

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ СЕМЕНА КУЗНЕЦЯ

ОСНОВИ ОБ'ЄКТНО-ОРІЄНТОВАНОГО
ПРОГРАМУВАННЯ

Робоча програма
для студентів спеціальності
122 "Комп'ютерні науки та інформаційні технології"
першого (бакалаврського) рівня

Харків
ХНЕУ ім. С. Кузнеця
2017

УДК 004.415(07.034)

О-75

Затверджено на засіданні кафедри інформаційних систем.
Протокол № 2 від 22.09.2016 р.

Самостійне електронне текстове мережеве видання

Укладачі: Ю. Е. Парфьонов
О. В. Щербаков
В. М. Федорченко

Основи об'єктно-орієнтованого програмування : робоча
О-75 програма для студентів спеціальності 122 "Комп'ютерні науки
та інформаційні технології" першого (бакалаврського) рівня :
[Електронне видання] / уклад. Ю. Е. Парфьонов, О. В. Щербаков,
В. М. Федорченко. – Харків : ХНЕУ ім. С. Кузнеця, 2017. – 52 с.

Наведено тематичний план навчальної дисципліни та її зміст за модулями
й темами, уміщено плани лекцій і лабораторних занять, матеріал для закріп-
лення знань (завдання для самостійної роботи, контрольні запитання), критерії
оцінювання знань студентів, професійні компетентності, якими повинен воло-
діти студент після вивчення дисципліни.

Рекомендовано для студентів спеціальності 122 "Комп'ютерні науки
та інформаційні технології".

УДК 004.415(07.034)

© Харківський національний економічний
університет імені Семена Кузнеця, 2017

Вступ

Сучасні умови господарювання потребують від фахівців з економічного управління всебічного використання новітніх інформаційних технологій. Широкі можливості комп'ютеризованих засобів щодо збирання та обробки необхідної інформації здатні значно підвищити якість економічних розрахунків, зробити більш ефективним процес обґрунтування економічних рішень.

Але використання потужних комп'ютеризованих засобів неможливе без програмного забезпечення. Важливість галузі розроблення програмного забезпечення збільшується, оскільки тенденції розвитку комп'ютерної техніки свідчать про те, що, з одного боку, складність і функціональні можливості комп'ютерної техніки постійно та швидко зростають, а з іншого – це потребує більш досконалих програмних засобів для задоволення потреб користувачів.

Істотною рисою таких програмних систем є рівень складності: для одного розробника практично неможливо охопити всі її аспекти. Причому ця складність є неминучою: із нею можна справитися, але позбавитися від неї неможливо.

У наш час найбільш поширеним методом боротьби зі складністю є об'єктно-орієнтований підхід до розроблення програмного забезпечення. Із використанням цього підходу розроблено більшу частину програм у всьому світі. Це потребує від відповідних фахівців чіткого уявлення концепцій об'єктно-орієнтованого програмування (ООП), що дає можливість їхнього практичного використання у процесі розроблення застосувань будь-якою мовою програмування.

Необхідним елементом успішного засвоєння навчального матеріалу дисципліни є самостійна робота студентів із технічною літературою та сучасними середовищами розроблення програм.

Навчальна дисципліна "Основи об'єктно-орієнтоване програмування" є базовою навчальною дисципліною, яку вивчають, згідно з навчальним планом підготовки фахівців освітнього ступеню "бакалавр" за спеціальністю 122 "Комп'ютерні науки та інформаційні технології".

1. Опис навчальної дисципліни

Найменування показників	Галузь знань, спеціальність, освітній ступінь	Характеристика навчальної дисципліни	
		денна форма навчання	заочна форма навчання
Кількість кредитів – 5	Галузь знань 12 "Інформаційні технології"	Базова	
Змістових модулів – 4	Спеціальність 122 "Комп'ютерні науки та інформаційні технології"	Рік підготовки	
Загальна кількість годин – 150		2-й	3-й
		Семестр	
	3-й	7-й	
Тижневих годин для денної форми навчання: аудиторних – 4 самостійної роботи студента – 5	Освітній ступінь: бакалавр	Лекції	
		34 год	12 год
		Лабораторні	
		34 год	12 год
		Самостійна робота	
		82 год	126 год
		Вид контролю	
іспит	іспит		

Співвідношення кількості годин аудиторних занять до самостійної та індивідуальної роботи становить:

для денної форми навчання – 82 %;

для заочної форми навчання – 19 %.

2. Мета та завдання навчальної дисципліни

Метою викладання навчальної дисципліни "Основи об'єктно-орієнтованого програмування" є формування компетентностей у використанні сучасної мови програмування та основних елементів об'єктно-орієнтованого підходу, необхідних для розроблення відповідних програмних застосувань.

Основними **завданнями** вивчення дисципліни "Основи об'єктно-орієнтованого програмування" є:

формування об'єктно-орієнтованого мислення;

володіння головними концепціями об'єктно-орієнтованого проектування, аналізу та програмування;

володіння об'єктно-орієнтованими мовами програмування;
формування навичок у розробленні об'єктно-орієнтованих програмних систем;

удосконалення навичок у використанні середовищ програмування.

Об'єктом навчальної дисципліни є основні елементи об'єктно-орієнтованого підходу до розроблення програмного забезпечення.

Предметом навчальної дисципліни є принципи об'єктно-орієнтованого програмування та основи об'єктно-орієнтованої технології.

Із метою найкращого засвоєння матеріалу, студенти повинні до початку вивчення дисципліни засвоїти теоретичні знання та опанувати практичні вміння з навчальних дисциплін "Програмування", "Алгоритми та структури даних", а також мати навички в роботі з персональним комп'ютером.

У процесі навчання студенти здобувають необхідні знання під час проведення аудиторних занять: лекційних і лабораторних. Велике значення у процесі вивчення та закріплення знань має самостійна робота студентів, у тому числі робота над курсовим проектом.

Зміст усіх видів занять розроблено, відповідно до кредитно-трансферної системи організації навчального процесу.

У результаті вивчення навчальної дисципліни студент повинен:

знати:

основні поняття об'єктно-орієнтованого аналізу, проектування та програмування;

принципи об'єктно-орієнтованого програмування;

поняття класу та об'єкта, співвідношення між ними;

типи відношень між класами;

порядок проектування класів;

життєвий цикл об'єктів;

особливості реалізації основних концепцій об'єктно-орієнтованого програмування в певних умовах програмування;

сучасні технології розроблення графічного інтерфейсу користувача;

певні мови програмування, технологічні засоби та способи їхнього використання у процесі виконання професійної діяльності та вирішення професійних завдань початкового рівня складності;

уміти:

продемонструвати здатність до об'єктно-орієнтованого мислення, застосування об'єктно-орієнтованого підходу під час проектування програмних систем; ґрунтовну підготовку до використання об'єктно-орієн-

тованих мов програмування, володіння комп'ютером, технологічними засобами та об'єктно-орієнтованою технологією;

проекувати компоненти програмного забезпечення;

проекувати людино-машинний інтерфейс інформаційних систем;

реалізовувати прототипи архітектури програмного забезпечення;

установлювати, налаштовувати й обслуговувати системне, інструментальне та прикладне програмне забезпечення та інформаційні системи.

У процесі викладання навчальної дисципліни основну увагу приділяють оволодінню студентами професійними компетентностями, наведеними в табл. 2.1.

Таблиця 2.1

Професійні компетентності, яких набувають студенти після вивчення навчальної дисципліни

Назва компетентності	Складові компетентності
Здатність до об'єктно-орієнтованого мислення, знання об'єктно-орієнтованої мови програмування та здатність застосовувати об'єктно-орієнтований підхід під час проектування програмних систем	Використовувати та проектувати компоненти програмного забезпечення
	Застосовувати мови програмування, інструментальні засоби під час проектування та створення програмних систем
	Проектувати людино-машинний інтерфейс програмних систем

Структуру складових професійних компетентностей та їхнє формування, відповідно до Національної рамки кваліфікацій України, наведено в табл. А.1 додатка А.

3. Програма навчальної дисципліни

Змістовий модуль 1

Основи об'єктно-орієнтованої парадигми

Тема 1. Основи Microsoft .NET та Java SE

1.1. Програмні платформи Microsoft .NET та Java SE: архітектура, компіляція та виконання програм, система типізації, стандартні бібліотеки класів, інструментальні засоби розроблення програм.

Тема 2. Основи об'єктно-орієнтованої мови програмування

2.1. Загальні відомості про мови C# та Java: алфавіт, типи даних, операції, оператори, структура програми, основи використання стандартних бібліотек класів Microsoft .NET та Java SE.

2.2. Одновимірні та багатовимірні масиви у C# та Java: створення, ініціалізація, оброблення, підтримка масивів у стандартних бібліотеках Microsoft .NET та Java SE.

2.3. Методи у C# та Java: визначення, механізми передачі параметрів, використання масиву як параметра, повернення масиву з методу, виклик методу.

Тема 3. Поняття об'єктно-орієнтованого аналізу, проектування та програмування

3.1. Об'єктно-орієнтована декомпозиція. Принципи об'єктно-орієнтованого підходу: абстракція, інкапсуляція, ієрархія, поліморфізм.

3.2. Поняття об'єкта. Характеристики об'єкта. Поняття класу. Співвідношення між класом та його об'єктом.

3.3. Об'єктно-орієнтований аналіз та його мета. Головні види вимог до програмної системи. Об'єктно-орієнтоване проектування. Об'єктно-орієнтоване програмування.

3.4. UML-діаграми класів. Відношення на діаграмі класів. CASE-засоби.

Змістовий модуль 2 Основні елементи ООП

Тема 4. Абстрагування даних та інкапсуляція

4.1. Абстрактні типи даних. Проектування абстрактного типу даних.

4.2. Класи та структури. Елементи класу. Особливості використання статичних елементів. Доступ до елементів класу, модифікатори доступу.

4.3. Поняття про створення, ініціалізацію та використання об'єктів класу. Посилання this.

4.4. 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".

Тема 5. Повторне використання класів

5.1. Поняття про асоціацію. Відношення композиції та агрегації як види асоціації. Реалізація композиції та агрегації в C# та Java.

5.2. Відношення успадкування. Реалізація відношення успадкування в C# та Java. Ініціалізація об'єкта базового класу. Використання конструкторів під час успадкування. Варіанти використання успадкування. Перевизначення методів.

5.3. Раннє та пізнє зв'язування. Віртуальні методи. Реалізація принципу поліморфізму в C# та Java. Рядкове подання об'єкта. Абстрактні класи та методи. Реалізація поліморфної поведінки на базі абстрактного класу. Інтерфейси. Реалізація поліморфної поведінки на базі інтерфейсу.

Тема 6. Принципи об'єктно-орієнтованого проектування класів

6.1. Система принципів SOLID. Принцип єдиної відповідальності.

6.2. Загальні відомості про шаблони проектування. Застосування основних шаблонів проектування.

Тема 7. Бібліотеки класів

7.1. Бібліотеки та їхнє використання. Статичні та динамічні бібліотеки.

7.2. Розроблення бібліотек на платформі Java SE. DLL-бібліотеки. Розроблення DLL-бібліотек на платформі Microsoft .NET.

Змістовий модуль 3

Оброблення винятків і бібліотеки класів

Тема 8. Оброблення виняткових ситуацій

8.1. Види помилок у програмах. Проблеми традиційного підходу до оброблення помилок.

8.2. Механізм оброблення винятків. Класи винятків стандартних бібліотек Microsoft .NET та Java SE. Синтаксис оброблення винятків.

Тема 9. Стандартні бібліотеки класів середовищ розробника програм

9.1. Призначення та застосування регулярних виразів. Підтримка регулярних виразів на платформах Microsoft .NET та Java SE. Спеціальні символи, які використовують у регулярних виразах.

9.2. Загальні відомості про колекції. Основні структури даних стандартних бібліотек колекцій Microsoft .NET та Java SE. Типізовані колекції.

9.3. Джерела та споживачі даних. Загальні відомості про потоки введення-виведення даних. Алгоритми роботи потоків введення-виведення даних. Основні класи стандартних бібліотек Microsoft .NET та Java SE для підтримки введення-виведення даних.

Змістовий модуль 4

Об'єктно-орієнтоване програмування застосувань із графічним інтерфейсом користувача

Тема 10. Основи розроблення графічних інтерфейсів користувача

10.1. Загальні відомості про події. Генерування подій. Обробники подій.

10.2. Огляд сучасних технологій розроблення застосувань із графічним інтерфейсом користувача на платформах Microsoft .NET та Java SE. Основи мови XML. Структура XML-документа. Простори імен XML. XML-схеми.

10.3. Загальні відомості про мову FXML. Використання FXML для опису графічного інтерфейсу користувача.

Тема 11. Розроблення графічних інтерфейсів користувача на платформі Java SE

11.1. Загальна структура застосування JavaFX. Форми. Події рівня форми. Використання форм та базових елементів управління JavaFX.

11.2. Оброблення подій від мишки, клавіатури та елементів управління. Використання основних елементів управління JavaFX.

4. Структура навчальної дисципліни

У процесі вивчення навчальної дисципліни "Основи об'єктно-орієнтованого програмування" студент має ознайомитися із програмою цієї дисципліни, її структурою, формами та методами навчання, видами та методами контролю за знаннями.

Тематичний план навчальної дисципліни "Основи об'єктно-орієнтованого програмування" складається з чотирьох змістових модулів, кожний із яких об'єднує в собі окремий, відносно самостійний блок, який логічно пов'язує кілька навчальних елементів цієї дисципліни за змістом та взаємозв'язками.

Структуру залікового кредиту навчальної дисципліни наведено в табл. 4.1.

Таблиця 4.1

Структура залікового кредиту навчальної дисципліни

Назви змістових модулів і тем	Кількість годин							
	денна форма				заочна форма			
	усього	у тому числі			усього	у тому числі		
		лекційні	лабораторні	самостійна робота		лекційні	лабораторні	самостійна робота
1	2	3	4	5	6	7	8	9
Змістовий модуль 1								
Основи об'єктно-орієнтованої парадигми								
<i>Тема 1.</i> Основи Microsoft .NET та Java SE	8	2	2	4	6	1	1	4
<i>Тема 2.</i> Основи об'єктно-орієнтованої мови програмування	8	2	2	4	11	1	1	9
<i>Тема 3.</i> Поняття об'єктно-орієнтованого аналізу, проектування та програмування	6	2	–	4	8	–	–	8
Разом за змістовим модулем 1	22	6	4	12	25	2	2	21
Змістовий модуль 2								
Основні елементи ООП								
<i>Тема 4.</i> Абстрагування даних та інкапсуляція	14	2	2	8	19	2	2	12
<i>Тема 5.</i> Повторне використання класів	28	8	8	10	24	2	2	20
<i>Тема 6.</i> Принципи об'єктно-орієнтованого проектування класів	12	2	4	6	10	–	–	10
<i>Тема 7.</i> Бібліотеки класів	6	2	–	4	10	–	–	10
Разом за змістовим модулем 2	60	14	14	28	63	4	4	52
Змістовий модуль 3								
Оброблення винятків і бібліотеки класів								
<i>Тема 8.</i> Оброблення виняткових ситуацій	8	2	–	6	11	1	–	10
<i>Тема 9.</i> Стандартні бібліотеки класів середовищ розробника програм	26	6	8	12	22	3	4	14
Разом за змістовим модулем 3	34	8	8	18	33	4	4	24

1	2	3	4	5	6	7	8	9
Змістовий модуль 4 Об'єктно-орієнтоване програмування застосувань із графічним інтерфейсом користувача								
<i>Тема 10. Основи розроблення графічних інтерфейсів користувача</i>	14	4	4	6	9	1	1	7
<i>Тема 11. Розроблення графічних інтерфейсів користувача на платформі Java SE</i>	14	2	4	8	14	1	1	12
Разом за змістовим модулем 4	28	6	8	14	23	2	2	19
<i>Підготовка до іспиту</i>	–	–	–	6	–	–	–	6
<i>Передекзаменаційні консультації</i>	–	–	–	1	–	–	–	1
<i>Іспит</i>	–	–	–	3	–	–	–	3
Усього годин	150	34	34	82	150	12	12	126

5. Теми лабораторних занять

Лабораторне заняття – це організаційна форма навчального заняття, на якому студенти під керівництвом викладача формують уміння й навички у практичному застосуванні основних теоретичних положень навчальної дисципліни шляхом виконання завдань до лабораторних робіт.

Лабораторні заняття з навчальної дисципліни "Основи об'єктно-орієнтоване програмування" проводять у спеціально обладнаному навчальному класі з використанням комп'ютерного устаткування, пристосованого до умов навчального процесу.

Із метою підвищення якості навчального процесу, під час проведення лабораторного заняття може бути призначено ще одного викладача й навчальну групу розподіляють на дві підгрупи. Кожний студент працює самостійно, виконуючи індивідуальне завдання для лабораторного дослідження.

На лабораторному занятті здійснюють поточний контроль за підготовленістю студентів до виконання конкретної лабораторної роботи, виконання завдань до теми заняття, оформлення індивідуального звіту з виконаної роботи та його захист перед викладачем.

Виконання лабораторної роботи оцінює викладач. Підсумкові оцінки, отримані студентом за виконання лабораторних робіт, ураховують під час виставлення семестрової підсумкової оцінки з навчальної дисципліни.

Тематику проведення лабораторних занять наведено в табл. 5.1.

Таблиця 5.1

План проведення лабораторних занять

Назви тем	Програмні питання	Кількість годин	Література
1	2	3	4
Змістовий модуль 1			
Основи об'єктно-орієнтованої парадигми			
<i>Тема 1.</i> Основи Microsoft .NET та Java SE	Основні поняття платформ Microsoft .NET та Java SE	4	Основна: [2; 3; 5]. Додаткова: [7; 8; 15; 16]
<i>Тема 2.</i> Основи об'єктно-орієнтованої мови програмування			
Змістовий модуль 2			
Основні елементи ООП			
<i>Тема 4.</i> Абстрагування даних та інкапсуляція	Розроблення застосувань із використанням базових елементів ООП	4	Основна: [1; 3; 5]. Додаткова: [7; 8; 15; 16]
<i>Тема 5.</i> Повторне використання класів			
<i>Тема 5.</i> Повторне використання класів	Застосування спадкування та поліморфізму	6	Основна: [1; 3; 5]. Додаткова: [7; 8; 15; 16]
<i>Тема 6.</i> Принципи об'єктно-орієнтованого проектування класів	Використання шаблонів проектування	4	Основна: [1; 3; 5]. Додаткова: [7; 15; 16]
Змістовий модуль 3			
Оброблення винятків і бібліотеки класів			
<i>Тема 9.</i> Стандартні бібліотеки класів середовищ розробника програм	Використання регулярних виразів	4	Основна: [2; 3; 5]. Додаткова: [7]
	Використання основних бібліотек Microsoft .NET Framework та Java SE	4	Основна: [2; 3; 5]. Додаткова: [7; 8; 15; 16]

1	2	3	4
Змістовий модуль 4 Об'єктно-орієнтоване програмування застосувань із графічним інтерфейсом користувача			
<i>Тема 10.</i> Основи розроблення графічних інтерфейсів користувача	Основи JavaFX	4	Основна: [10]. Додаткова: [9; 15; 16]
<i>Тема 11.</i> Розроблення графічних інтерфейсів користувача на платформі Java SE	Використання елементів управління JavaFX	4	Основна: [10]. Додаткова: [9; 15; 16]
Усього годин		34	

5.1. Приклади типових лабораторних завдань до тем

Змістовий модуль 1

Основи об'єктно-орієнтованої парадигми

Тема 1. Основи Microsoft .NET та Java SE. Тема 2. Основи об'єктно-орієнтованої мови програмування

Використовуючи елементи класу System.Math, розробити консольну програму-"калькулятор" мовою C# для виконання будь-якої з чотирьох арифметичних дій ("+", "-", "x", "/") із двома числами й додаткової операції exp(a). Додаткову операцію має бути реалізовано у вигляді статичної функції класу (*рівні 2, 3*) або всі операції калькулятора – у вигляді статичних функцій класу (*рівень 4*). Вихідні дані (два числа та знак операції) мають вводити з консолі. Результат виконання операції – на консоль. Числове значення результату мають виводити з використанням функцій форматованого виведення з трьома знаками після коми (*рівень 3*) або бажаною кількістю знаків після коми, яку вводить користувач програми з консолі (*рівень 4*). Програма має дозволяти одноразово ввести вихідні дані, визначити результат обчислення та завершити роботу (*рівні 1, 2*) або багаторазово повторювати процес уведення вихідних даних і визначення результату обчислення до введення користувачем певного рядка символів, після чого завершити роботу (*рівні 3, 4*).

Змістовий модуль 2 Основні елементи ООП

Тема 4. Абстрагування даних та інкапсуляція. Тема 5. Повторне використання класів

Розробити застосування мовою Java, призначене для автоматизації оброблення відомості нарахування зарплати співробітникам підприємства (табл. 5.2):

Таблиця 5.2

Відомість нарахування зарплати

Прізвище	Зарплата, грн	Утримано, грн	Видано, грн
F	Z	P	$S = Z - P$
...
Разом	Σ	Σ	Σ

Для цього необхідно створити два класи: Клас 1 та Клас 2.

Клас 1 призначено для опису будь-якого із записів відомості. Поля цього класу мають відповідати полям відомості, призначеним для зберігання вихідних даних. Отримання поточних значень полів, розрахунки за формулами мають виконувати за допомогою відповідних нестатичних методів цього класу. Для встановлення значень полів мають використовувати конструктор із параметрами.

Клас 2 має містити статичні методи: метод `main()` – точку входу у програму; необхідну кількість методів для підрахунку підсумкових даних відомості; метод для виведення всіх даних відомості на консоль.

У методі `main()` має бути визначено масив об'єктів типу *Клас 1* для зберігання відповідних об'єктів.

Застосування має забезпечувати:

можливість використання текстового меню для вибору варіантів дій користувача;

уведення з консолі кількості записів, що буде міститися у відомості;

уведення з консолі значень вихідних полів кожного запису відомості, створення об'єктів *Класу 1*, кількість яких відповідає кількості записів відомості, додавання об'єктів *Класу 1* у масив;

обчислення підсумкових даних;

виведення на консоль вихідних та розрахункових даних кожного запису відомості, а також підсумкової інформації відомості у вигляді

необхідної кількості колонок із заголовками (*рівні 1, 2*) або справжньої таблиці з горизонтальними та вертикальними лініями сітки (числові значення мають виводитися з певною кількістю знаків після коми з використанням засобів форматного виведення (*рівні 3, 4*);
усунення помилок під час виконання програми (*рівень 4*).

Змістовий модуль 3

Оброблення винятків і бібліотеки класів

Тема 9. Стандартні бібліотеки класів середовищ розробника програм

Розробити консольну програму для виконання аналізу вхідної послідовності символів, що вводить користувач, із метою визначення її належності до граматики, яка описує поняття "умовний_оператор", де:

умовний_оператор ::= if (умова) оператор {else оператор}?

оператор ::= {змінна++;} | {змінна--;} |

умова ::= змінна {< | >} змінна

змінна ::= літера⁺цифра^{*}

літера ::= a|b|c

цифра ::= 0|1|2|3|4|5|6|7|8|9

Програма має містити блок синтаксичного аналізу вхідної послідовності символів на базі регулярних виразів.

Якщо вхідна послідовність припустима, то програма має видавати відповідне повідомлення користувачу, інакше – аналогічним чином повідомляти його про неприпустимість вхідної послідовності.

У будь-якому випадку далі виконання програми має тривати до введення користувачем рядка-ознаки завершення програми.

Змістовий модуль 4

Об'єктно-орієнтоване програмування застосувань із графічним інтерфейсом користувача

Тема 10. Основи розроблення графічних інтерфейсів користувача

Розробити застосування JavaFX, що відповідає таким вимогам:

має одне вікно;

колір клієнтської області вікна встановлено за замовчуванням (*рівень 1*) або використано один зі стандартних кольорів, відповідно до варіанта завдання (*рівні 2 – 4*);

піктограму системного меню встановлено за замовчуванням (*рівні 1, 2*) або використано власну піктограму (*рівні 3, 4*);

курсор мишки встановлено за замовчуванням (*рівень 1*) або використано один зі стандартних курсорів, відповідно до варіанта завдання (*рівні 2, 3*) або власний курсор (завантажується з файла) (*рівень 4*);

обробляє подію клацання кнопки мишки або натискання клавіш на клавіатурі комп'ютера, відповідно до варіанта завдання;

в обробнику події клацання кнопки мишки необхідно визначити, чи відповідає натиснута кнопка (ліва або права), а також кількість клацань (один або два) зазначеним у варіанті завдання, і, якщо це так, показати стандартне діалогове вікно із кнопкою "ОК" і повідомленням про натиснуту кнопку мишки (ліву або праву) (*рівень 1*) або стандартне діалогове вікно з повідомленням, у якому відображено поточні координати курсора мишки в момент клацання, а також кнопки або піктограму, відповідно до варіанта завдання (*рівні 2 – 4*);

в обробнику події натискання клавіш на клавіатурі необхідно показати стандартне діалогове вікно з кнопкою "ОК" і повідомленням про натиснуту клавішу (*рівень 1*) чи стандартне діалогове вікно з повідомленням, у якому відображено символічний код натиснутої клавіші, або символ, відповідний натиснутій алфавітно-цифровій клавіші, а також кнопки або піктограму, відповідно до варіанта завдання (*рівні 2 – 4*).

6. Самостійна робота студентів

Для опанування матеріалу навчальної дисципліни "Основи об'єктно-орієнтованого програмування", окрім лекційних та лабораторних занять, тобто аудиторної роботи, значну увагу необхідно приділяти самостійній роботі.

Самостійна робота студента (СРС) – це форма організації навчального процесу, за якої заплановані завдання студент виконує самостійно під методичним керівництвом викладача.

Мета СРС – засвоєння в повному обсязі навчальної програми та формування у студентів загальних і професійних компетентностей, які відіграють суттєву роль у становленні майбутнього фахівця вищого рівня кваліфікації.

Навчальний час, відведений для самостійної роботи студентів денної форми навчання, визначено навчальним планом, і він становить 55 % (82 години) від загального обсягу навчального часу на вивчення навчальної дисципліни (150 годин). У ході самостійної роботи студент має стати активним учасником навчального процесу, навчитися свідомо ставитися до оволодіння теоретичними та практичними знаннями, вільно орієнтуватися в інформаційному просторі, брати на себе індивідуальну відповідальність за якість власної професійної підготовки. СРС містить: опрацювання лекційного матеріалу; опрацювання та вивчення рекомендованої літератури, основних термінів та понять за темами навчальної дисципліни; підготовку до лабораторних занять; поглиблене опрацювання окремих лекційних тем або питань; виконання індивідуальних завдань; пошук (підбір) та огляд літературних джерел на задану тематику навчальної дисципліни; контрольну перевірку студентами особистих знань за запитаннями для самодіагностики; підготовку до контрольних робіт та інших форм контролю; систематизацію вивченого матеріалу, із метою підготовки до семестрового екзамену.

Необхідним елементом успішного засвоєння матеріалу навчальної дисципліни є самостійна робота студентів із вітчизняною та зарубіжною технічною літературою в галузі об'єктно-орієнтованого програмування. Основні види самостійної роботи, запропоновані студентам для засвоєння теоретичних знань із навчальної дисципліни, наведено в табл. 6.1.

Таблиця 6.1

Завдання для самостійної роботи студентів та форми контролю

Назви тем	Зміст самостійної роботи	Кількість годин	Форми контролю	Література
1	2	3	4	5
Змістовий модуль 1. Основи об'єктно-орієнтованої парадигми				
<i>Тема 1.</i> Основи Microsoft .NET та Java SE	Призначення віртуальної машини та основних стандартних бібліотек класів	4	Оцінювання результатів виконання л/р	Основна: [2 – 4]. Додаткова: [7; 8; 15; 16]
<i>Тема 2.</i> Основи об'єктно-орієнтованої мови програмування	Використання коментарів. Налягодження програм у середовищі програмування	4		Основна: [2 – 4]. Додаткова: [7; 8; 15; 16]

12	2	3	4	5
<i>Тема 3.</i> Поняття об'єктно-орієнтованого аналізу, проектування та програмування	Ознаки складних програмних систем. Переваги об'єктно-орієнтованого підходу до розроблення програмних систем	4		Основна: [1]. Додаткова: [6]
Разом за змістовим модулем 1		12		
Змістовий модуль 2. Основні елементи ООП				
<i>Тема 4.</i> Абстрагування даних та інкапсуляція	Принципи об'єктно-орієнтованого підходу. Основні властивості конструктора. Конструктор за замовчуванням. "Збирання сміття"	8	Оцінювання результатів виконання л/р	Основна: [1; 2; 4]. Додаткова: [7; 8; 15; 16]
<i>Тема 5.</i> Повторне використання класів	Пізнє та раннє зв'язування. Використання конструкторів за успадкування	10		Основна: [2 – 4]. Додаткова: [7; 8; 15; 16]
<i>Тема 6.</i> Принципи об'єктно-орієнтованого проектування класів	Принципи SOLID та їхнє використання	6		Основна: [2 – 4]. Додаткова: [7; 8; 15; 16]
<i>Тема 7.</i> Бібліотеки класів	Проблема "Аду DLL" та напрями її вирішення	4		Основна: [2, 5]. Додаткова: [6]
Разом за змістовим модулем 2		28		
Змістовий модуль 3. Оброблення винятків і бібліотеки класів				
<i>Тема 8.</i> Оброблення виняткових ситуацій	Розроблення власних класів винятків. Генерування та перехоплення користувальницьких винятків	6	Оцінювання результатів виконання л/р	Основна: [2 – 4].
<i>Тема 9.</i> Стандартні бібліотеки класів середовищ розробника програм	Хеш-таблиці та хеш-функції. Сортування вмісту контейнера. Розроблення користувальницьких контейнерів. Засоби форматування рядків	12		Основна: [2 – 4]. Додаткова: [7]
Разом за змістовим модулем 3		18		

1	2	3	4	5
Змістовий модуль 4. Об'єктно-орієнтоване програмування застосувань із графічним інтерфейсом користувача				
<i>Тема 10.</i> Основи розроблення графічних інтерфейсів користувача	Декларативний опис елементів графічних інтерфейсів користувача. Інтерфейси зворотного виклику. Події та їхнє оброблення	6	Оцінювання результатів виконання л/р	Основна: [5]. Додаткова: [9; 15; 16]
<i>Тема 11.</i> Розроблення графічних інтерфейсів користувача на платформі Java SE	Використання "колекції" візуальних елементів управління форми. Застосування обробників подій рівня форми. Розміщення візуальних елементів управління на формі. Використання SceneBuilder для розроблення графічних інтерфейсів користувача	8		Основна: [5]. Додаткова: [9; 15; 16]
Разом за змістовим модулем 4		14		
<i>Підготовка до екзамену</i>		6		
<i>Передекзаменаційні консультації</i>		1		
<i>Екзамен</i>		3		
Усього годин		82		

6.1. Контрольні запитання для самодіагностики

Тема 1. Основи Microsoft .NET та Java SE

1. Для чого призначено віртуальну машину?
2. Як створюють проект консольного застосування в середовищі програмування?
3. Як відкрити проект у середовищі програмування?
4. Як додати файл до проекту в середовищі програмування?
5. Як виконати компіляцію та запуск програми на виконання в середовищі програмування?
6. Як виконати налагодження програми в середовищі програмування?

Тема 2. Основи об'єктно-орієнтованої мови програмування

1. Для чого призначено головний метод програми?
2. Для чого призначено посилання `this` та як його використовувати?
3. Які існують способи передачі параметрів методів?
4. Охарактеризуйте структуру програми на мові C# та Java.
5. Перелічіть убудовані типи даних C# та Java.
6. Яке призначення мають методи класу `Math`?

Тема 3. Поняття об'єктно-орієнтованого аналізу, проектування та програмування

1. Охарактеризуйте прості та складні програмні системи.
2. Як можна боротися зі складністю програм?
3. У чому полягає сутність об'єктно-орієнтованої декомпозиції?
4. Які основні елементи має об'єктно-орієнтована технологія?
5. Яке співвідношення існує між класами та об'єктами?
6. Які характеристики має будь-який об'єкт?
7. Сформулюйте принципи об'єктно-орієнтованого підходу.

Тема 4. Абстрагування даних та інкапсуляція

1. Який синтаксис опису класу?
2. Які специфікатори доступу до елементів класу є в мові програмування?
3. Який порядок ініціалізації об'єкта класу?

Тема 5. Повторне використання класів

1. Що таке "агрегація" та "успадкування"?
2. Наведіть приклад синтаксису успадкування в C#.
3. Яка послідовність виклику конструкторів за успадкування?
4. Як перевизначити "базовий" метод?
5. У чому полягає принцип поліморфізму?
6. Які переваги має концепція поліморфізму?
7. Що таке "абстрактні класи" та яке призначення вони мають?
8. Які правила використання абстрактних класів існують?
9. Дайте характеристику поняття "інтерфейси" та їхнього призначення.
10. Які правила використання інтерфейсів існують?

Тема 6. Принципи об'єктно-орієнтованого проектування класів

1. У чому полягає сутність принципу єдиної відповідальності?
2. У чому полягає сутність принципу відкриття-закриття?
3. У чому полягає сутність принципу інверсії залежностей?
4. У чому полягає сутність принципу заміщення Барбери Лісков?
5. У чому полягає сутність принципу ізоляції інтерфейсу?
6. Для чого використовують шаблон "Абстрактна фабрика"?
7. Для чого використовують шаблон "Фабричний метод"?
8. Для чого використовують шаблон "Адаптер"?
9. Для чого використовують шаблон "Компонувальник"?
10. Для чого використовують шаблон "Фасад"?
11. Для чого використовують шаблон "Команда"?
12. Для чого використовують шаблон "Спостерігач"?
13. Для чого використовують шаблон "Стратегія"?

Тема 7. Бібліотеки класів

1. Як створити DLL-бібліотеку на платформі Microsoft .NET?
2. Як створити JAR-бібліотеку на платформі Java SE?

Тема 8. Оброблення виняткових ситуацій

1. Які проблеми має традиційний підхід до оброблення помилок під час виконання програми?
2. Які переваги оброблення винятків, порівняно із традиційним підходом до оброблення помилок?

Тема 9. Стандартні бібліотеки класів середовищ розробника програм

1. Охарактеризуйте склад бібліотеки контейнерів Java SE та Microsoft .NET.
2. Які основні інтерфейси існують у бібліотеці контейнерів Java SE та Microsoft .NET?
3. Дайте коротку характеристику структури даних "динамічний масив" і її реалізації в Java SE та Microsoft .NET.
4. Дайте коротку характеристику структури даних "двозв'язний список" і її реалізації в Java SE та Microsoft .NET.
5. Що таке "хеш-таблиця" й "хеш-функція"?
6. Дайте коротку характеристику структури даних "асоціативний масив" і її реалізації в Java SE та Microsoft .NET.

7. Які засоби розбору рядків на лексеми є у Java SE та Microsoft .NET?
8. Які особливості реалізації рядкового типу даних є у Java SE та Microsoft .NET?
9. Охарактеризуйте класи стандартної бібліотеки Java SE та Microsoft .NET для подання рядків.
10. До яких елементів програми можливе застосування атрибутів?
11. Для чого використовують атрибути умовної компіляції?
12. Як створити клас, об'єкти котрого можливо серіалізувати?
13. Які формати серіалізації існують?
14. Укажіть базові класи байтових та символьних потоків уведення-виведення Java SE та Microsoft .NET.
15. Для чого призначено основні класи символьних потоків уведення-виведення Java SE та Microsoft .NET?
16. Для чого призначено основні класи байтових потоків уведення-виведення Java SE та Microsoft .NET?

Тема 10. Основи розроблення графічних інтерфейсів користувача

1. Що таке "подія"?
2. Дайте характеристику механізму оброблення подій.
3. Обґрунтуйте використання слухачів подій у Java SE.
4. Назвіть основні елементи синтаксису мови XML.

Тема 11. Розроблення графічних інтерфейсів користувача на платформі Java SE

1. Назвіть основні елементи синтаксису мови FXML.
2. Обґрунтуйте використання Eclipse для розроблення застосунків JavaFX.
3. Обґрунтуйте використання SceneBuilder для розроблення графічного інтерфейсу користувача.

7. Індивідуально-консультативна робота

Індивідуально-консультативну роботу здійснюють за графіком індивідуально-консультативної роботи у формі індивідуальних занять, консультацій, перевірки виконання індивідуальних завдань, перевірки та захисту завдань, винесених на поточний контроль, тощо.

Із теоретичної частини навчальної дисципліни індивідуально-консультативну роботу здійснюють у формі:

1) індивідуальних консультацій, на яких студент дістає відповідь від викладача на конкретні запитання або пояснення певних теоретичних положень чи аспектів їхнього практичного застосування;

2) групових консультацій, на яких викладач розглядає типові приклади з використання концепцій об'єктно-орієнтованого програмування.

Із практичної частини навчальної дисципліни індивідуально-консультативну роботу здійснюють у формі:

1) індивідуальних консультацій, на яких викладач розглядає лабораторні завдання, щодо яких виникли запитання у студента;

2) групових консультацій, на яких викладач розглядає практичні ситуації, які потребують колективного обговорення.

Індивідуально-консультативну роботу для комплексного оцінювання засвоєння програмного матеріалу здійснюють у формі:

1) індивідуального захисту самостійних та індивідуальних завдань;

2) підготовки рефератів для виступу на науковому семінарі;

3) підготовки рефератів для виступу на науковій конференції.

8. Методи навчання

У процесі викладання навчальної дисципліни "Основи об'єктно-орієнтованого програмування" для активізації навчально-пізнавальної діяльності студентів передбачено застосування як активних, так і інтерактивних навчальних технологій, серед яких: лекції проблемного характеру, робота в малих групах, семінари-дискусії, презентації.

Лекції проблемного характеру – один із найважливіших елементів проблемного навчання студентів. Вони передбачають разом із розглядом основного лекційного матеріалу встановлення та розгляд кола проблемних питань дискусійного характеру, які недостатньо розроблено в науці й мають актуальне значення для теорії та практики. Лекції проблемного характеру відрізняються поглибленою аргументацією матеріалу, що викладають. Вони сприяють формуванню у студентів самостійного творчого мислення, прищеплюють їм пізнавальні навички. Студенти стають учасниками наукового пошуку та вирішення проблемних ситуацій.

Міні-лекції передбачають викладення навчального матеріалу за короткий проміжок часу й характеризуються значною наповненістю, складністю логічних побудов, образів, доказів та узагальнень. Їх проводять, переважно, як частину заняття-дослідження. Міні-лекції відрізняються від повноформатних лекцій значно меншою тривалістю. Зазвичай міні-лекції тривають не більше ніж 10 – 15 хвилин і їх використовують для того, щоб стисло повідомити нову інформацію всім слухачам. Міні-лекції часто застосовують як частини цілісної теми, яку бажано викладати повноформатною лекцією, щоб не втомлювати аудиторію. Тоді інформацію надають по черзі кількома окремими сегментами, між якими застосовують інші форми й методи навчання.

Семінари-дискусії передбачають обмін думками й поглядами учасників щодо цієї теми, а також розвивають мислення, допомагають формувати погляди й переконання, виробляють уміння формулювати думки та висловлювати їх.

Робота в малих групах дає змогу структурувати практично-семинарські заняття за формою та змістом, створює можливості для участі кожного студента в роботі над темою заняття, забезпечує формування особистісних якостей і досвіду соціального спілкування.

Презентації – виступи перед аудиторією, що використовують для подання певних досягнень, результатів роботи групи звіту про виконання індивідуальних завдань, проектних робіт. Презентації можуть бути як індивідуальними, наприклад виступ одного слухача, так і колективними, тобто виступи двох та більше слухачів.

Розподіл форм і методів активізації процесу навчання за темами навчальної дисципліни "Основи об'єктно-орієнтованого програмування" наведено в табл. 8.1.

Таблиця 8.1

**Розподіл форм та методів активізації процесу навчання
за темами навчальної дисципліни**

Теми	Практичне застосування навчальних технологій
1	2
<i>Тема 1.</i> Основи Microsoft .NET та Java SE	Лекція проблемного характеру з питання "Основні поняття платформ Microsoft .NET та Java SE"

1	2
<i>Тема 2.</i> Основи об'єктно-орієнтованої мови програмування	Лекція проблемного характеру з питання "Основи мов програмування Java та C#"
<i>Тема 3.</i> Поняття об'єктно-орієнтованого аналізу, проектування та програмування	Лекція проблемного характеру з питання "Основні концептуальні засади об'єктно-орієнтованого програмування"
<i>Тема 4.</i> Абстрагування даних та інкапсуляція	Лекція проблемного характеру з питання "Основи об'єктно-орієнтованого програмування мовами C# та Java"
<i>Тема 5.</i> Повторне використання класів	Міні-лекції з питань "Агрегація та композиція", "Успадкування", "Поліморфізм", "Абстрактні класи та інтерфейси"; семінар-дискусія
<i>Тема 6.</i> Принципи об'єктно-орієнтованого проектування класів	Лекція проблемного характеру з питання "Принципи SOLID та їхнє практичне використання"
<i>Тема 7.</i> Бібліотеки класів	Лекція проблемного характеру з питання "Розроблення бібліотек класів"
<i>Тема 8.</i> Оброблення виняткових ситуацій	Міні-лекція з питання "Винятки", семінар-дискусія
<i>Тема 9.</i> Стандартні бібліотеки класів середовищ розробника програм	Лекції проблемного характеру з питань "Використання регулярних виразів", "Колекції", "Уведення-виведення даних"
<i>Тема 10.</i> Основи розроблення графічних інтерфейсів користувача	Лекція проблемного характеру з питання "Сучасні технології розроблення графічних інтерфейсів користувача"
<i>Тема 11.</i> Розроблення графічних інтерфейсів користувача на платформі Java SE	Лекції проблемного характеру з питань "Основи JavaFX", "Використання елементів управління JavaFX"

Використання методик активізації процесу навчання наведено в табл. 8.2.

Використання методик активізації процесу навчання

Теми навчальної дисципліни	Практичне застосування методик	Методики активізації процесу навчання
Тема 5. Повторне використання класів	<i>Лабораторне заняття.</i> Тема: "Застосування успадкування та поліморфізму"	Робота в малих групах; презентації
Тема 8. Стандартні бібліотеки класів середовищ розробника програм	<i>Лабораторне заняття.</i> Тема: "Використання основних бібліотек Microsoft .NET Framework та Java SE"	Робота в малих групах; презентації
Тема 10. Розроблення графічних інтерфейсів користувача на платформі Java SE	<i>Лабораторне заняття.</i> Тема: "Використання елементів управління JavaFX"	Робота в малих групах; презентації

9. Методи контролю

Система оцінювання сформованих компетентностей (див. табл. 2.1) у студентів урахує види занять, які, згідно із програмою навчальної дисципліни, передбачають лекційні, лабораторні заняття, а також самостійну роботу. Оцінювання сформованих компетентностей у студентів здійснюють за накопичувальною 100-бальною системою. Відповідно до Тимчасового положення "Про порядок оцінювання результатів навчання студентів за накопичувальною бальна-рейтинговою системою" ХНЕУ ім. С. Кузнеця, контрольні заходи містять:

поточний контроль, що здійснюють протягом семестру під час проведення лекційних, лабораторних занять та оцінюють за сумою набраних балів (максимальна сума – 60 балів, а мінімальна сума, що дозволяє студенту скласти іспит, – 35 балів);

модульний контроль, що здійснюють з урахуванням поточного контролю за відповідний змістовий модуль, він має на меті інтегроване оцінювання результатів навчання студента після вивчення матеріалу з логічно завершеної частини дисципліни – змістового модуля;

підсумковий/семестровий контроль проводять у формі семестрового екзамену, відповідно до графіка навчального процесу.

Поточний контроль із цієї навчальної дисципліни проводять у таких формах:

активна робота на лекційних заняттях;
активна участь у виконанні завдань лабораторних занять;
активна участь у дискусії та презентації матеріалу на заняттях;
проведення поточного тестування або письмової контрольної роботи;
експрес-опитування.

Модульний контроль із цієї навчальної дисципліни проводять у формі тесту. Тест містить запитання одиничного та множинного вибору, а також запитання відкритого типу щодо перевірки знань основних категорій навчальної дисципліни та їхнього практичного використання. Приклади завдань модульного контролю наведено далі.

Приклад тестових завдань за модулем 1

Варіант 1

1. Є програма:

```
using System; // №1
namespace ConsoleTest
{
    class Program
    {
        static void Main( )
        {
            Console.WriteLine("Привет, мир");
        }
    }
}
```

Припустіть, що інструкція № 1 відсутня. Змініть програму таким чином, щоб вона залишилася працездатною.

2. Знайдіть відповідності між пунктами двох списків.

Список 1:

1. Термін "змінна екземпляра" – це друга назва для ____.
2. Термін "змінна класу" – це друга назва для ____.
3. Термін "метод екземпляра" – це друга назва для ____.

4. Локальну змінну оголошено всередині ____.

5. ____ становить "контейнер" для зберігання фіксованої кількості значень одного типу.

Список 2: а) *об'єкт*; б) *статичне поле*; в) *статичний метод*; г) *метод*; д) *масив*; е) *клас*; є) *структура*; ж) *поле*.

Приклад тестових завдань за модулем 2

Варіант 1

1. Ваш друг написав клас та назвав його CoolClass. Відомо, що в цьому класі є конструктор за замовчуванням. Напишіть код на Java, який призначено для створення одного об'єкта класу CoolClass та його ініціалізації за допомогою цього конструктора.

2. Знайдіть помилки у цьому коді C#:

```
class Base
{
    string GetBeta( )
    {
        return "beta";
    }
}

class Derived : Base
{
    void GetGamma( )
    {
        return GetBeta();
    }
}
```

Приклад тестових завдань за модулем 3

Варіант 1

1. Для пошуку деяких фрагментів тексту у програмі використовують регулярний вираз: $\wedge(a|b|v)+\$$.

Які фрагменти будуть знайдені:

- а) в;
- б) абвабв;
- в) абвб;
- г) аааааааа;
- д) аабв;
- е) вб;
- є) а;
- ж) бв;
- з) б;
- и) бб.

2. Виберіть помилкові висловлювання:

- а) у разі виникнення будь-якого винятку блоки catch виконують у порядку їхнього слідування;
- б) тільки один із блоків catch може отримати управління в разі виникнення деякого винятку;
- в) будь-який блок catch завжди генерує виняток;
- г) будь-блок catch після коригування причини, що викликала виняток, повертає управління в точку його виникнення.

Приклад тестових завдань за модулем 4

Варіант 1

1. Є вихідний код методу start застосування JavaFX:

```
public class Main extends Application {  
    @Override  
    public void start(Stage primaryStage) {  
        AnchorPane root = new AnchorPane();  
        Scene scene = new Scene(root,400,400);  
        primaryStage.setScene(scene);  
        //Вставте Ваш код сюди  
        primaryStage.show(); }  
}
```

Що потрібно додати до цього коду, замість рядка з коментарем, щоб установити червоний колір фону клієнтської області вікна:

- a) scene.setFill(Color.RED);
- б) root.setFill(Color.RED);
- в) primaryStage.setFill(Color.red);
- г) scene.setFill(Color.red);
- д) root.setFill(Color.red);
- е) primaryStage.setFill(Color.red).

2. Виберіть правильні варіанти підключення обробника події ActionEvent до кнопки (btn – ім'я посилання на об'єкт кнопки, void myHandler(ActionEvent ae) – метод-обробник події):

- a) btn.setOnAction(this::myHandler);
- б) btn.setOnAction(e -> myHandler(e));
- в) addEventHandler(ActionEvent.Action, e -> myHandler(e));
- г) addEventHandler(ActionEvent.Action, e -> myHandler);
- д) btn.setOnAction(this::myHandler(e));
- е) btn.setOnAction(e -> myHandler).

Підсумковий/семестровий контроль проводять у формі семестрового екзамену. **Семестрові екзамени** – форма оцінювання підсумкового засвоєння студентами теоретичного та практичного матеріалу з окремої навчальної дисципліни, що проводять як контрольний захід.

Порядок здійснення поточного оцінювання знань студентів. Оцінювання знань студента під час лабораторних здійснюють за накопичувальною 100-бальною системою за такими критеріями:

розуміння, ступінь засвоєння теорії та методології проблем, що розглядають;

ступінь засвоєння фактичного матеріалу навчальної дисципліни;

ознайомлення з рекомендованою літературою, а також сучасною літературою з питань, що розглядають;

уміння поєднувати теорію із практикою під час розгляду виробничих ситуацій, розв'язання задач;

логіка, структура, стиль викладу матеріалу в письмових роботах і під час виступів в аудиторії, уміння обґрунтовувати свою позицію, здійснювати узагальнення інформації та робити висновки;

логічна, арифметична правильність та ступінь відповідності наведених рішень вимогам до них.

Максимально можливий бал за конкретним завданням ставлять за умови відповідності індивідуального завдання студента або його усної відповіді всім зазначеним критеріям. Відсутність тієї або іншої складової частини зменшує кількість балів. Під час оцінювання індивідуальних завдань увагу також приділяють якості, самостійності та своєчасності здавання виконаних завдань викладачу, згідно із графіком навчального процесу. Якщо якусь із вимог не буде виконано, то бали буде знижено.

Критерії оцінювання позааудиторної самостійної роботи студентів. Загальними критеріями, за якими здійснюють оцінювання позааудиторної самостійної роботи студентів, є: глибина й міцність знань, рівень мислення, уміння систематизувати знання за окремими темами, уміння робити обґрунтовані висновки, володіння категорійним апаратом, навички та прийоми виконання практичних завдань, уміння знаходити необхідну інформацію, здійснювати її систематизацію та оброблення, самореалізація на лабораторних заняттях.

Порядок здійснення підсумкового контролю з навчальної дисципліни. Підсумковий контроль за знаннями та компетентностями студентів із навчальної дисципліни здійснюють на підставі проведення семестрового екзамену. Екзаменаційний білет охоплює програму дисципліни й передбачає визначення рівня знань та ступеня опанування студентами компетентностей (див. табл. 2.1).

Завданням екзамену є перевірка розуміння студентом програмного матеріалу загалом, логіки та взаємозв'язків між окремими розділами, до здатності творчого використання накопичених знань, тощо. В умовах реалізації компетентнісного підходу екзамен оцінює рівень засвоєння студентом компетентностей, передбачених кваліфікаційними вимогами. Кожен екзаменаційний білет складається із практичних ситуацій, які передбачають вирішення типових професійних завдань фахівця на робочому місці та дозволяють діагностувати рівень теоретичної підготовки студента й рівень його компетентності з навчальної дисципліни.

Екзаменаційний білет містить два евристичних завдання, які оцінюють, відповідно до Тимчасового положення "Про порядок оцінювання результатів навчання студентів за накопичувальною бально-рейтинговою системою" ХНЕУ ім. С. Кузнеця.

Студент, який із поважних причин, підтверджених документально, не мав можливості брати участь у формах поточного контролю, тобто

не склав змістовий модуль, має право на його відпрацювання у двотижневий термін після повернення до навчання за розпорядженням декана факультету, відповідно до встановленого терміну.

Студента **не може бути допущено** до складання екзамену, якщо кількість балів, набраних за результатами перевірки успішності під час поточного та модульного контролю, відповідно до змістового модуля, упродовж семестру, у сумі не досягла 35 балів. Після екзаменаційної сесії декан факультету видає розпорядження про ліквідацію академічної заборгованості. У встановлений термін студент добирає залікові бали.

Студента слід **уважати атестованим**, якщо сума балів, набраних за результатами підсумкової/семестрової перевірки успішності, дорівнює або перевищує 60. Мінімально можлива кількість балів за поточний і модульний контроль упродовж семестру – 35 та мінімально можлива кількість балів, набраних на екзамені, – 25.

Результат семестрового екзамену оцінюють у балах (максимальна кількість – 40 балів, мінімальна кількість, що зараховують, – 25 балів) і проставляють у відповідній графі *екзаменаційної відомості обліку успішності*.

Підсумкову оцінку з навчальної дисципліни розраховують з урахуванням балів, набраних під час екзамену, та балів, набраних під час поточного контролю, за накопичувальною системою. Сумарний результат у балах за семестр становить: *"60 і більше балів – зараховано"*, *"59 і менше балів – не зараховано"* та його заносять у *залікову відомість обліку успішності* навчальної дисципліни. У разі набрання менше ніж 60 балів студент обов'язково складає залік після закінчення екзаменаційної сесії у встановлений деканом факультету термін, але не пізніше від двох тижнів після початку семестру. У разі повторного набрання менше ніж 60 балів декан факультету призначає комісію у складі трьох викладачів на чолі із завідувачем кафедри та визначає термін перескладання заліку, після чого приймає рішення, відповідно до чинного законодавства: якщо "зараховано" – студент продовжує навчання за графіком навчального процесу, а якщо "не зараховано", тоді декан факультету пропонує студенту повторне вивчення навчальної дисципліни протягом наступного навчального періоду самостійно.

Проведення підсумкового іспиту

Іспит проводять на комп'ютерах за екзаменаційними білетами.

Зразок екзаменаційного білета

Харківський національний економічний університет імені Семена Кузнеця

Освітній ступінь: бакалавр

Спеціальність: 122 "Комп'ютерні науки та інформаційні технології"

Семестр: III

Навчальна дисципліна: "Основи об'єктно-орієнтованого програмування"

ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 1

Завдання 1 (евристичне). Є відношення успадкування: геометричне тіло (Solid) – абстрактний базовий клас, прямокутний паралелепіпед (RectSolid) та куб (Cube) – похідні класи. Клас Solid має абстрактний метод `GetVolume()`, призначений для обчислення та повернення значення об'єму геометричного тіла. Клас Cube має поле `A` – довжина ребра куба. Клас RectSolid має поля `C`, `D` – довжина та ширина основи прямокутного паралелепіпеда – та `H` – його висота. Кожний із похідних класів має конструктор із параметрами та перевизначає метод `GetVolume()` базового класу.

Розробити консольну програму, яка використовує принцип поліморфізму під час обчислення об'ємів куба ($V = A^3$) та прямокутного паралелепіпеда ($V = C \cdot D \cdot H$). Для цього програма має:

1. Створити стек (об'єкт класу `System.Collections.Stack`).
2. Створити через посилання на тип `Solid` по одному об'єкту класів `Cube` та `RectSolid` (значення довжини ребра куба, довжини та ширини основи прямокутного паралелепіпеда, його висоти ввести з консолі через параметри конструктора) та додати їх до стека.
3. За допомогою оператора циклу `foreach` для кожного елемента стека викликати метод `GetVolume()`.
4. Вивести значення об'ємів куба та прямокутного паралелепіпеда на консоль.

Завдання 2 (евристичне). Розробити програму із графічним інтерфейсом користувача (WPF) для знаходження середнього арифметичного елементів матриці цілих чисел розміру 2×2 .

Програма має:

1. Для введення елементів матриці використовувати необхідну кількість компонентів TextBox.
2. Дозволяти користувачу використовувати для обчислень значення елементів матриці тільки з інтервалу [-5; 30], інакше повідомляти його про відповідну помилку.
3. Виводити результати роботи програми в компонент Label.
4. Для організації взаємодії з користувачем використовувати панель меню (компонент Menu) з меню "Файл" (пункт "Вийти із програми") та меню "Матриця" (пункти "Обчислити", "Очистити матрицю").

Затверджено на засіданні кафедри "Інформаційних систем".
Протокол № 11 від 01.04.2016 р.

Завідувач кафедри _____ проф. О. Г. Руденко

Екзаменатор _____ доц. Ю. Е. Парфьонов

Критерії оцінювання екзаменаційної роботи

Екзаменаційний білет складається із двох завдань. Кожне завдання оцінюють від 0 до 6 умовних балів, відповідно до такої шкали (табл. 9.1):

Таблиця 9.1

Критерії оцінювання

Умовні бали	Необхідні вимоги для набрання
1	2
6	Завдання виконано в повному обсязі. Програма працює правильно. Інтерфейс та вихідний код програми задовольняють установленим вимогам
5	Завдання виконано. Програма працює правильно, але одну з її функціональних можливостей реалізовано з порушенням вимог, указаних у завданні

1	2
4	Завдання в основному виконано. Програма працює, але дві з її функціональних можливостей реалізовано з порушенням вимог, указаних у завданні
3	Завдання виконано, але не в повному обсязі. Програма працює, але не реалізовано одну з функціональних вимог, указаних у завданні, або три з функціональних можливостей програми реалізовано з порушенням вимог, указаних у завданні
2	Завдання не виконано. Програма запускається, але не реалізовано дві з функціональних вимог, указаних у завданні, або чотири з функціональних можливостей програми реалізовано з порушенням вимог, указаних у завданні
1	Програма запускається та частково відповідає поставленому завданню, але не реалізовано більше від двох з функціональних вимог, указаних у завданні, або більш від чотирьох з функціональних можливостей програми реалізовано з порушенням вимог, указаних у завданні. Програма не запускається або завершується аварійно, але є програмний код, розроблений студентом, який відповідає поставленому завданню
0	Програма відсутня. Програма не містить програмного коду, розробленого студентом. Програма не відповідає поставленому завданню. Програма має явні ознаки несаможитності її розроблення

Невиконання або суттєве порушення студентом будь-якої із загальних вимог до розроблених програм зменшує кількість умовних балів за завдання на 1 (один) бал. Загальні вимоги до програм наведено далі.

Вимоги до інтерфейсу користувача

1. Інтерфейс користувача має відповідати поставленому завданню.
2. Інтерфейс користувача консольної програми має складатися з текстових повідомлень українською, англійською або російською мовою, які належать до введення та виведення основних та допоміжних даних.
3. Елементи графічного інтерфейсу користувача мають супроводжуватися допоміжними текстовими мітками або мати заголовки, які виконують українською, англійською або російською мовою.

Вимоги до вихідного тексту програм

1. Дотримання принципу інкапсуляції щодо рівнів доступу до полів, властивостей та методів класів.

2. Вихідний код кожного із класів програми повинен міститися в окремому файлі.

Підсумкову оцінку за іспит визначають за набраною сумою балів за виконання двох завдань (табл. 9.2):

Таблиця 9.2

Визначення підсумкової оцінки

Набрана сума умовних балів	Підсумкова оцінка за іспит за 40-бальною шкалою
12	40
11	38 – 39
10	36 – 37
9	34 – 35
8	32 – 33
7	30 – 31
6	28 – 29
5	26 – 27
4	24 – 25
3	16 – 23
2	8 – 15
1	1 – 7
0	0

10. Розподіл балів, які отримують студенти

Систему оцінювання рівня сформованості професійних компетентностей студентів денної форми навчання наведено в табл. 10.1.

Система оцінювання рівня сформованості професійних компетентностей

Професійні компетентності	Навчальний тиждень	Години	Методи та форми навчання		Оцінювання рівня сформованості компетентностей		
					Форми контролю	Максимальний бал	
1	2	3	4		5	6	
Змістовий модуль 1. Основи об'єктно-орієнтованої парадигми							
Здатність розробляти найпростіші програми з використанням основних елементів мов C# та Java	1	Ауд.	2	Лекція	Лекція 1. Основні поняття платформ Microsoft .NET та Java SE	Робота на лекції	0,5
			2	Лабораторне заняття	Лабораторна робота 1. Основні поняття платформ Microsoft .NET та Java SE	Активна участь у виконанні лабораторної роботи	0,5
		СРС	4	Підготовка до занять	Пошук, підбір та огляд літературних джерел З'ясування завдання на лабораторну роботу	–	–
	2	Ауд.	2	Лекція	Лекція 2. Основи мов програмування Java та C#	Робота на лекції	0,5
			2	Лабораторне заняття	Лабораторна робота 1. Основні поняття платформ Microsoft .NET та Java SE	Активна участь у виконанні лабораторної роботи Захист лабораторної роботи 1	0,5 3
		СРС	4	Підготовка до занять	Пошук, підбір та огляд літературних джерел Виконання завдань щодо розроблення програм за темою лабораторної роботи	–	–
	3	Ауд.	2	Лекція	Лекція 3. Основи об'єктно-орієнтованого підходу	Робота на лекції	0,5
			2	Лабораторне заняття	Лабораторна робота 2. Розроблення застосувань із використанням базових елементів ООП З'ясування завдання на лабораторну роботу	Активна участь у виконанні лабораторної роботи	0,5

1	2	3	4		5	6		
		СРС	4	Підготовка до занять	Пошук, підбір та огляд літературних джерел	–	–	
					Виконання завдань щодо розроблення програм за темою лабораторної роботи			
Змістовий модуль 2. Основні елементи ООП								
Здатність використовувати основні елементи ООП у процесі розроблення програм	4	Ауд.	2	Лекція	Лекція 4. Основи об'єктно-орієнтованого програмування мовами C# та Java	Робота на лекції	0,5	
			2	Лабораторне заняття	Лабораторна робота 2. Розроблення застосувань із використанням базових елементів ООП	Активна участь у виконанні лабораторної роботи	0,5	
						Захист лабораторної роботи 2	4	
						Контрольна робота 1	2	
			СРС	8	Підготовка до занять	Пошук, підбір та огляд літературних джерел Виконання завдань щодо розроблення програм за темою лабораторної роботи, підготовка до контрольної роботи 1	–	–
	5	Ауд.	2	Лекція	Лекція 5. Агрегація та композиція	Робота на лекції	0,5	
			2	Лабораторне заняття	Лабораторна робота 3. Застосування успадкування та поліморфізму	Активна участь у виконанні лабораторної роботи	0,5	
				СРС	3	Підготовка до занять	Пошук, підбір та огляд літературних джерел Виконання завдань щодо розроблення програм за темою лабораторної роботи	–
	6	Ауд.	2	Лекція	Лекція 6. Успадкування	Робота на лекції	0,5	
			2	Лабораторне заняття	Лабораторна робота 3. Застосування успадкування та поліморфізму	Активна участь у виконанні лабораторної роботи	0,5	

Продовження табл. 10.1

1	2	3	4		5	6		
	СРС	3	Підготовка до занять	Пошук, підбір та огляд літературних джерел	-	-		
				З'ясування завдання на лабораторну роботу, виконання завдань щодо розроблення програм за темою лабораторної роботи				
				Виконання завдань щодо розроблення програм за темою лабораторної роботи, підготовка до експрес-опитування				
	7	Ауд.	2	Лекція	Лекція 7. Реалізація поліморфізму в C# та Java	Робота на лекції	0,5	
			2	Лабораторне заняття	Лабораторна робота 3. Застосування успадкування та поліморфізму	Активна участь у виконанні лабораторної роботи	0,5	
						Захист лабораторної роботи 3	5	
		СРС	2	Підготовка до занять	Пошук, підбір та огляд літературних джерел	-	-	
					Виконання завдань щодо розроблення програм за темою лабораторної роботи			
					Підготовка до контрольної роботи			
	8	Ауд.	2	Лекція	Лекція 8. Абстрактні класи та інтерфейси у Java і C#	Робота на лекції	0,5	
			2	Лабораторне заняття	Лабораторна робота 4. Використання шаблонів проектування	Активна участь у виконанні лабораторної роботи	0,5	
		СРС	2	Підготовка до занять	З'ясування завдання на лабораторну роботу, пошук, підбір та огляд літературних джерел	-	-	
					Виконання завдань щодо розроблення програм за темою лабораторної роботи			
		9	Ауд.	2	Лекція	Лекція 9. Принципи об'єктно-орієнтованого проектування класів	Робота на лекції	0,5
				2	Лабораторне заняття	Лабораторна робота 4. Використання шаблонів проектування	Активна участь у виконанні лабораторної роботи	0,5
Захист лабораторної роботи 4	4							

1	2	3	4		5	6	
		СРС	6	Підготовка до занять	Пошук, підбір та огляд літературних джерел	–	–
					Виконання завдань щодо розроблення програм за темою лабораторної роботи		
	10	Ауд.	2	Лекція	Лекція 10. Розроблення бібліотек класів	Робота на лекції	0,5
						Лабораторне заняття	Лабораторна робота 5. Використання регулярних виразів
		Контрольна робота 2	3				
		СРС	4	Підготовка до занять	Пошук, підбір та огляд літературних джерел З'ясування завдання на лабораторну роботу, підготовка до контрольної роботи 2	–	–
Змістовий модуль 3. Оброблення винятків і бібліотеки класів							
Здатність обробляти винятки, використовувати регулярні вирази, колекції	11	Ауд.	2	Лекція	Лекція 11. Оброблення винятків	Робота на лекції	0,5
						Лабораторне заняття	Лабораторна робота 5. Використання регулярних виразів
		Захист лабораторної роботи 5	4				
		СРС	6	Підготовка до занять	Пошук, підбір та огляд літературних джерел Виконання завдань щодо розроблення програм за темою лабораторної роботи	–	–
	12	Ауд.	2	Лекція	Лекція 12. Регулярні вирази	Робота на лекції	0,5
						Лабораторне заняття	Лабораторна робота 6. Використання основних бібліотек Microsoft .NET Framework та Java SE
		СРС	4	Підготовка до занять	Пошук, підбір та огляд літературних джерел Виконання завдань щодо розроблення програм за темою лабораторної роботи		

1	2	3	4		5	6	
	13	Ауд.	2	Лекція	Лекція 13. Колекції	Робота на лекції	0,5
			2	Лабораторне заняття	Лабораторна робота 6. Використання основних бібліотек Microsoft .NET Framework та Java SE	Активна участь у виконанні лабораторної роботи	0,5
						Захист лабораторної роботи 6	5
						Контрольна робота 3	3
		СРС	4	Підготовка до занять	Пошук, підбір та огляд літературних джерел Виконання завдань щодо розроблення програм за темою лабораторної роботи, підготовка до контрольної роботи 3	–	–
Здатність використовувати систему введення-виведення даних	14	Ауд.	2	Лекція	Лекція 14. Введення-виведення даних на платформі Microsoft .NET та Java SE	Робота на лекції	0,5
			2	Лабораторне заняття	Лабораторна робота 7. Основи JavaFX	Активна участь у виконанні лабораторної роботи	0,5
		СРС	4	Підготовка до занять	Пошук, підбір та огляд літературних джерел Виконання завдань щодо розроблення програм за темою лабораторної роботи	–	–
Здатність розробляти прості графічні інтерфейси користувача на платформі Java SE	15	Ауд.	2	Лекція	Лекція 15. Основи розроблення графічних інтерфейсів користувача	Робота на лекції	0,5
			2	Лабораторне заняття	Лабораторна робота 7. Основи JavaFX	Активна участь у виконанні лабораторної роботи	0,5
						Захист лабораторної роботи 7	4
		СРС	4	Підготовка до занять	Пошук, підбір та огляд літературних джерел Виконання завдань щодо розроблення програм за темою лабораторної роботи	–	–

Закінчення табл. 10.1

1	2	3	4		5	6	
	16	Ауд.	2	Лекція	Лекція 16. Основи JavaFX	Робота на лекції	0,5
			2	Лабораторне заняття	Лабораторна робота 8. Використання елементів управління JavaFX	Активна участь у виконанні лабораторної роботи	0,5
						Контрольна робота 4	2
		CPC	2	Підготовка до занять	Пошук, підбір та огляд літературних джерел	–	–
					Виконання завдань щодо розроблення програм за темою лабораторної роботи, підготовка до контрольної роботи 4		
		Здатність розробляти прості графічні інтерфейси користувача на платформі Java SE	17	Ауд.	2	Лекція	Лекція 17. Використання елементів управління JavaFX
2	Лабораторне заняття				Лабораторна робота 8. Використання елементів управління JavaFX	Активна участь у виконанні лабораторної роботи	0,5
						Захист лабораторної роботи 8	4
CPC	8			Підготовка до занять	Пошук, підбір та огляд літературних джерел	с	–
		Виконання завдань щодо розроблення програм за темою лабораторної роботи					
	Сесія	CPC	1	Передекзаменаційна консультація	Вирішення практичних завдань на різні теми, що входять до підсумкового контролю	Підсумковий контроль	40
			6	Підготовка до екзамену	Повторення матеріалів змістових модулів	–	–
			3	Екзамен	Виконання завдань екзаменаційного білета	–	–
Усього годин			150	Загальна максимальна кількість балів із дисципліни			100

Розподіл балів у межах змістових модулів наведено в табл. 10.2.

Розподіл балів за темами

Поточне оцінювання та самостійна робота											Підсумковий екзамен	Сума
ЗМ 1			ЗМ 2				ЗМ 3		ЗМ 4		40	100
T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11		
1	4	1	7	9	5	4	5	11	8	5		

Примітка. T1 – T11 – теми змістових модулів.

Максимальну кількість балів, яку може накопичити студент протягом тижня за формами та методами навчання, наведено в табл. 10.3.

Розподіл балів за тижнями

Теми змістових модулів			Лекційні заняття	Лабораторні заняття	Захист лабораторних робіт	Поточні КР	Усього
1			2	3	4	5	6
Змістовий модуль 1	Тема 1	1 тиждень	0,5	0,5	–	–	1
	Тема 2	2 тиждень	0,5	0,5	3	–	4
	Тема 3	3 тиждень	0,5	0,5	–	–	1
Змістовий модуль 2	Тема 4	4 тиждень	0,5	0,5	4	2	7
	Тема 5	5 тиждень	0,5	0,5	–	–	1
		6 тиждень	0,5	0,5	–	–	1
		7 тиждень	0,5	0,5	5	–	6
		8 тиждень	0,5	0,5	–	–	1
	Тема 6	9 тиждень	0,5	0,5	4	–	5
Тема 7	10 тиждень	0,5	0,5	–	3	4	

1			2	3	4	5	6
Змістовий модуль 3	Тема 8	11 тиждень	0,5	0,5	4	–	5
	Тема 9	12 тиждень	0,5	0,5	–	–	1
		13 тиждень	0,5	0,5	5	3	9
		14 тиждень	0,5	0,5	–	–	1
Змістовий модуль 4	Тема 10	15 тиждень	0,5	0,5	4	–	5
		16 тиждень	0,5	0,5	–	2	3
	Тема 11	17 тиждень	0,5	0,5	4	–	5
Усього			8,5	8,5	33	10	60

Підсумкову оцінку з навчальної дисципліни визначають, відповідно до Тимчасового положення "Про порядок оцінювання результатів навчання студентів за накопичувальною бально-рейтинговою системою" ХНЕУ ім. С. Кузнеця (табл. 10.4).

Таблиця 10.4

Шкала оцінювання: національна та ЄКТС

Сума балів за всі види навчальної діяльності	Оцінка ЄКТС	Оцінка за національною шкалою	
		для екзамену, курсового проекту (роботи), практики	для заліку
90 – 100	A	відмінно	зараховано
82 – 89	B	добре	
74 – 81	C		
64 – 73	D		
60 – 63	E		
35 – 59	FX	незадовільно	не зараховано
1 – 34	F		

Оцінки за цією шкалою заносять до відомостей обліку успішності, індивідуального навчального плану студента та іншої академічної документації.

11. Рекомендована література

11.1. Основна

1. Объектно-ориентированный анализ и проектирование с примерами приложений / [Г. Буч, Р. Максимчук, М. Энгл и др.] ; [пер. с англ.] – Москва : ИД "Вильямс", 2008. – 720 с.
2. Троелсен Э. Язык программирования C# 2010 и платформа .NET 4.0 / Э. Троелсен ; [пер. с англ.]. – Москва : ИД "Вильямс", 2011. – 1392 с.
3. Шилдт Г. Java. Руководство для начинающих / Г. Шилдт ; [пер. с англ.]. – Москва : ИД "Вильямс", 2012. – 624 с.
4. Pro JavaFX 8: A Definitive Guide to Building Desktop, Mobile, and Embedded Java Clients / J. Vos, W. Gao, S. Chin [et al.]. – New York : Apress, 2014. – 588 p.
5. Wu. C. Th. An introduction to object-oriented programming with Java / C. Th. Wu. – New York : McGraw-Hill, 2010. – 987 p.

11.2. Додаткова

6. Блинов И. Н. Java. Методы программирования : учеб.-метод. пособ. / И. Н. Блинов, В. С. Романчик. – Минск : Изд-во "Четыре четверти", 2013. – 768 с.
7. Леоненков А. Самоучитель UML 2 / А. Леоненков. – Санкт-Петербург : BHV, 2007. – 576 с.
8. Michaelis M. Essential C# 6.0 / M. Michaelis, E. Lippert. – Boston : Addison-Wesley, 2016. – 1004 p.
9. Sharan K. Learn JavaFX 8 / K. Sharan. – New York : Apress, 2015. – 1200 p.

11.3. Інформаційні ресурси

10. Программирование на Java для детей, родителей, дедушек и бабушек [Электронный ресурс]. – Режим доступа : http://myflex.org/books/java4kids/JavaKid8x11_ru.pdf.

11. Уроки Java для начинающих [Электронный ресурс]. – Режим доступа : <http://cybern.ru/category/java/begin-java>.

12. Учебник по JavaFX 8 [Электронный ресурс]. – Режим доступа : <http://code.makery.ch/library/javafx-8-tutorial/ru>.

13. Programming Tutorials and Source Code Examples [Electronic resource]. – Access mode : <http://www.java2s.com>.

14. The Java Tutorials [Electronic resource]. – Access mode : <http://download.oracle.com/javase/tutorial>.

11.4. Методичне забезпечення

15. Парфьонов Ю. Е. Методичні рекомендації до виконання лабораторних робіт з навчальної дисципліни "Основи об'єктно-орієнтованого програмування" [Електронний ресурс] / Ю. Е. Парфьонов. – Режим доступу : <http://www.ikt.hneu.edu.ua/course/view.php?id=879>.

16. Парфьонов Ю. Е. Основи об'єктно-орієнтованого програмування : презентації лекцій [Електронний ресурс] / Ю. Е. Парфьонов. – Режим доступу : <http://www.ikt.hneu.edu.ua/course/view.php?id=879>.

Додатки

Додаток А

Таблиця А.1

Структура складових професійних компетентностей із навчальної дисципліни "Основи об'єктно-орієнтованого програмування" за Національною рамкою кваліфікацій України

Складові компетентності, яку формують у межах теми	Мінімальний досвід	Знання	Уміння	Комунікації	Автономність і відповідальність
1	2	3	4	5	6
Тема 1. Основи Microsoft .NET та Java SE					
Уявляти особливості розроблення застосунків для платформ Microsoft .NET та Java SE	Поняття про класичні етапи розроблення програм	Знання особливостей розроблення програм для платформ Microsoft .NET та Java SE	Використовувати програмне забезпечення, необхідне для розроблення застосунків для платформ Microsoft .NET та Java SE	Описувати прийняті рішення з використанням усного та письмового мовлення, символічної мови та презентувати їхні результати	Самостійно приймати рішення та відповідати за їхню коректність та терміни реалізації
Тема 2. Основи об'єктно-орієнтованої мови програмування					
Розробляти найпростіші програми мовою програмування Java	Загальні відомості про платформи Microsoft .NET, Java SE та мови Java і C#	Знання системи типізації, операцій, операторів, структури програми, базових елементів стандартної бібліотеки класів, синтаксису та особливостей використання масивів, синтаксису щодо сигнатури та виклику методів	Розробляти лінійні програми	Описувати прийняті рішення з використанням усного та письмового мовлення, символічної мови та презентувати їхні результати	Самостійно приймати рішення та відповідати за їхню коректність та терміни реалізації

1	2	3	4	5	6
Тема 3. Поняття об'єктно-орієнтованого аналізу, проектування та програмування					
Здатність до об'єктно-орієнтованого мислення	Основні поняття об'єктно-орієнтованої парадигми	Знання змісту основних етапів об'єктно-орієнтованої технології	Використовувати базові елементи об'єктно-орієнтованого аналізу та проектування, зокрема із застосуванням CASE-засобів	Описувати прийняті рішення з використанням усного та письмового мовлення, символічної мови та презентувати їхні результати	Самостійно приймати рішення та відповідати за їхню коректність та терміни реалізації
Тема 4. Абстрагування даних та інкапсуляція					
Здатність до об'єктно-орієнтованого мислення	Поняття про класи та об'єкти, виділення та звільнення пам'яті	Знання елементів класу, рівнів доступу до них, основ використання об'єктів, послідовності створення та ініціалізації об'єкту, особливостей застосування конструкторів та статичних членів класу	Розробляти прості об'єктно-орієнтовані програми з використанням конструкторів та статичних членів класу	Описувати прийняті рішення з використанням усного та письмового мовлення, символічної мови та презентувати їхні результати	Самостійно приймати рішення та відповідати за їхню коректність та терміни реалізації
Тема 5. Повторне використання класів					
Здатність використовувати відношення агрегації, композиції та спадкування у програмах	Поняття про повторне використання програмного коду	Знання варіантів повторного використання класів та відповідних відношень між класами та об'єктами	Розробляти прості об'єктно-орієнтовані програми з використанням відношень агрегації, композиції та успадкування	Описувати прийняті рішення з використанням усного та письмового мовлення, символічної мови та презентувати їхні результати	Самостійно приймати рішення та відповідати за їхню коректність та терміни реалізації

1	2	3	4	5	6
Тема 6. Принципи об'єктно-орієнтованого проектування класів					
Здатність використовувати принципи SOLID та базові шаблони проектування	Поняття про об'єктно-орієнтоване проектування	Знання принципів SOLID та основних шаблонів проектування	Реалізовувати принципи SOLID та деякі шаблони проектування у процесі розроблення об'єктно-орієнтованих програм	Описувати прийняті рішення з використанням усного та письмового мовлення, символічної мови та презентувати їхні результати	Самостійно приймати рішення та відповідати за їхню коректність та терміни реалізації
Тема 7. Бібліотеки класів					
Здатність створювати та використовувати бібліотеки класів	Поняття про програмні бібліотеки	Знання подання бібліотек класів на платформах Microsoft .NET та Java SE	Створювати бібліотеки класів на платформах Microsoft .NET та Java SE	Описувати прийняті рішення з використанням усного та письмового мовлення, символічної мови та презентувати їхні результати	Самостійно приймати рішення та відповідати за їхню коректність та терміни реалізації
Тема 8. Оброблення виняткових ситуацій					
Здатність використовувати механізми оброблення винятків	Види помилок у програмах	Знання структури основних класів винятків стандартної бібліотеки Microsoft .NET та Java SE, синтаксису оброблення винятків	Розробляти об'єктно-орієнтовані програми з використанням механізмів оброблення винятків	Описувати прийняті рішення з використанням усного та письмового мовлення, символічної мови та презентувати їхні результати	Самостійно приймати рішення та відповідати за їхню коректність та терміни реалізації
Тема 9. Стандартні бібліотеки класів середовищ розробника програм					
Уявляти особливості різних видів колекцій, використовувати потоки введення та виведення	Поняття про колекцію об'єктів, потоки введення та виведення	Знання основних структур даних стандартної бібліотеки колекцій Microsoft .NET та Java SE, основних абстракцій стандартної бібліотеки введення та виведення	Вибирати потрібний вид колекцій, потрібний вид потоку введення або виведення та використовувати їх у розроблюваних програмах	Описувати прийняті рішення з використанням усного та письмового мовлення, символічної мови та презентувати їхні результати	Самостійно приймати рішення та відповідати за їхню коректність та терміни реалізації

1	2	3	4	5	6
Тема 10. Основи розроблення графічних інтерфейсів користувача					
Здатність до використання сучасних технологій розроблення графічних інтерфейсів користувача	Поняття про графічні інтерфейси користувача	Знання послідовності оброблення подій, основних елементів XML та основ використання FXML для опису графічного інтерфейсу користувача	Використовувати FXML у процесі розроблення графічного інтерфейсу користувача	Описувати прийняті рішення з використанням усного та письмового мовлення, символічної мови та презентувати їхні результати	Самостійно приймати рішення та відповідати за їхню коректність та терміни реалізації
Тема 11. Розроблення графічних інтерфейсів користувача на платформі Java SE					
Здатність до розроблення сучасних графічних інтерфейсів користувача	Поняття про JavaFX	Знання основних елементів JavaFX	Використовувати технологію FXML та відповідні інструментальні засоби у процесі розроблення простих графічних інтерфейсів користувача	Описувати прийняті рішення з використанням усного та письмового мовлення, символічної мови та презентувати їхні результати	Самостійно приймати рішення та відповідати за їхню коректність та терміни реалізації

Зміст

Вступ.....	3
1. Опис навчальної дисципліни	4
2. Мета та завдання навчальної дисципліни	4
3. Програма навчальної дисципліни	6
4. Структура навчальної дисципліни.....	9
5. Теми лабораторних занять.....	11
5.1. Приклади типових лабораторних завдань до тем.....	13
6. Самостійна робота студентів.....	16
6.1. Контрольні запитання для самодіагностики	19
7. Індивідуально-консультативна робота.....	22
8. Методи навчання	23
9. Методи контролю	26
10. Розподіл балів, які отримують студенти	36
11. Рекомендована література.....	45
11.1. Основна	45
11.2. Додаткова	45
11.3. Інформаційні ресурси.....	45
11.4. Методичне забезпечення	46
Додатки.....	47

НАВЧАЛЬНЕ ВИДАННЯ

ОСНОВИ ОБ'ЄКТНО-ОРІЄНТОВАНОГО ПРОГРАМУВАННЯ

**Робоча програма
для студентів спеціальності
122 "Комп'ютерні науки та інформаційні технології"
першого (бакалаврського) рівня**

Самостійне електронне текстове мережеве видання

Укладачі: **Парфьонов** Юрій Едуардович
Щербаков Олександр Всеволодович
Федорченко Володимир Миколайович

Відповідальний за видання *О. Г. Руденко*

Редактор *О. Г. Доценко*

Коректор *О. Г. Доценко*

План 2017 р. Поз. № 98 ЕВ. Обсяг 52 с.

Видавець і виготовлювач – ХНЕУ ім. С. Кузнеця, 61166, м. Харків, просп. Науки, 9-А

*Свідоцтво про внесення суб'єкта видавничої справи до Державного реєстру
ДК № 4853 від 20.02.2015 р.*