

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ СЕМЕНА КУЗНЕЦЯ

"ЗАТВЕРДЖУЮ"

Заступник керівника
(проректор з науково-педагогічної роботи)


М. В. Афанасьєв



ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ
робоча програма навчальної дисципліни

Галузь знань	12 Інформаційні технології
Спеціальність	121 "Інженерія програмного забезпечення"
Освітній рівень	перший (бакалаврський) рівень
Освітня програма	Інженерія програмного забезпечення

Вид дисципліни	базова
Мова викладання, навчання та оцінювання	українська

Завідувач кафедри інформаційних систем

 I. O. Ушакова

Харків
ХНЕУ ім. С. Кузнеця

ЗАТВЕРДЖЕНО

на засіданні кафедри інформаційних систем
Протокол № 1 від 30.08.2019 р.

Розробник:

Парфьонов Юрій Едуардович, к.т.н., доцент кафедри інформаційних систем

Лист оновлення та перезатвердження робочої програми навчальної дисципліни

Навчальний рік	Дата засідання кафедри – розробника РПНД	Номер протоколу	Підпис завідувача кафедри

1. Вступ

Анотація навчальної дисципліни: У наш час найбільш поширеним методом боротьби зі складністю програмного забезпечення є об'єктно-орієнтований підхід до його розроблення. Це потребує від відповідних фахівців чіткого уявлення концепцій об'єктно-орієнтованого програмування, що дає можливість їхнього практичного використання у процесі розроблення застосунків будь-якою мовою програмування.

Навчальна дисципліна "Об'єктно-орієнтоване програмування" є базовою навчальною дисципліною, яку вивчають, згідно з навчальним планом підготовки фахівців першого (бакалаврського) рівня за спеціальністю 121 "Інженерія програмного забезпечення".

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

Мета навчальної дисципліни: формування компетентностей з використання об'єктно-орієнтованого підходу та сучасних мов програмування, необхідних для розроблення відповідних програмних застосунків.

Курс	2		
Семестр	1, 2		
Кількість кредитів ECTS	10		
Аудиторні навчальні заняття		1-й семестр	2-й семестр
	лекції	32	32
	лабораторні	32	32
Самостійна робота		86	86
Форма підсумкового контролю		залік	іспит

Структурно-логічна схема вивчення навчальної дисципліни:

Попередні дисципліни	Наступні дисципліни
Програмування	Програмування Інтернет
Основи алгоритмізації	Веб-технології та веб-дизайн

2. Компетентності та результати навчання за дисципліною:

Компетентності	Результати навчання
Здатність до об'єктно-орієнтованого мислення	Після успішного закінчення навчання студент має продемонструвати здатність застосовувати об'єктно-орієнтований підхід під час проектування програмних систем
Знання об'єктно-орієнтованої мови програмування	Після успішного закінчення навчання студент має продемонструвати знання основних понять об'єктно-орієнтованого аналізу, проектування та програмування; принципів об'єктно-орієнтованого підходу; понять класу та об'єкту; типів відношень між класами; особливостей реалізації основних концепцій об'єктно-орієнтованого програмування в певних мовах програмування; технологічних засобів та способів їх використання в процесі виконання професійної діяльності та розв'язання професійних завдань початкового рівня складності.

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. Форматування рядків.

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, повертання масиву з методу, виклик методу.

2.4. Перевантаження методів класу. Принципи перевантаження операцій.

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. Моделювання програмного забезпечення. Основні поняття візуального моделювання. Основи мови UML. UML-діаграми. UML-діаграми класів. Відношення на діаграмі класів. CASE-засоби.

Змістовий модуль 2. Технологія ООП

Тема 4. Абстрагування даних та інкапсуляція

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

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

4.3. 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 кода

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. Особливості реалізації об'єктно-орієнтованого підходу в Python

Синтаксис визначення класів. Атрибути та методи. Об'єкти. Доступ до елементів класів. Успадкування. Множинне успадкування.

Тема 7. Принципи об'єктно-орієнтованого проектування SOLID

7.1. Система принципів SOLID. Принцип єдиної відповідальності. Принцип відкриття-закриття. Принцип інверсії залежностей. Принцип заміщення Барбери Лісков. Принцип ізоляції інтерфейсу.

Тема 8. Вступ до шаблонів проектування

8.1. Загальні відомості про шаблони проектування. Переваги та недоліки використання шаблонів проектування. Елементи шаблону проектування. Класифікація шаблонів проектування GoF. Застосування основних шаблонів проектування GoF. Огляд шаблонів розподілу відповідальностей GRASP.

Тема 9. Бібліотеки класів

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

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

Змістовий модуль 3. Оброблення винятків і бібліотеки класів

Тема 10. Оброблення виняткових ситуацій

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

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

Тема 11. Стандартні бібліотеки класів середовищ розробника програм

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

11.2. Використання лямбда-виразів

11.3. Використання LINQ та Stream API

11.4. Особливості реалізації рядкового типу даних на платформах Microsoft .NET та Java SE. Класи стандартних бібліотек Microsoft .NET та Java SE для подання рядків та особливості їхнього використання.

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

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

11.7. Збереження та відновлення стану об'єктів на платформах Microsoft .NET та Java SE. Серіалізація та десеріалізація. «Граф» об'єктів при серіалізації. Створення класів, об'єкти яких можливо серіалізувати. Процеси серіалізації та десеріалізації. Формати серіалізації.

Змістовий модуль 4. Об'єктно-орієнтоване програмування застосунків із графічним інтерфейсом користувача

Тема 12. Основи розроблення графічних інтерфейсів користувача

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

12.2. Огляд сучасних технологій розроблення застосунків із графічним інтерфейсом користувача на платформах Microsoft .NET, Java SE та в мові Python. Основи мови XML. Структура XML-документа. Простори імен XML. XML-схеми.

12.3. Загальні відомості про мови FXML та XAML. Використання FXML та XAML для опису графічного інтерфейсу користувача.

Тема 13. Розроблення графічних інтерфейсів користувача на платформі Java SE.

13.1. Загальна структура застосунку JavaFX. Форми. Події рівня форми.

13.2. Використання форм та базових елементів управління JavaFX. Оброблення подій від миші, клавіатури та елементів управління.

13.3. Управління розташуванням візуальних компонентів. Використання основних елементів управління JavaFX.

13.4. Прив'язки до даних у JavaFX. Стили. Використання графічних можливостей JavaFX.

Тема 14. Розроблення графічних інтерфейсів користувача мовою Python.

14.1. Вступ до PyQt. Управління розташуванням візуальних компонентів

14.2. Віджети. Слоти та сигнали.

14.3. Стили. Графічні можливості PyQt.

Тема 15. Розроблення графічних інтерфейсів користувача на платформі Microsoft .NET

15.1. Вступ до технології WPF. Основні класи WPF. Загальна структура застосунку WPF.

15.2. Управління розташуванням візуальних компонентів у WPF.

15.3. Маршрутизація подій у WPF. Використання елементів управління WPF, основи використання механізму прив'язки до даних.

15.4. Ресурси, стилі та тригери WPF.

Теми лабораторних занять

1. Основи використання мов програмування C# та Java.

2. Проектування класів з використанням мови UML.

3. Розроблення застосунків з використанням базових елементів ООП.

4. Застосування спадкування та поліморфізму

5. Розроблення об'єктно-орієнтованих застосунків мовою Python.

6. Використання шаблонів проектування та бібліотек класів.

7. Використання колекцій.

8. Використання лямбда-виразів, LINQ та Stream API.

9. Використання регулярних виразів.

10. Використання введення-виведення даних.

11. Основи JavaFX.

12. Використання елементів управління та графічних можливостей JavaFX.

13. Використання головних елементів PyQt.

14. Використання головних елементів WPF.

4. Порядок оцінювання результатів навчання

Система оцінювання сформованих компетентностей у студентів враховує види занять, які згідно з програмою навчальної дисципліни передбачають лекційні, лабораторні заняття, а також виконання самостійної роботи. Оцінювання сформованих компетентностей у студентів здійснюється за накопичувальною 100-бальною системою. Відповідно до Тимчасового положення "Про порядок оцінювання результатів навчання студентів за накопичувальною бально-рейтинговою системою" ХНЕУ ім. С. Кузнеця, контрольні заходи включають:

поточний контроль, що здійснюється протягом семестру під час проведення лабораторних занять, і оцінюється сумою набраних балів (максимальна сума – 60 балів; мінімальна сума, що дозволяє студенту скласти іспит, – 35 балів);

модульний контроль, що проводиться у формі контрольних робіт і має на меті інтегровану оцінку результатів навчання студента після вивчення матеріалу споріднених змістових модулів;

підсумковий контроль, що проводиться у формі семестрового екзамену, відповідно до графіку навчального процесу.

Порядок проведення поточного оцінювання знань студентів. Оцінювання знань студента під час лабораторних занять та виконання індивідуальних завдань проводиться за такими критеріями:

розуміння, ступінь засвоєння основних положень об'єктно-орієнтованого підходу; ступінь засвоєння фактичного матеріалу навчальної дисципліни; ознайомлення з рекомендованою літературою, а також із сучасною літературою щодо об'єктно-орієнтованої парадигми та її реалізації в певних мовах програмування; вміння поєднувати теорію з практикою при розробленні програм у процесі виконання індивідуальних завдань та завдань, винесених на розгляд в аудиторії; логіка, структура, стиль викладу матеріалу в письмових роботах і при виступах в аудиторії, вміння обґрунтовувати свою позицію, здійснювати узагальнення інформації та робити висновки; здатність проводити критичну та незалежну оцінку проблемних питань щодо розроблення програм з використанням об'єктно-орієнтованого підходу; вміння пояснювати альтернативні погляди та наявність власної точки зору, позиції на певне проблемне питання; застосування аналітичних підходів; якість і чіткість викладення міркувань; логіка, структуризація та обґрунтованість висновків щодо конкретної проблеми; самостійність виконання роботи; грамотність подачі матеріалу; використання методів порівняння, узагальнення понять та явищ; оформлення роботи.

Загальними критеріями, за якими здійснюється оцінювання позааудиторної самостійної роботи студентів, є: глибина і міцність знань, рівень мислення, вміння систематизувати знання за окремими темами, вміння робити обґрунтовані висновки, володіння категорійним апаратом, навички і прийоми виконання практичних завдань, вміння знаходити необхідну інформацію, здійснювати її систематизацію та обробку, самореалізація на практичних та семінарських заняттях.

Підсумковий контроль знань та компетентностей студентів з навчальної дисципліни здійснюється на підставі проведення семестрового екзамену, завданням якого є перевірка розуміння студентом програмного матеріалу в цілому, логіки та взаємозв'язків між окремими розділами, здатності творчого використання накопичених знань, вміння формулювати своє ставлення до певної проблеми навчальної дисципліни тощо.

Екзаменаційний білет охоплює програму дисципліни і передбачає визначення рівня знань та ступеня опанування студентами компетентностей.

Кожен екзаменаційний білет складається із 2 практичних завдань (два евристичних завдання), які передбачають вирішення типових професійних завдань фахівця на робочому місці та дозволяють діагностувати рівень теоретичної підготовки студента і рівень його компетентності з навчальної дисципліни.

Результат семестрового екзамену оцінюється в балах (максимальна кількість – 40 балів, мінімальна кількість, що зараховується, – 25 балів) і проставляється у відповідній графі екзаменаційної "Відомості обліку успішності".

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

Підсумкова оцінка з навчальної дисципліни розраховується з урахуванням балів, отриманих під час екзамену, та балів, отриманих під час поточного контролю за накопичувальною системою. Сумарний результат у балах за семестр складає: "60 і

більше балів – зараховано", "59 і менше балів – не зараховано" та заноситься у залікову "Відомість обліку успішності" навчальної дисципліни.

Розподіл балів за тижнями

Теми змістових модулів			Лекційні заняття	Лабораторні заняття	Захист лабораторних робіт	Поточні КР	Усього
1			2	3	4	5	6
Семестр 1							
Змістовий модуль 1	Тема 1	1 тиждень	0,5	0,5	–	–	1
	Тема 2	2 тиждень	0,5	0,5	–	–	1
	Тема 3	3 тиждень	0,5	0,5	7	–	8
		4 тиждень	0,5	0,5	–	–	1
Змістовий модуль 2	Тема 4	5 тиждень	0,5	0,5	5		6
	Тема 5	6 тиждень	0,5	0,5	–	–	1
		7 тиждень	0,5	0,5	8	–	9
		8 тиждень	0,5	0,5	–	–	1
		9 тиждень	0,5	0,5	–	10	11
	Тема 6	10 тиждень	0,5	0,5	12	–	13
	Тема 7	11 тиждень	0,5	0,5	–	–	1
	Тема 8	12 тиждень	0,5	0,5	10	–	11
Тема 9	13 тиждень	0,5	0,5	–	–	1	
Змістовий модуль 3	Тема 10	14 тиждень	0,5	0,5	12	–	13
	Тема 11	15 тиждень	0,5	0,5		10	11
		16 тиждень	0,5	0,5	10	–	11
Усього			8	8	64	20	100
Семестр 2							
Змістовий модуль 3	Тема 11	1 тиждень	0,5	0,5	–	–	0,5
		2 тиждень	0,5	0,5	–	–	0,5
		3 тиждень	0,5	0,5	6	–	7,5
		4 тиждень	0,5	0,5	–	–	0,5
		5 тиждень	0,5	0,5	4	–	5,5
		6 тиждень	0,5	0,5	–	–	0,5

1		2	3	4	5	6	
Змістовий модуль 4	Тема 12	7 тиждень	0,5	0,5	3	–	4,5
	Тема 13	8 тиждень	0,5	0,5	–	–	0,5
		9 тиждень	0,5	0,5	4	5	9,5
		10 тиждень	0,5	0,5	–	–	0,5
		11 тиждень	0,5	0,5	–	–	0,5
	Тема 14	12 тиждень	0,5	0,5	6	–	8,5
		13 тиждень	0,5	0,5	–	–	0,5
		14 тиждень	0,5	0,5	6	–	7,5
	Тема 15	15 тиждень	0,5	0,5	–	5	5,5
		16 тиждень	0,5	0,5	5	–	7,5
Іспит						40	
Усього		8	8	34	10	100	

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	Задовільно	
60 – 63	E		
35 – 59	FX	Незадовільно	не зараховано
1 – 34	F		

5. Рекомендована література

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. , Джепикс Ф. Язык программирования C# 7 и платформы .NET и .NET Core / Э. Троелсен, Ф. Джепикс; [пер. с англ.]. – Москва : ИД "Вильямс", 2018. – 1328 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.
6. J. Burton Browning, M. Alchin. Pro Python 3 – Apress, 2019. – 458 p.

Додаткова

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

Інформаційні ресурси в Інтернеті

11. Об'єктно-орієнтоване програмування (121) [Електронний ресурс]. – Режим доступа : <https://pns.hneu.edu.ua/course/view.php?id=5377>.
12. Уроки Java для начинающих [Электронный ресурс]. – Режим доступа : <http://cybern.ru/category/java/begin-java>.
13. Учебник по JavaFX 8 [Электронный ресурс]. – Режим доступа : <http://code.makery.ch/library/javafx-8-tutorial/ru>.
14. Programming Tutorials and Source Code Examples [Electronic resource]. – Access mode : <http://www.java2s.com>.
15. JDK 12 Documentation [Electronic resource]. – Access mode: <https://docs.oracle.com/en/java/javase/12/>.
16. Java Tutorial [Electronic resource]. – Access mode: <https://www.w3schools.com/java/default.asp>
17. About PyQt [Electronic resource]. – Access mode: <https://wiki.python.org/moin/PyQt>
18. PyQt5 tutorial [Electronic resource]. – Access mode: <https://build-system.fman.io/pyqt5-tutorial>