

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ

ДОКТОР ФІЛОСОФІЇ

Кафедра інформаційних технологій



«ЗАТВЕРДЖУЮ»

Проректор з навчально-методичної
роботи

/ Г.М. Тонкачєєв /

_____ 2020 року

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

"Дисципліни спецкурсу за темою наукового дослідження".

**"Сучасні методології проектування
та розробка програмного забезпечення"**

(назва навчальної дисципліни)

шифр	назва спеціальності
122	Комп'ютерні науки
	назва освітньо-наукової програми
	Комп'ютерні науки

Розробник(и):

Київська К.І., к.т.н., доцент

(прізвище та ініціали, науковий ступінь, звання)

Цюцюра М.І., к.т.н., доцент

(прізвище та ініціали, науковий ступінь, звання)

(підпис)

(підпис)

Робоча програма затверджена на засіданні кафедри інформаційних технологій

протокол № 10 від "11" 02 2020 року

Завідувача кафедри

(підпис)

(Цюцюра С.В.)

(прізвище та ініціали)

Схвалено навчально-методичною радою КНУБА

Протокол № 6 від "25" 02 2020 року

Голова НМР

(підпис)

(Тонкачєєв Г.М.)

(прізвище та ініціали)

ВИТЯГ З НАВЧАЛЬНОГО ПЛАНУ 2019-2021 рр.

шифр	Доктор філософії ОНП	Форма навчання:										Форма контролю	Семестр	Відмітка про погодження
	Назва спеціальності	Кредитів на сем.	Обсяг годин					Кількість індивідуальних робіт						
			Всього	аудиторних										
				Разом	у тому числі			КП	КР	РГ	роб			
Л	Лр	Пз												
122	Комп'ютерні науки	5	150	50			50				1	Зал	3	

шифр	Доктор філософії ОНП	Форма навчання:										Форма контролю	Семестр	Відмітка про погодження
	Назва спеціальності	Кредитів на сем.	Обсяг годин					Кількість індивідуальних робіт						
			Всього	аудиторних										
				Разом	у тому числі			КП	КР	РГ	роб			
Л	Лр	Пз												
122	Комп'ютерні науки	5	150	50			50				1	Зал	3	

Мета та завдання навчальної дисципліни

Метою дисципліни є засвоєння сучасних методології проектування, розуміння процесу побудови програмного забезпечення, оволодіння методологічними та методичними основами побудови архітектури програмних застосунків в галузі інформаційних технологій.

Компетенції аспірантів, що формуються в результаті засвоєння дисципліни

Інтегральна компетентність (ІК)	Здатність розв'язувати комплексні задачі та проблеми в області ІСТ, здійснювати в цій галузі дослідницько-інноваційну діяльність, що передбачає глибоке осмислення наявних знань, створення нових цілісних знань, оволодіння методологією наукової та педагогічної діяльності, проведення власного наукового дослідження, результати якого мають наукову новизну, теоретичне та практичне значення, практичне впровадження отриманих результатів.
Загальні компетентності (ЗК)	ЗК06. Здатність використовувати поглиблені теоретичні та практичні знання, частина яких знаходиться на передовому рубежі даної науки.
Фахові компетентності спеціальності (ФК)	ФК01. Здатність самостійно здобувати за допомогою інформаційних технологій і використовувати в практичній діяльності нові знання і вміння, в тому числі в нових галузях знань, безпосередньо не пов'язаних зі сферою діяльності, розширювати і поглиблювати свій науковий світогляд. ФК04. Здатність оформляти, представляти і доповідати результати виконаної роботи. ФК05. Здатність розробляти методики, плани і програми проведення наукових досліджень і розробок, готувати завдання для виконавців, організувати проведення експериментів і випробувань, аналізувати і узагальнювати їх результати.
Програмні результати навчання	
За загальними та загально-професійними компетентностями (ПРН)	ПРН03. Здатність орієнтуватися в постановці завдання і визначати, яким чином слід шукати засоби її рішення. ПРН10. Вміння готувати презентації, оформляти результати досліджень у вигляді статей і доповідей

	на науково-технічних конференціях.
--	------------------------------------

	ПРН11. Володіння сучасними інформаційними технологіями.
--	--

Програма навчальної дисципліни

Модуль 1. Патерни проектування

Змістовий модуль 1. Введення до патернів проектування

Тема 1. ПОВТОРЕННЯ ЗАГАЛЬНИХ ПРИНЦИПІВ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Проектування класів. Метод функціональної декомпозиції. Поняття класу, властивості, методи, модифікатори доступу до даних класів. Відносини між класами: клієнти та нащадки. Конструктори та деструктори. Побудова діаграми класів. Розширення функціоналу батьківського класу: перевантаження, приховування, перевизначення. Спадкування, інкапсуляція та поліморфізм. Множинне наслідування. Інтерфейси. Класи з подіями. Коректна робота з подіями.

Тема 2. ПРИНЦИПИ ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА ОСНОВІ АНАЛІЗУ ПРЕДМЕТНОЇ ОБЛАСТІ

Поняття проектування програмного забезпечення на основі аналізу предметної області. Метод аналізу спільності/змінності. Визначення аналізу предметної області, еквівалентність понять сутність предметної області та класу об'єктів. Загальні принципи проектування. Типові причин перепроєктування. Якісні характеристики коду: гнучкість, переносність, розширюваність. Механізми повторного використання коду: спадкування, композиція, делегування обов'язків. Плюси і мінуси описаних механізмів, випадки найефективнішого використання.

Тема 3. ВВЕДЕННЯ В ШАБЛони ПРОЕКТУВАННЯ

Передумови виникнення шаблонів проектування. Забезпечення багаторазового використання коду. Класифікація патернів. Визначення, історія та причини виникнення патернів проектування: патерни проектування класів/об'єктів (GoF), архітектурні системні патерни, патерни інтеграції корпоративних інформаційних систем. Класифікація патернів проектування класів/об'єктів: породжувальні, структурні та поведінкові патерни. Складові частини патернів проектування. Послідовність дій по розробці програмного забезпечення із використанням патернів проектування.

Тема 4. ПОРОДЖУВАЛЬНІ ПАТЕРНИ

Породжувальні патерни – загальне визначення, призначення, способи використання, випадки застосування: Абстрактна Фабрика (Abstract Factory), Прототип (Prototype), Будівельник (Builder), Одинак (Singleton), Фабричний Метод (Factory Method): визначення, мета використання, переваги та недоліки

використання, застосовність, діаграма класів, учасники шаблонів, програмна реалізація. Загальні висновки стосовно областей та способів використання породжувальних патернів, порівняльна характеристика. Приклади практичного застосування породжувальних патернів.

Тема 5. ЗАГАЛЬНА ХАРАКТЕРИСТИКА СТРУКТУРНИХ ПАТЕРНІВ

Загальний опис структурних патернів. Структурні патерни рівня класів та об'єктів. Програмні механізми, що використовуються при застосуванні структурних патернів. Спільність, різниця, способи реалізації. Випадки та приклади застосування. Патерн Адаптер (Adapter), його різновиди, призначення, проблемні області застосування, переваги та недоліки використання, діаграма класів, учасники шаблону, відношення учасників, результати використання, програмна реалізація, відомі застосування.

Змістовий модуль 2. Застосування патернів проектування

Тема 6. СТРУКТУРНІ ПАТЕРНИ

Патерни Міст (Bridge), Компонувальник (Composite), Фасад (Facade), Декоратор (Decorator), Заступник (Proxy), Легковаговик (Flyweight):

Призначення, синонімічні назви, мотивація використання, застосовність, переваги та недоліки використання, діаграма класів, учасники шаблону, відношення учасників, результати використання, програмна реалізація, відомі застосування.

Тема 7. ПАТЕРНИ ПОВЕДІНКИ

Загальний опис патернів поведінки. Програмні механізми, що використовуються при застосуванні патернів поведінки. Спільність, різниця, способи реалізації. Випадки та приклади застосування. Патерни Ланцюжок Відповідальності (Change of Responsibility), Команда (Command), Ітератор (Iterator), Спостерігач (Observer), Посередник (Mediator), Зберігач (Memento), Стратегія (Strategy), Стан (State), Шаблонний метод (Template Method), Відвідувач (Visitor): призначення, синонімічні назви, мотивація використання, застосовність, переваги та недоліки використання, діаграма класів, учасники шаблону, відношення учасників, результати використання, програмна реалізація, відомі застосування.

Тема 8. ЗАГАЛЬНІ ВИСНОВКИ СТОСОВНО ПАТЕРНІВ ПРОЕКТУВАННЯ КЛАСІВ/ОБ'ЄКТІВ (GOF)

Загальні принципи використання патернів проектування класів/об'єктів: породжувальних, структурних, патернів поведінки. Передумови, недоліки та переваги застосування. Використання патернів проектування – як спосіб створення якісного програмного коду. Порівняльна характеристика, способи та приклади сумісного застосування патернів проектування.

Самостійна робота

№	Назва теми
1	Розв'язання задач із використанням основних принципів програмного забезпечення.
2	Введення в проектування програмного забезпечення із використанням шаблонів. Приклади використання породжувального патерну Фабричний метод (Factory Method).
3	Породжувальний патерн Абстрактна фабрика (Abstract Factory). Розв'язання задач із використанням структурного патерну Адаптер (Adapter).
4	Побудова програмних додатків із використанням структурного патерну фасад (Façade).
5	Загальні принципи побудови програмних додатків із використанням шаблонів поведінки. Патерн ітератор (Iterator).
6	Розробка програмних додатків із використанням патерну поведінки Спостерігач (Observer). Принципи використання патерну поведінки Стратегія (Strategy), програмна реалізація

Методи контролю та оцінювання знань аспірантів Політика щодо академічної доброчесності

Тексти індивідуальних завдань (в т.ч. у разі, коли вони виконуються у формі презентацій або в інших формах) перевіряються на плагіат. Для цілей захисту індивідуального завдання оригінальність тексту має складати не менше 70%. Виключення становлять випадки зарахування публікацій аспірантів у матеріалах наукових конференціях та інших наукових збірниках, які вже пройшли перевірку на плагіат.

Списування під час тестування та інших опитувань, які проводяться у письмовій формі, заборонені (в т.ч. із використанням мобільних девайсів). У разі виявлення фактів списування з боку студента він отримує інше завдання. У разі повторного виявлення призначається додаткове заняття для проходження тестування.

Політика щодо відвідування

Аспірант, який пропустив аудиторне заняття з поважних причин, має продемонструвати викладачу та надати до деканату факультету (відділу докторантури і аспірантури) документ, який засвідчує ці причини.

Аспірант, який пропустив лекційне заняття, повинен законспектувати зміст цього заняття та продемонструвати конспект викладачу до складання заліку.

Аспірант, який пропустив практичне заняття, повинен законспектувати

джерела, які були визначені викладачем як обов'язкові для конспектування, та продемонструвати конспект викладачу до складання заліку, а також виконати індивідуальне завдання, якщо його виконання було передбачене планом заняття.

За об'єктивних причин (хвороба, міжнародне стажування тощо) навчання може відбуватись в он-лайн формі за погодженням із керівником курсу.

Методи контролю

Основні форми участі аспірантів у навчальному процесі, що підлягають поточному контролю: виступ на практичних заняттях; доповнення, запитання до виступаючого, рецензія на виступ; участь у дискусіях; аналіз першоджерел; письмові завдання (тестові, індивідуальні роботи у формі рефератів); та інші письмові роботи, оформлені відповідно до вимог. Кожна тема курсу, що винесена на лекційні та практичні заняття, відпрацьовується аспірантами у тій чи іншій формі, наведеній вище. Обов'язкова присутність на лекційних заняттях, активність впродовж семестру, відвідування/відпрацювання усіх семінарських занять, виконання інших видів робіт, передбачених навчальним планом з цієї дисципліни.

При оцінюванні рівня знань аспіранта аналізу підлягають:

- характеристики відповіді: цілісність, повнота, логічність, обґрунтованість, правильність;
- якість знань (ступінь засвоєння фактичного матеріалу): осмисленість, глибина, гнучкість, дієвість, системність, узагальненість, міцність;
- ступінь сформованості умінь поєднувати теорію і практику під час розгляду ситуацій, практичних завдань;
- рівень володіння розумовими операціями: вміння аналізувати, синтезувати, порівнювати, абстрагувати, узагальнювати, робити висновки з проблем, що розглядаються;
- досвід творчої діяльності: вміння виявляти проблеми, розв'язувати їх, формувати гіпотези;
- самостійна робота: робота з навчально-методичною, науковою, допоміжною вітчизняною та зарубіжною літературою з питань, що розглядаються, вміння отримувати інформацію з різноманітних джерел (традиційних; спеціальних періодичних видань, ЗМІ, Internet тощо).

Тестове опитування може проводитись за одним або кількома змістовими модулями. В останньому випадку бали, які нараховуються аспіранту за відповіді на тестові питання, поділяються між змістовими модулями.

Індивідуальне завдання підлягає захисту аспірантом на заняттях, які признаються додатково.

Індивідуальне завдання може бути виконане у різних формах. Зокрема, аспіранти можуть зробити його у вигляді реферату. Реферат повинен мати обсяг від 18 до 24 сторінок А4 тексту (кегель Times New Roman, шрифт 14, інтервал 1,5), включати план, структуру основної частини тексту відповідно

до плану, висновки і список літератури, складений відповідно до ДСТУ 8302:2015. В рефераті можна також помістити словник базових понять до теми. Водночас індивідуальне завдання може бути виконане в інших формах, наприклад, у вигляді презентації у форматі Power Point. В цьому разі обсяг роботи визначається індивідуально – залежно від теми.

Література, що рекомендується для виконання індивідуального завдання, наведена у цій робочій програмі, а в електронному вигляді вона розміщена на Освітньому сайті КНУБА, на сторінці кафедри.

Також як виконання індивідуального завдання за рішенням викладача може бути зарахована участь аспіранта у міжнародній або всеукраїнській науково-практичній конференції з публікацією у матеріалах конференції тез виступу (доповіді) на одну з тем, дотичних до змісту дисципліни, або публікація статті на одну з таких тем в інших наукових виданнях.

Текст індивідуального завдання подається викладачу не пізніше, ніж за місяць до початку залікової сесії. Заняття із захисту індивідуальних завдань призначаються не пізніше, ніж за 2 тижні до початку сесії. Викладач має право вимагати від студента доопрацювання індивідуального завдання, якщо воно не відповідає встановленим вимогам.

Результати поточного контролю заносяться до журналу обліку роботи. Позитивна оцінка поточної успішності аспірантів за відсутності пропущених та невідпрацьованих семінарських занять та позитивні оцінки за індивідуальну роботу є підставою до підсумкової форми контролю – заліку . Бали за аудиторну роботу відпрацьовуються у разі пропусків.

Підсумковий контроль здійснюється під час проведення залікової сесії з урахуванням підсумків поточного та модульного контролю. Під час семестрового контролю враховуються результати здачі усіх видів навчальної роботи згідно зі структурою кредитів.

Оцінювання проводиться за 100-бальною шкалою. Участь в роботі впродовж семестру – 100.

Форма підсумкового контролю – залік.

Бали нараховуються за наступним співвідношенням:

- семінарські завдання 30% семестрової оцінки;
- індивідуальна робота 30 % семестрової оцінки;
- модульний: тестовий (заліковий) – 40 % семестрової оцінки.

Розподіл балів, які отримують аспіранти

Поточне оцінювання (кількість балів)		Сума
Змістовий модуль №1	Змістовий модуль № 2	
60	40	100

Шкала оцінювання: національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	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 з можливістю повторного складання	не зараховано з можливістю повторного складання
0-34	F	Незадовільно з обов'язковим повторним вивченням дисципліни	не зараховано з обов'язковим повторним вивченням дисципліни

Умови допуску до підсумкового контролю

Аспіранту, який має підсумкову оцінку за дисципліну від 35 до 59 балів, призначається додаткова залікова сесія. В цьому разі він повинен виконати додаткові завдання, визначені викладачем.

Аспірант, який не здав та/або не захистив індивідуальне завдання, не допускається до складання заліку.

Аспірант, який не виконав вимог робочої програми по змістових модулях, не допускається до складання підсумкового контролю. В цьому разі він повинен виконати визначене викладачем додаткове завдання по змісту відповідних змістових модулів в період між основною та додатковою сесіями.

Аспірант має право на опротестування результатів контролю (апеляцію). Правила подання та розгляду апеляції визначені внутрішніми документами КНУБА, які розміщені на сайті КНУБА та зміст яких доводиться до аспірантів на початку вивчення дисципліни.

Методичне забезпечення дисципліни

1. Brett D. McLaughlin, Gary Pollice, David West. Head first object-oriented analysis and design. – O'Reilly Media, Inc, 2007. – 632 p.
2. Д. Нильссон. Применение проблемно-ориентированного проектирования и шаблонов проектирования приложений с примера на C# и .NET. – М.: Вильямс, 2008. – 560с.

3. Д. Кириевски. Рефакторинг с использованием шаблонов. – М.: Вильямс, 2006. – 400с.
4. Э. Фримен, Э. Фримен, К. Сьерра, Б. Гейтс. Паттерны проектирования. – СПб: Питер, 2011. – 656 с.
5. Влссидес Джон. Применение шаблонов проектирования.
6. Дополнительные штрихи. – Пер. с англ. – М.: Издательский дом «Вильямс», 2003. – 144 с.
7. Гамма Э., Хелм Р., Джонсон Р., Влссидес Дж. Приемы объектноориентированного проектирования. Паттерны проектирования. – СПб: Питер, 2001. – 368 с.
8. Будай А. Дизайн-паттерни – простіше простого. – Львів, 2012. – 90с.
9. Cooper J.W. Java Design Patterns. Addison-Wesley Professional, 2000 – 542.
10. James W. Cooper. Introduction to design patterns in C#. – IBM T J Watson Research Center, 2002. – 424 p.
11. Jean Paul V.A. Design Patterns in C#. 2012. – 94 p.

Інформаційні ресурси

1. <http://library.knuba.edu.ua/>
2. <http://cpp-reference.ru/patterns/>
3. <http://www.dofactory.com/Patterns/Patterns.aspx>
4. <http://skillcoding.com/Default.aspx?id=222>
5. <http://msdn.microsoft.com/ru-ru/library/dd460654.aspx#Classes>
6. <http://www.skillcoding.com/Default.aspx?id=123>