувала всі найкращі надбання структурного та модульного програмування, використавши їх для реалізації ряду принципово нових підходів до проектування програмного забезпечення. Головним завданням об'єктно-орієнтованого підходу є забезпечення способу структурування програми та керування складними взаємозв'язками між великою кількістю компонентів системи.

Мета: отримання студентами знань в області розробки програм зі застосуванням об'єктно-орієнтованих технологій. Оволодіння такими знаннями дозволить реалізовувати задачі автоматизації обробки інформації, автоматизації керування об'єктами.

Завдання: використання і застосування об'єктно-орієнтованих технологій програмування як при подальшому навчанні, так і після отримання фахової передвищої освіти у своїй професійній діяльності.

Процес вивчення дисципліни ВНПП03.01.02 Об'єктно-орієнтоване програмування спрямований на формування елементів наступних компетентностей:

а) загальні компетентності (КЗ):

КЗ 3. Здатність до абстрактного мислення, аналізу та синтезу.

КЗ 4 Здатність спілкуватися державною мовою як усно, так і письмово.

КЗ 7. Здатність до пошуку, оброблення та аналізу інформації з різних джерел та практичного її застосування.

КЗ 8. Здатність вчитися і бути сучасно навченим

б) спеціальні (фахові) компетентності (КФ):

КФ 6. Здатність застосовувати засоби сучасних мов програмування, основи структур даних для розробки програмного забезпечення.

КФ 7. Здатність використовувати професійно-орієнтовані знання і практичні навички з дисциплін циклу професійної та практичної підготовки для проектування, побудови та обслуговування сучасних комп'ютерних мереж різного виду та призначення.

КФ 8. Здатність використовувати знання сучасних технологій та інструментальних засобів розробки складних програмних систем (інженерії програмного забезпечення), уміння їх застосовувати на всіх етапах життєвого циклу розробки.

КФ 9. Здатність брати участь в модернізації та реконструкції апаратних та програмних засобів комп'ютерної інженерії, зокрема з метою підвищення їх ефективності.

КФ 11. Здатність до ділових комунікацій у професійній сфері, знання основ ділового спілкування, навички роботи в команді.

КФ 14. Здатність аргументувати вибір методів розв'язування спеціалізованих задач, критично оцінювати отримані результати, обґрунтовувати прийняті рішення.

в) очікувані результати навчання:

РН 1. Знати способи аналізу, синтезу та подальшого сучасного навчання. Вміти проводити аналіз інформації, приймати обґрунтовані рішення, вміти придбати сучасні знання. Встановлювати відповідні зв'язки для досягнення цілей. Нести відповідальність за своєчасне набуття сучасних знань.

РН 2. Мати спеціалізовані концептуальні знання, набуті у процесі навчання. Вміти розв'язувати складні задачі і проблеми, які виникають у професійній діяльності. Зрозуміле і недвозначне донесення власних висновків, знань та пояснень, що їх обґрунтовують, до фахівців та нефахівців. Відповідати за прийняття рішень у складних умовах.

РН 3. Мати глибокі знання із структури професійної діяльності. Вміти здійснювати професійну діяльність, що потребує оновлення та інтеграції знань. Здатність ефективно формувати комунікаційну стратегію у професійній діяльності. Нести відповідальність за професійний розвиток, здатність до подальшого професійного навчання з високим рівнем автономності.

PH 4. Знати види та способи адаптації, принципи дії в новій ситуації. Вміти застосувати засоби саморегуляції, вміти пристосовуватися до нових ситуацій (обставин) життя та діяльності. Встановлювати відповідні зв'язки для досягнення результату. Нести відповідальність своєчасне використання методів саморегуляції.

PH 5. Знати тактики та стратегії спілкування, закони та способи комунікативної поведінки. Вміти приймати обґрунтоване рішення, обирати способи та стратегії спілкування для забезпечення ефективної командної роботи. Нести відповідальність за вибір та тактику способу комунікації.

PH 6. Мати досконалі знання державної мови та базові знання іноземної мови. Вміти застосовувати знання державної мови, як усно так і письмово, вміти спілкуватись іноземною мовою. Використовувати при фаховому та діловому спілкуванні та при підготовці документів державну мову. Використовувати іноземну мову у професійній діяльності.

PH 12. Вміти застосовувати базові знання стандартів в області інформаційних технологій при розробці та впровадженні інформаційних систем і технологій

PH 14. Володіти навиками аналізу навчальної і спеціальної літератури, нормативних положень, технічної документації для вирішення проблем, що виникають у професійній діяльності.

PH 17. Знати сучасні методи побудови та аналізу ефективних алгоритмів, основи теорії чисельних методів, вміти аналізувати, оцінювати та вибирати існуючі алгоритми, розробляти нові алгоритми, які пов'язані з проектуванням апаратних та програмних компонент КСМ

PH 19. Володіти засобами сучасних мов програмування для створення програмних продуктів.

PH 28. Вміти опановувати та розробляти документацію на системи, продукти і сервіси інформаційних технологій, професійно спілкуватись рідною та англійською мовою

PH 29. Вміти використовувати існуючі та розробляти нові математичні методи, використовувати відповідні закони фізики для вирішення задач, пов'язаних з проектуванням та використанням КСМ.

PH 34. Уміти застосовувати засоби сучасних мов програмування під час програмної реалізації алгоритмів професійних задач.

PH 42. Уміння застосовувати технології та інструментальні засоби проектування для створення програмних систем.

PH 45. Практично володіти рідною та однією з іноземних мов в обсязі тематики, зумовленої професійними потребами.

PH 46. Використовувати відповідну термінологію у власних дослідженнях та професійній діяльності державною мовою та/або іноземною; спілкуватися в діалоговому режимі в галузі професійної діяльності; вміти презентувати результати власних досліджень та описувати їх у фахових публікаціях, використовуючи сучасні інформаційні та комунікативні технології

PH 48. Вдосконалювати професійний та особистісний розвиток протягом усього життя з метою поглиблення набутих та здобуття нових фахових знань, удосконалення креативного мислення.

PH 49. Критично осмислювати основні теорії, принципи, методи і поняття у навчанні та професійній діяльності.

PH 50. Дотримуватися етичних норм, враховуючи авторське право та норми академічної доброчесності при проведенні досліджень, розробці програмних продуктів, проектів, презентацій результатів роботи.

Навчально-тематичний план дисципліни

№п п	Назви розділів.	К-ть годин			
		Загальна	Лекції	Лабораторні	Самостійна робота
1.	Розділ 1. Основи об'єктно-орієнтованої парадигми	58	14	24	20
2.	Розділ 2. Основні елементи об'єктно-орієнтованого програмування	56	12	24	20
3.	Розділ 3. Оброблення винятків і бібліотеки класів	56	8	28	20
4.	Розділ 4. Об'єктно-орієнтоване програмування застосувань із графічним інтерфейсом користувача	70	8	42	20
Всього		240	42	118	80

Календарно-тематичний план дисципліни

№	Назва розділу, теми заняття	Кількість годин		
		Лекції	Лабораторні	С. р.
	Розділ 1. Основи об'єктно-орієнтованої парадигми	14	24	20
1.	Парадигми програмування.	2		
2.	Базові поняття програмування.	2		
3.	Базові поняття програмування.	2		
4.	Основи Microsoft.NET та Java SE		2	
5.	Основи об'єктно-орієнтованої мови програмування	2		
6.	Одновимірні та багатовимірні масиви у Java.		2	
7.	Методи у Java.	2		
8.	Поняття об'єктно-орієнтованого аналізу, проектування та програмування	2		
9.	Основи програмування на Java.		2	
10.	Основи програмування на Java		2	
11.	Основи програмування на Java		2	
12.	Основи програмування на Java		2	
13.	Робота з рядками. Об'єкти (змінні) типу String.		2	
14.	Робота з рядками.		2	

№	Назва розділу, теми заняття	Кількість годин		
		Лекції	Лабораторні	С. р.
15.	Поняття масиву рядків.	2		
16.	Робота з рядками		2	
17.	Лямбда-вирази.		2	
18.	Регулярні вирази		2	
19.	Класи Java для роботи з регулярними виразами		2	
	Розділ 2. Основні елементи об'єктно-орієнтованого програмування	12	24	20
20.	Абстрагування даних та інкапсуляція	2		
21.	Життєвий цикл об'єктів.		2	
22.	Концепція об'єктно-орієнтованого програмування.		2	
23.	Повторне використання класів. Поняття про асоціацію.		2	
24.	Повторне використання класів. Відношення успадкування.		2	
25.	Повторне використання класів. Раннє та пізнє зв'язування.		2	
26.	Концепція об'єктно-орієнтованого програмування.	2		
27.	Принципи об'єктно-орієнтованого проектування класів	2		
28.	Бібліотеки класів	2		
29.	Концепція об'єктно-орієнтованого програмування.	2		
30.	Структури даних.		2	
31.	Структури даних.		2	
32.	Колекції. Типи колекцій.	2		
33.	Колекції		2	
34.	Колекції. Сортування.		2	
35.	Додаткові класи.		2	
36.	Колекції		2	
37.	Додаткові класи. Робота з датами.		2	
	Розділ 3. Оброблення винятків і бібліотеки класів	8	28	20
38.	Оброблення виняткових ситуацій	2		
39.	Обробка виключень.		2	
40.	Обробка виключень.		2	
41.	Обробка виключень.		2	
42.	Обробка виключень		2	
43.	Стандартні бібліотеки класів середовищ розробки	2		

№	Назва розділу, теми заняття	Кількість годин		
		Лекції ї	Лабораторні	С. р.
	програм			
44.	Загальні відомості про колекції.	2		
45.	Джерела та споживачі даних.	2		
46.	Регулярні вирази.		2	
47.	Регулярні вирази		2	
48.	Потоки введення-виведення. Робота з файлами.		2	
49.	Потоки введення-виведення. Робота з файлами		2	
50.	Потоки введення-виведення. Робота з файлами		2	
51.	Потоки введення-виведення. Робота з файлами.		2	
52.	Потоки введення-виведення.		2	
53.	Робота з файлами.		2	
54.	Багатопоточне програмування.		2	
55.	Багатопоточне програмування		2	
	Розділ 4. Об'єктно-орієнтоване програмування застосувань із графічним інтерфейсом користувача	8	42	20
56.	Основи розроблення графічних інтерфейсів користувача	2		
57.	Мова FXML.		2	
58.	Розроблення графічних інтерфейсів користувача на платформі Java SE		2	
59.	Створення графічного інтерфейсу користувача на Java	2		
60.	Створення графічного інтерфейсу користувача на Java		2	
61.	Створення графічного інтерфейсу користувача на Java.	2		
62.	Створення графічного інтерфейсу користувача на Java		2	
63.	Оброблення подій від миші, клавіатури й елементів управління.		2	
64.	Створення графічного інтерфейсу користувача на Java.		2	
65.	Аплети (Applets) в Java.	2		
66.	Аплети (Applets) в Java.		2	
67.	Принципи роботи веб-аплікацій.		2	
68.	Розробка web-аплікацій на Java Обробка Http - запитів.		2	

№	Назва розділу, теми заняття	Кількість годин		
		Лекції	Лабораторні	С. р.
69.	Розробка web-аплікацій на Java		2	
70.	Розробка web-аплікацій на Java		2	
71.	War-архіви та запуск веб-аплікацій.		2	
72.	Android-програмування Початок роботи з Android.		2	
73.	Android-програмування. Робота зі зображеннями.		2	
74.	Android-програмування. Робота з мережею. WebView.		2	
75.	Android-програмування. Робота з мультимедіа.		2	
76.	Android-програмування. Сервіси.		2	
77.	Android-програмування. Публікація додатків.		2	
78.	Модульність. Створення модуля.		2	
79.	Android-програмування		2	
80.	Модульність		2	
	Всього по предмету	42	118	80

Теми і плани лекційних занять

Розділ 1. Основи об'єктно-орієнтованої парадигми

Тема 1. Парадигми програмування.

План:

1. Парадигми і мови програмування.
2. Процедурне програмування.
3. Об'єктне (модульне) програмування.
4. Об'єктно-орієнтовне програмування.
5. Програмні середовища

Тема 2. Базові поняття програмування.

План:

1. Фундаментальні типи даних і операції над ними. Розрядність фундаментальних типів даних. Операції над даними фундаментальних типів.
2. Літерали та їхні типи. Числові літерали. Символьні літерали.
3. Сталі та змінні

Тема 3. Базові поняття програмування.

План:

1. Указники. Указник як засіб непрямого адресування. Указники та динамічна пам'ять. Указники констант. Типізація указників. Операції над указниками.
2. Відсилки.
3. Інструкції. Прості інструкції. Структури керування.
4. Налагодження програм

Тема 4. Основи об'єктно-орієнтованої мови програмування

План:

1. Загальні відомості про мову Java.
2. Алфавіт, типи даних, операції, оператори, структура програми, основи використання стандартних бібліотек класів Microsoft .NET і Java.
3. Віртуальна Java-машина.
4. Байт-код.
5. JIT- компіляція. Категорії програм, написаних на мові Java. Середовища розробки Java-проектів.

Тема 5. Методи у Java.

План:

1. Визначення, механізми передавання параметрів, використання масиву як параметра, повертання масиву з методу, виклик методу

Тема 6. Поняття об'єктно-орієнтованого аналізу, проектування та програмування

План:

1. Об'єктно-орієнтована декомпозиція. Принципи об'єктно-орієнтованого підходу: абстракція, інкапсуляція, ієрархія, поліморфізм
2. Поняття об'єкта. Характеристики об'єкта. Поняття класу. Співвідношення між класом та його об'єктом
3. Об'єктно-орієнтований аналіз та його мета. Головні види вимог до програмної системи. Об'єктно-орієнтоване проектування. Об'єктно-орієнтоване програмування
4. UML-діаграми класів. Відношення на діаграмі класів. CASE-засоби

Тема 7. Поняття масиву рядків.

План:

1. Загальна форма оголошення одновимірного масиву рядків.
2. Двовимірний масив рядків.
3. Довжина масиву рядків. Властивість `length`.
4. Пошук заданого рядка в одновимірному масиві рядків.
5. Сортування одновимірного масиву рядків за алфавітом методом вставки.
6. Визначення кількості входжень заданого рядка у двовимірному масиві рядків.
7. Заміна рядка у двовимірному масиві рядків.
8. Класи `StringBuffer` і `StringBuilder`.

Розділ 2. Основні елементи об'єктно-орієнтованого програмування

Тема 8. Абстрагування даних та інкапсуляція

План:

1. Абстрактні типи даних. Проектування абстрактного типу даних
2. Класи та структури. Елементи класу. Особливості використання статичних елементів. Доступ до елементів класу, модифікатори доступу
3. Поняття про створення, ініціалізацію та використання об'єктів класу.

Посилання `this`

Тема 9. Концепція об'єктно-орієнтованого програмування.

План:

1. Перерахування `enum`.
2. Клас `Object` і його методи.
3. Узагальнення (`Generics`).
4. Обмеження узагальнень.
5. Спадкування і узагальнення.
6. Довідкові типи і копіювання об'єктів.

Тема 10. Принципи об'єктно-орієнтованого проектування класів

План:

1. Система принципів SOLID. Принцип єдиної відповідальності
2. Загальні відомості про шаблони проектування. Застосування основних шаблонів проектування

Тема 11. Бібліотеки класів

План:

1. Бібліотеки та їх використання. Статичні та динамічні бібліотеки.
2. Розроблення бібліотек на платформі Java SE. DLL-бібліотеки. Розроблення DLL-бібліотек на платформі Microsoft .NET

Тема 12. Концепція об'єктно-орієнтованого програмування.

План:

1. Класи і об'єкти.
2. Пакети.
3. Модифікатори доступу і інкапсуляція.
4. Статичні члени і модифікатор static.
5. Об'єкти як параметри методів.
6. Внутрішні і вкладені класи

Тема 13. Колекції. Типи колекцій.

План:

1. Інтерфейс Collection.
2. Клас ArrayList і інтерфейс List.
3. Черги і клас ArrayDeque.
4. Клас LinkedList.
5. Інтерфейс Set і клас HashSet, SortedSet, NavigableSet, TreeSet.

Розділ 3. Оброблення винятків і бібліотеки класів

Тема 14. Оброблення виняткових ситуацій

План:

1. Види помилок у програмах.
2. Проблеми традиційного підходу до оброблення помилок
3. Механізм оброблення винятків. Класи винятків стандартних бібліотек Microsoft.NET і Java SE. Синтаксис оброблення винятків

Тема 15. Стандартні бібліотеки класів середовищ розробки програм

План:

1. Призначення та застосування регулярних виразів.
2. Підтримка регулярних виразів на платформах Microsoft .NET і Java SE.
3. Спеціальні символи, використовувані у регулярних виразах

Тема 16. Загальні відомості про колекції.

План:

1. Основні структури даних стандартних бібліотек колекцій Microsoft .NET і Java SE.
2. Типізовані колекції

Тема 17. Джерела та споживачі даних.

План:

1. Загальні відомості про потоки введення – виведення даних.
2. Алгоритми роботи потоків введення – виведення даних.
3. Основні класи стандартних бібліотек Microsoft.NET і Java SE для підтримки введення – виведення даних.
4. Серіалізація.

Розділ 4. Об'єктно-орієнтоване програмування застосувань із графічним інтерфейсом користувача

Тема 18. Основи розроблення графічних інтерфейсів користувача

План:

1. Загальні відомості про події. Генерування подій. Обробники подій
2. Огляд сучасних технологій розроблення застосунків із графічним інтерфейсом користувача на платформах Microsoft .NET і Java SE. Основи мови XML. Структура XML-документа. Простори імен XML. XML-схеми.

Тема 19. Створення графічного інтерфейсу користувача на Java

План:

1. Графічний інтерфейс користувача: визначення, стилі, принципи побудови, моделі, методи, засоби та інструменти розроблення.
2. Бібліотеки для створення графічного інтерфейсу.

Тема 20. Створення графічного інтерфейсу користувача на Java.

План:

1. Бібліотека Swing.
2. Архітектура MVC.
3. Компоненти і контейнери.
4. Менеджери розмітки.
5. Графічні компоненти Swing.

6. Події. Механізм обробки подій Swing

Тема 21. Аплети (Applets) в Java.

План:

1. Поняття аплета.
2. Кросплатформність.
3. Життєвий цикл.

Теми і плани лабораторних занять

№	Назва розділу, теми заняття	Кіль-т ь годин
	Розділ 1. Основи об'єктно-орієнтованої парадигми	24
1.	Основи Microsoft.NET та Java SE 1. Програмна платформа Microsoft.NET 2. Програмна платформа Java SE	2
2.	Одновимірні та багатовимірні масиви у Java. 1. Створення, ініціалізація, оброблення, підтримка масивів у стандартних бібліотеках Microsoft .NET і Java SE	2
3.	Основи програмування на Java. 1. Структура програми. 2. Змінні і константи. 3. Типи даних. Консольне введення / виведення в Java. 4. Арифметичні операції. 5. Побітові операції. 6. Умовні вирази. 7. Операції присвоювання і пріоритет операцій. 8. Перетворення базових типів даних.	2
4.	Основи програмування на Java 1. Умовні конструкції. 2. Цикли. 3. Масиви.	2
5.	Основи програмування на Java 1. Методи. 2. Параметри методів. 3. Оператор return.	2
6.	Основи програмування на Java 1. Перевантаження методів. 2. Рекурсивні функції. 3. Обробка виключних ситуацій.	2

№	Назва розділу, теми заняття	Кіль-т ь годин
7.	<p>Робота з рядками. Об'єкти (змінні) типу String.</p> <ol style="list-style-type: none"> 1. Перевірка двох рядків на рівність. Метод equals(). Довжина рядка. Метод length(). Визначення символу за заданим індексом. Метод charAt(). 2. Методи codePointAt() та codePointBefore(). Методи compareTo() та compareToIgnoreCase(). Об'єднання рядків. 3. Визначення наявності підрядка в рядку. Метод contains(). 4. Метод endsWith(). Розбиття рядка на масив цілих чисел. Метод getBytes(). Визначення першого та останнього входження символу або рядка. Методи indexOf() та lastIndexOf(). Перевірка, чи рядок є пустий. 5. Метод isEmpty(). Заміна символів в тексті. Метод replace(). Метод startsWith(). Метод substring(). 	2
8.	<p>Робота з рядками.</p> <ol style="list-style-type: none"> 1. Розбиття рядка на символи. 2. Метод toCharArray(). 3. Приведення до потрібного регістру символів. 4. Методи toLowerCase() та toUpperCase(). 5. Перетворення масиву символів char в рядок String. 6. Метод copyValueOf(). Перетворення числа в рядок. Метод valueOf(). 	2
9.	<p>Робота з рядками</p> <ol style="list-style-type: none"> 1. Пошук заданого рядка в одновимірному масиві рядків. 2. Сортування одновимірному масиву рядків за алфавітом методом вставки. 3. Визначення кількості входжень заданого рядка у двовимірному масиві рядків. 4. Заміна рядка у двовимірному масиві рядків. 	2
10.	<p>Лямбда-вирази.</p> <ol style="list-style-type: none"> 1. Поняття лямбда-виразів. 2. Лямбда як параметри і результати методів. 3. Вбудовані функціональні інтерфейси 	2
11.	<p>Регулярні вирази</p> <ol style="list-style-type: none"> 1. Поняття регулярного виразу. 2. Синтаксис регулярного виразу. 3. Операція Альтернації. 4. Одиночний метасимвол. 5. Квантіфікатори. 6. Класи символів. 	2

№	Назва розділу, теми заняття	Кіль-т ь годин
	7. Спеціальні символи. 8. Анкери.	
12.	Класи Java для роботи з регулярними виразами 1. Клас Pattern. 2. Клас Matcher. 3. Клас PatternSyntaxException. 4. Тестування регулярних виразів.	2
	Розділ 2. Основні елементи об'єктно-орієнтованого програмування	24
13.	Життєвий цикл об'єктів. 1. Послідовність створення та ініціалізації об'єкта. 2. Конструктори. 3. Конструктор за замовчуванням. 4. Основні властивості конструкторів. Перевантаження конструкторів. 5. Звільнення пам'яті. Система «збирання сміття»	2
14.	Концепція об'єктно-орієнтованого програмування. 1. Спадкування. Абстрактні класи. 2. Ієрархія. 3. Наслідування і перетворення типів. 4. Інтерфейси. Інтерфейси в механізмі зворотного виклику	2
15.	Повторне використання класів. Поняття про асоціацію. 1. Відношення композиції та агрегації як види асоціації. 2. Реалізація композиції та агрегації	2
16.	Повторне використання класів. Відношення успадкування. 1. Реалізація відношення успадкування. 2. Ініціалізація об'єкта базового класу. 3. Використання конструкторів під час успадкування. 4. Варіанти використання успадкування. 5. Перевизначення методів	2
17.	Повторне використання класів. Раннє та пізнє зв'язування. 1. Віртуальні методи. 2. Реалізація принципу поліморфізму. 3. Рядкове подання об'єкта. 4. Абстрактні класи та методи. 5. Реалізація поліморфної поведінки на базі абстрактного класу. 6. Інтерфейси. 7. Реалізація поліморфної поведінки на базі інтерфейсу	2

№	Назва розділу, теми заняття	Кіль-т ь годин
18.	Структури даних. 1. Інтерфейс Enumeration. 2. Клас BitSet. 3. Клас Vector. 4. Клас Stack.	2
19.	Структури даних. 1. Клас Dictionary. 2. Клас Hashtable (хеш-таблиця). 3. Клас Properties	2
20.	Колекції 1. Інтерфейси Comparable і Comparator.	2
21.	Колекції. Сортування. 1. Інтерфейс Map і клас HashMap. 2. Інтерфейси SortedMap і NavigableMap. 3. Клас TreeMap. Ітератори	2
22.	Додаткові класи. 1. Математичні обчислення і клас Math. 2. Великі числа BigInteger і BigDecimal.	2
23.	Колекції	2
24.	Додаткові класи. Робота з датами. 1. LocalDate	2
	Розділ 3. Оброблення винятків і бібліотеки класів	28
25.	Обробка виключень. 1. Виключення (exceptions). 2. Виключна ситуація. 3. Ієрархія виключень.	2
26.	Обробка виключень. 1. Ключові слова try, catch, finally. 2. Оператори throw, throws. 3. Основні типи виключень в Java.	2
27.	Обробка виключень. 1. Класи Java для обробки виключних ситуацій з пакету java.lang. 2. Методи класу Throwable. 3. Клас Exception. 4. Створення власних класів виключень.	2
28.	Обробка виключень	2
29.	Регулярні вирази.	2

№	Назва розділу, теми заняття	Кіль-ть ь годин
	1. Можливості застосування Regular Expressions. 2. Звичайні символи (літерали) і спеціальні символи (метасимволи). 3. Символьні класи (набори символів).	
30.	Регулярні вирази 1. Групи. 2. Переліки. 3. Квантифікація (пошук послідовностей). 4. Групування.	2
31.	Потоки введення-виведення. Робота з файлами. 1. Потоки введення-виведення. 2. Читання і запис файлів. 3. FileInputStream і FileOutputStream. 4. Закриття потоків. 5. Класи ByteArrayInputStream і ByteArrayOutputStream. 6. Буферизовані потоки BufferedInputStream і BufferedOutputStream.	2
32.	Потоки введення-виведення. Робота з файлами 1. Потоки введення-виведення. 2. Читання і запис файлів. 3. FileInputStream і FileOutputStream. 4. Закриття потоків.	2
33.	Потоки введення-виведення. Робота з файлами 1. Форматування введення і виведення. PrintStream і PrintWriter.	2
34.	Потоки введення-виведення. Робота з файлами. 1. Класи DataOutputStream і DataInputStream. 2. Читання і запис текстових файлів. 3. Буферизація символьних потоків. 4. BufferedReader і BufferedWriter. Серіалізація об'єктів. 5. Клас File. 6. Робота з файлами і каталогами. 7. Робота з ZIP-архівами. Клас Console.	2
35.	Потоки введення-виведення. 1. Класи DataOutputStream і DataInputStream. 2. Читання і запис текстових файлів. 3. Буферизація символьних потоків. 4. BufferedReader і BufferedWriter. Серіалізація об'єктів.	2
36.	Робота з файлами.	2

№	Назва розділу, теми заняття	Кіль-т ь годин
	1. Клас File. 2. Робота з файлами і каталогами. 3. Робота з ZIP-архівами. Клас Console.	
37.	Багатопоточне програмування. 1. Клас Thread. 2. Створення та виконання потоків. Завершення і переривання потоку. Синхронізація потоків. 3. Оператор synchronized. 4. Методи wait і notify. 5. Семафори. 6. Обмін між потоками. 7. Клас Exchanger. 8. Клас Phaser. 9. Блокування. ReentrantLock. Умови в блокування	2
38.	Багатопоточне програмування	2
	Розділ 4. Об'єктно-орієнтоване програмування застосувань із графічним інтерфейсом користувача	42
39.	Мова FXML. 1. Загальні відомості про мову FXML. 2. Використання FXML для опису графічного інтерфейсу користувача	2
40.	Розроблення графічних інтерфейсів користувача на платформі Java SE 1. Загальна структура застосунку JavaFX. 2. Форми. 3. Події рівня форми. 4. Використання форм і базових елементів управління JavaFX	2
41.	Створення графічного інтерфейсу користувача на Java 1. Графічний інтерфейс користувача: визначення, стилі, принципи побудови, моделі, методи, засоби та інструменти розроблення. 2. Бібліотеки для створення графічного інтерфейсу.	2
42.	Створення графічного інтерфейсу користувача на Java 1. Бібліотека Swing. 2. Архітектура MVC. 3. Компоненти і контейнери. 4. Менеджери розмітки.	2

№	Назва розділу, теми заняття	Кіль-т ь годин
	5. Графічні компоненти Swing. 6. Події. Механізм обробки подій Swing	
43.	Оброблення подій від миші, клавіатури й елементів управління. 1. Використання основних елементів управління JavaFX.	2
44.	Створення графічного інтерфейсу користувача на Java. 1. Основні концепції платформи JavaFX. 2. Компонування в JavaFX. 3. Використання елементів управління (controls). 4. Обробка подій у JavaFX. Дочірні та діалогові вікна в JavaFX	2
45.	Аплети (Applets) в Java. 1. Виконання аплета. 2. Клас JApplet.	2
46.	Принципи роботи веб-аплікацій. 1. Фреймворки для розробки веб-аплікацій. 2. Приклади розробки MVC архітектури. 3. Технологія Servlets API. 4. Клас HttpServlet. 5. API-інтерфейси пакету javax.servlet.http.	2
47.	Розробка web-аплікацій на Java Обробка Http - запитів. 1. Контейнер сервлетів Tomcat. 2. Способи побудови веб-сторінок. 3. Розгортання веб-аплікацій	2
48.	Розробка web-аплікацій на Java	2
49.	Розробка web-аплікацій на Java	2
50.	War-архіви та запуск веб-аплікацій. 1. Конфігурування за допомогою web.xml та анотацій. 2. Використання Maven для побудови аплікації.	2
51.	Android-програмування Початок роботи з Android. 1. Основи створення інтерфейсу. 2. Основні елементи управління. 3. Ресурси. Activity.	2
52.	Android-програмування. Робота зі зображеннями. 1. Адаптери і списки. 2. Стили і теми. 3. Меню. 4. Фрагменти.	2
53.	Android-програмування. Робота з мережею. WebView. 1. Багатопоточність і асинхронність.	2

№	Назва розділу, теми заняття	Кіль-т ь годин
	2. Робота з мережею. WebView.	
54.	Android-програмування. Робота з мультимедіа. 1. Налаштування і стан додатки. 2. Робота з файловою системою. 3. Робота з базами даних SQLite.	2
55.	Android-програмування. Сервіси. 1. Перегортання сторінок і ViewPager. 2. Сервіси. Телефонія і комунікація. 3. Діалогові вікна. 4. Анімація.	2
56.	Android-програмування. Публікація додатків. 1. Провайдери контенту. 2. Публікація додатків. 3. Робота з XML. 4. Робота з Github	2
57.	Модульність. Створення модуля. 1. Створення модуля. 2. Залежні модулі. 3. Взаємодія між модулями.	2
58.	Android-програмування	2
59.	Модульність	2
	Всього по предмету	118

Теми для самостійної роботи студентів

Форма контролю – застосування набутих знань під час виконання лабораторних робіт.

№	Назва теми	Кількість годин
	Розділ 1. Основи об'єктно-орієнтованої парадигми	20
1	Основи Microsoft .NET та Java SE 1. Програмна платформа Microsoft .NET 2. Програмна платформа Java SE	4
2	Основи об'єктно-орієнтованої мови програмування 1. Загальні відомості про мови C# і Java: алфавіт, типи даних, операції, оператори, структура програми, основи використання стандартних бібліотек класів Microsoft .NET і Java SE2.2.	7

№	Назва теми	Кількість годин
	<p>Одновимірні та багатовимірні масиви у C# і Java: створення, ініціалізація, оброблення, підтримка масивів у стандартних бібліотеках Microsoft .NET і Java SE</p> <p>2. Методи у C# і Java: визначення, механізми передавання параметрів, використання масиву як параметра, повертання масиву з методу, виклик методу</p>	
3	<p>Поняття об'єктно-орієнтованого аналізу, проектування та програмування</p> <p>1. Об'єктно-орієнтована декомпозиція. Принципи об'єктно-орієнтованого підходу: абстракція, інкапсуляція, ієрархія, поліморфізм</p> <p>2. Поняття об'єкта. Характеристики об'єкта. Поняття класу. Співвідношення між класом та його об'єктом</p> <p>3. Об'єктно-орієнтований аналіз та його мета. Головні види вимог до програмної системи. Об'єктно-орієнтоване проектування. Об'єктно-орієнтоване програмування</p> <p>4. UML-діаграми класів. Відношення на діаграмі класів. CASE-засоби</p>	9
	Розділ 2. Основні елементи об'єктно-орієнтованого програмування	20
4	<p>Абстрагування даних та інкапсуляція</p> <p>1. Абстрактні типи даних. Проектування абстрактного типу даних</p> <p>2. Класи та структури. Елементи класу. Особливості використання статичних елементів. Доступ до елементів класу, модифікатори доступу</p> <p>3. Поняття про створення, ініціалізацію та використання об'єктів класу. Посилання this</p> <p>4. Життєвий цикл об'єктів. Послідовність створення та ініціалізації об'єкта. Конструктори. Конструктор за замовчуванням. Основні властивості конструкторів. Перевантаження конструкторів. Звільнення пам'яті. Система «збирання сміття»</p>	5
5	<p>Повторне використання класів</p> <p>1. Поняття про асоціацію. Відношення композиції та агрегації як види асоціації. Реалізація композиції та агрегації в C# і Java</p> <p>2. Відношення успадкування. Реалізація відношення успадкування в C# і Java. Ініціалізація об'єкта базового класу.</p>	5

№	Назва теми	Кількість годин
	<p>Використання конструкторів під час успадкування. Варіанти використання успадкування. Перевизначення методів</p> <p>3. Раннє та пізнє зв'язування. Віртуальні методи. Реалізація принципу поліморфізму в C# і Java. Рядкове подання об'єкта. Абстрактні класи та методи. Реалізація поліморфної поведінки на базі абстрактного класу. Інтерфейси. Реалізація поліморфної поведінки на базі інтерфейсу</p>	
6	<p>Принципи об'єктно-орієнтованого проектування класів</p> <p>1. Система принципів SOLID. Принцип єдиної відповідальності</p> <p>2. Загальні відомості про шаблони проектування. Застосування основних шаблонів проектування</p>	5
7	<p>Бібліотеки класів</p> <p>1. Бібліотеки та їх використання. Статичні та динамічні бібліотеки</p> <p>2. Розроблення бібліотек на платформі Java SE. DLL-бібліотеки. Розроблення DLL-бібліотек на платформі Microsoft .NET</p>	5
	Розділ 3. Оброблення винятків і бібліотеки класів	20
8	<p>Оброблення виняткових ситуацій</p> <p>1. Види помилок у програмах. Проблеми традиційного підходу до оброблення помилок</p> <p>2. Механізм оброблення винятків. Класи винятків стандартних бібліотек Microsoft .NET і Java SE. Синтаксис оброблення винятків</p>	10
9	<p>Стандартні бібліотеки класів середовищ розробки програм</p> <p>1. Призначення та застосування регулярних виразів. Підтримка регулярних виразів на платформах Microsoft .NET і Java SE. Спеціальні символи, використовувані у регулярних виразах</p> <p>2. Загальні відомості про колекції. Основні структури даних стандартних бібліотек колекцій Microsoft .NET і Java SE. Типізовані колекції</p> <p>3. Джерела та споживачі даних. Загальні відомості про потоки введення – виведення даних. Алгоритми роботи потоків введення – виведення даних. Основні класи стандартних бібліотек Microsoft .NET і Java SE для підтримки введення – виведення даних. Серіалізація</p>	10

№	Назва теми	Кількість годин
	Розділ 4. Об'єктно-орієнтоване програмування застосувань із графічним інтерфейсом користувача	20
10	Основи розроблення графічних інтерфейсів користувача 1. Загальні відомості про події. Генерування подій. Обробники подій 2. Огляд сучасних технологій розроблення застосунків із графічним інтерфейсом користувача на платформах Microsoft .NET і Java SE. Основи мови XML. Структура XML-документа. Простори імен XML. XML-схеми 3. Загальні відомості про мову FXML. Використання FXML для опису графічного інтерфейсу користувача	10
11	Розроблення графічних інтерфейсів користувача на платформі Java SE 1. Загальна структура застосунку JavaFX. Форми. Події рівня форми. Використання форм і базових елементів управління JavaFX 2. Оброблення подій від миші, клавіатури й елементів управління. Використання основних елементів управління JavaFX.	10
	Всього годин	80

Методи активізації навчального процесу

При викладанні дисципліни використовуються наступні методи навчання:

Лекція (проблемна, інтерактивна). На лекціях застосовуються мультимедійні засоби та дискусії.

Лабораторна робота – для використання набутих знань до розв'язування практичних завдань. Лабораторні роботи проводяться у двох частинах – пояснення завдання та захист студентами їхнього виконання.

Проблемне навчання – створення проблемної ситуації для зацікавленого і активного сприйняття матеріалу.

Проектне навчання (індивідуальне, малі групи, групове).

Он-лайн навчання. Для розповсюдження усіх матеріалів як з боку викладача, так і з боку студента, використовуються додатки платформи Google Workspace for Education.

Система поточного і підсумкового контролю знань

Система поточного контролю знань:

- виконання лабораторних робіт студентами.

Система підсумкового контролю знань:

- екзамен.

Питання до екзамену

1. Що таке Java. Віртуальна Java-машина.
2. Байт-код. JIT-компіляція.
3. Категорії програм, написаних на мові Java.
4. Середовища розробки Java-проектів.
5. Структура Java-програми.
6. Змінні і константи. Типи даних.
7. Консольне введення / виведення в Java.
8. Арифметичні операції.
9. Побітові операції.
10. Умовні вирази.
11. Операції присвоювання і пріоритет операцій.
12. Перетворення базових типів даних.
13. Умовні конструкції.
14. Цикли.
15. Масиви.
16. Методи. Параметри методів. Оператор return.
17. Перевантаження методів.
18. Рекурсивні функції.
19. Об'єкти (змінні) типу String.
20. Методи для роботи з рядками.
21. Поняття лямбда-виразів.
22. Лямбда як параметри і результати методів.
23. Вбудовані функціональні інтерфейси.
24. Принципи ООП.
25. Класи і об'єкти.
26. Модифікатори доступу і інкапсуляція.
27. Статичні члени і модифікатор static.
28. Об'єкти як параметри методів.
29. Внутрішні і вкладені класи.
30. Спадкування.

31. Абстрактні класи.
32. Ієрархія. Наслідування і перетворення типів.
33. Інтерфейси. Інтерфейси в механізмі зворотного виклику.
34. Перерахування enum.
35. Клас Object і його методи.
36. Узагальнення (Generics). Обмеження узагальнень.
37. Спадкування і узагальнення.
38. Довідкові типи і копіювання об'єктів.
39. Інтерфейс Enumeration.
40. Клас BitSet.
41. Клас Vector.
42. Клас Stack.
43. Клас Dictionary.
44. Клас Hashtable (хеш-таблиця).
45. Клас Properties.
46. Типи колекцій.
47. Інтерфейс Collection.
48. Клас ArrayList і інтерфейс List.
49. Черги і клас ArrayDeque.
50. Клас LinkedList.
51. Інтерфейс Set і клас HashSet.
52. SortedSet, NavigableSet, TreeSet.
53. Інтерфейси Comparable і Comparator.
54. Інтерфейс Map і клас HashMap.
55. Інтерфейси SortedMap і NavigableMap. клас TreeMap.
56. Ітератори.
57. Математичні обчислення і клас Math.
58. Великі числа BigInteger і BigDecimal.
59. Робота з датами. LocalDate.
60. Виключення (exceptions). Виключна ситуація.
61. Ієрархія виключень.
62. Ключові слова try, catch, finally.
63. Оператори throw, throws.
64. Основні типи виключень в Java.
65. Класи Java для обробки виключних ситуацій з пакету java.lang.
66. Методи класу Throwable.
67. Клас Exception.
68. Створення власних класів виключень.
69. Можливості застосування Regular Expressions.
70. Звичайні символи (літерали) і спеціальні символи (метасимволи).

71. Символьні класи (набори символів).
72. Групи. Переліки. Квантифікація (пошук послідовностей).
73. Групування.
74. Потоки введення-виведення.
75. Читання і запис файлів. `FileInputStream` і `FileOutputStream`.
76. Закриття потоків.
77. Класи `ByteArrayInputStream` і `ByteArrayOutputStream`.
78. Буферизовані потоки `BufferedReader` і `BufferedWriter`.
79. Форматування введення і виведення. `PrintStream` і `PrintWriter`.
80. Класи `DataOutputStream` і `DataInputStream`.
81. Читання і запис текстових файлів.
82. Буферизація символьних потоків. `BufferedReader` і `BufferedWriter`.
83. Сериалізація об'єктів.
84. Клас `File`. Робота з файлами і каталогами.
85. Робота з ZIP-архівами.
86. Клас `Console`.
87. Клас `Thread`.
88. Створення та виконання потоків.
89. Завершення і переривання потоку.
90. Синхронізація потоків.
91. Оператор `synchronized`.
92. Методи `wait` і `notify`.
93. Семафори.
94. Обмін між потоками.
95. Клас `Exchanger`.
96. Клас `Phaser`.
97. Блокування. `ReentrantLock`. Умови в блокуванні.
98. Графічний інтерфейс користувача: визначення, стилі, принципи побудови, моделі, методи, засоби та інструменти розроблення.
99. Бібліотеки для створення графічного інтерфейсу.
100. Бібліотека `Swing`.
101. Архітектура `MVC`.
102. Компоненти і контейнери.
103. Менеджери розмітки.
104. Графічні компоненти `Swing`.
105. Події. Механізм обробки подій `Swing`.
106. Основні концепції платформи `JavaFX`.
107. Компонування в `JavaFX`.
108. Використання елементів управління (controls).
109. Обробка подій у `JavaFX`.

110. Дочірні та діалогові вікна в JavaFX.
111. Поняття аплету.
112. Кросплатформність.
113. Життєвий цикл.
114. Виконання аплету.
115. Клас JApplet.
116. Принципи роботи веб-аплікацій.
117. Фреймворки для розробки веб-аплікацій.
118. Приклади розробки MVC архітектури.
119. Технологія Servlets API.
120. Клас HttpServlet.
121. API-інтерфейси пакету javax.servlet.http.
122. Обробка Http - запитів.
123. Контейнер сервлетів Tomcat.
124. Способи побудови веб-сторінок.
125. Розгортання веб-аплікацій.
126. War-архіви та запуск веб-аплікацій.
127. Конфігурування за допомогою web.xml та анотацій.
128. Використання Maven для побудови аплікації.
129. Початок роботи з Android.
130. Основи створення інтерфейсу.
131. Основні елементи управління. Ресурси.
132. Activity.
133. Робота зі зображеннями.
134. Адаптери і списки.
135. Стили і теми. Меню. Фрагменти.
136. Багатопоточність і асинхронність.
137. Робота з мережею. WebView.
138. Робота з мультимедіа. Налаштування і стан додатка.
139. Робота з файловою системою.
140. Робота з базами даних SQLite.
141. Перегортання сторінок і ViewPager.
142. Сервіси. Телефонія і комунікація.
143. Діалогові вікна.
144. Анімація.
145. Провайдери контенту.
146. Публікація додатків.
147. Робота з XML.
148. Робота з Github.

Критерії оцінювання результатів навчання студентів

Оцінювання результатів навчальної діяльності студентів здійснюється за середнім балом виведеному по кожному запитанню.

При оцінці «5» студент:

у повному обсязі володіє навчальним матеріалом, вільно, самостійно та аргументовано його викладає, глибоко та всебічно розкриває зміст теоретичних запитань та практичних завдань. Використовуючи при цьому обов'язкову та додаткову літературу, вільно послуговується науковою термінологією, розв'язує завдання стандартним або оригінальним способом, наводить аргументи на підтвердження власних думок, здійснює аналіз та робить висновки

При оцінці «4» студент:

достатньо повно володіє навчальним матеріалом, обґрунтовано його викладає, в основному розкриває зміст теоретичних запитань та практичних завдань. Але при висвітленні деяких питань не вистачає достатньої глибини та аргументації, допускаються при цьому окремі неістотні неточності та незначні помилки.

При оцінці «3» студент:

відтворює значну частину навчального матеріалу, висвітлює його основний зміст, виявляє елементарні знання окремих положень. Виконує практичні завдання. Не здатний до глибокого, всебічного аналізу, обґрунтування та аргументації, не користується необхідною літературою, допускає істотні неточності та помилки.

При оцінці «2» студент:

не володіє навчальним матеріалом у достатньому обсязі, проте фрагментарно, поверхово (без аргументації та обґрунтування) викладає окремі питання навчальної дисципліни, не розкриває зміст теоретичних питань і практичних завдань.

Рекомендована література

Основна

1. Міловідов Ю.О. «Об'єктно-орієнтоване програмування» Навчальний посібник. – Видавничий центр НУБіП України, 2019. – 301 с.
2. Об'єктно орієнтоване програмування на Java. Конспект лекцій з дисципліни «Об'єктно орієнтоване програмування» для студентів спеціальностей 121 – «Програмна інженерія», 123 – «Комп'ютерна інженерія». /Укл.: Бивойно П.Г. – Чернігів: ЧНТУ, 2019. – 136 с. Режим доступу до ресурсу: http://www.dgma.donetsk.ua/docs/kafedry/avp/metod/%D0%9E%D0%9E%D0%9F%20%D0%9A%D0%BE%D0%BD%D1%81%D0%BF%D0%B5%D0%BA%D1%82%20%D0%BB%D0%B5%D0%BA%D1%86_%D0%B9%20%D0%91%D0%B8%D0%B2%D0%BE%D0%B9%D0%BD%D0%BE.pdf
3. Щербаков О. В. Основи об'єктно-орієнтованого програмування [Електронний ресурс] : навчальний посібник / О. В. Щербаков, Ю. Е. Парфьонов, В. М. Федорченко. – Харків : ХНЕУ ім. С. Кузнеця, 2019. – 237 с.

Додаткова

4. Алгоритми і структура даних: Навчальний посібник / В.М.Ткачук. – Івано-Франківськ : Видавництво Прикарпатського національного університету імені Василя Стефаника, 2016. – 286 с.
5. Алгоритми та структури даних. Навчальний посібник / Т. О. Коротєєва. Львів : Видавництво Львівської політехніки, 2014. – 280 с.
6. Глоба Л. С. Розробка інформаційних ресурсів та систем [Електронний ресурс] : конспект лекцій / Л. С. Глоба, Т. М. Кот. – Київ : НТУУ «КПІ», 2014. – 318 с.
7. Грязнова В. О., Єфіменко С. В. Основи методології програмування. – К.: ВПЦ «Київський університет», 2010.
8. Інженерія якості програмного забезпечення: навч. посібник / Г.В Табунщик, Р.К. Кудерметов, Т.І. Брагіна. – Запоріжжя: ЗНТУ, 2013. – 180 с.
9. Кравець П.О. Об'єктно-орієнтоване програмування. Львів: Видавництво Львівської політехніки, 2012. – 624 с.

Інформаційні ресурси

10. Microsoft. Overview of Windows Programming in C++ URL:
<https://docs.microsoft.com/en-us/cpp/windows>
11. Working Draft, Standard for Programming. Language C++. ISO/IEC
N4582, 1514 pp. URL:
<https://www.openstd.org/jtcl/sc22/wg21/docs/papers/2016/n4582.pdf>
12. Електронний навчальний курс «Об'єктно-орієнтоване
програмування» – Режим доступу:
<https://elearn.nubip.edu.ua/course/view.php?id=876>

ДОДАТКИ

Приклад різнорівневого завдання

Початковий рівень – по 0,5 балів.

№1. Елементи об'єктно-орієнтованої програми

Питання: кожна об'єктно-орієнтована програма, по суті, складається з набору...

Виберіть один варіант зі списку:

1. взаємодіючі об'єкти ✓
2. взаємодіючі методи
3. взаємодіючі класи
4. взаємодіючі змінні

№2. Основні принципи

Питання: Об'єктно-орієнтоване програмування базується на 4 принципах. Знайдіть той, який пояснюється неправильно:

Виберіть один варіант зі списку:

1. Абстракція передбачає надання користувачеві лише найбільш релевантних характеристик об'єкта.
2. Поліморфізм — це визначення різної логіки одного методу.
3. Інкапсуляція — це приховування внутрішніх даних об'єктів від світу.
4. Успадкування полягає в іменуванні змінних одним традиційним способом. ✓

№3. Стан і поведінка

Питання: Що можна охарактеризувати станом і поведінкою?

Виберіть один варіант зі списку:

1. метод
2. клас
3. об'єкт ✓
4. поле (або атрибут)

Пояснення: Об'єкт можна охарактеризувати як стан та поведінку в програмуванні. Стан об'єкта відображається в значеннях його полів або атрибутів, які можуть змінюватись під час виконання програми. Поведінка об'єкта визначається методами, які виконуються над цим об'єктом і можуть змінювати його стан.

№4. Взаємодіючі об'єкти

Питання: Як правило, об'єкти взаємодіють один з одним через...

Виберіть один варіант зі списку:

1. спадкування
2. класи
3. методи ✓
4. поля

Пояснення: Об'єкти взаємодіють один з одним через методи. Методи дозволяють об'єктам виконувати певні дії та взаємодіяти з іншими об'єктами шляхом передачі параметрів. Таким чином, методи дозволяють здійснювати комунікацію між об'єктами та виконувати певні операції.

Середній рівень – по 0,5 балів.

№5. Атрибути об'єктів

Питання: Які параметри, наведені нижче, можуть бути атрибутом об'єкта класу Car?

Виберіть один або кілька варіантів зі списку:

1. їздити
2. ціна ✓
3. дорожні знаки
4. подивитися на карту
5. пасажиромісткість ✓
6. марка автомобіля ✓

Пояснення: Атрибутом об'єкта класу Car можуть бути такі варіанти: ціна, пасажиромісткість, марка автомобіля. Інші варіанти не є атрибутами об'єкта класу Car. «Їздити» та «дивитися на карту» — це дії чи поведінка, які може виконувати об'єкт «Автомобіль», тоді як «дорожні знаки» не є специфічним для конкретного об'єкта «Автомобіль» і більше пов'язаний із середовищем, у якому працює автомобіль.

№6. Що таке проєкт (план)

Питання: Клас — це шаблон (або проєкт) для створення ()...

Виберіть один варіант зі списку

1. метод
2. тип
3. поле
4. об'єкт ✓

Пояснення: План належить до класу, який є шаблоном для створення об'єкта.

Достатній рівень – 1 бал

№7. Установіть відповідність між основними поняттями

Запитання: Установіть відповідність між поняттями та відповідними визначеннями.

Клас – тип об'єкта.

Поле – стан об'єкта.

Метод – поведінка об'єкта.

Об'єкт – окремий екземпляр класу.

Високий рівень – 1 бал

Завдання. Реалізувати клас, що має як мінімум 2 числових та 2 текстових поля, 4 методи та 4 конструктори. В основному коді створити 2 об'єкти класу та продемонструвати роботу з ними.

**Запитання для самоконтролю з теми: «Оброблення подій від миші,
клавіатури й елементів управління.»**

1. Використання "колекції" візуальних елементів управління формою.
2. Застосування обробників подій рівня форми.
3. Розміщення візуальних елементів управління на формі.
4. Використання SceneBuilder для розроблення графічних інтерфейсів користувача. Назвіть основні елементи синтаксису мови FXML.
5. Обґрунтуйте використання Eclipse для розроблення застосунків JavaFX.
6. Обґрунтуйте використання SceneBuilder для розроблення графічного інтерфейсу користувача.