

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ПРИКАРПАТСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАСИЛЯ СТЕФАНИКА**

Фізико-технічний факультет
Кафедра комп'ютерної інженерії та електроніки

**СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
Об'єктно-орієнтоване проектування**

Освітня програма Бакалавр
Галузь знань 12 Інформаційні технології
Спеціальність 123 Комп'ютерна інженерія

Затверджено на засіданні кафедри
Протокол № 1 від "30" серпня 2023 р.

Івано-Франківськ – 2023 рік

ЗМІСТ

1. Загальна інформація
2. Анотація до курсу
3. Мета та цілі курсу
4. Компетентності
5. Результати навчання
6. Організація навчання курсу
7. Система оцінювання курсу
8. Політика курсу
9. Рекомендована література

1. Загальна інформація про викладача і дисципліну

Назва дисципліни	Об'єктно-орієнтоване проектування
Рівень вищої освіти	Перший рівень вищої освіти
Викладач	Старший викладач, кандидат технічних наук Котик Михайло Васильович
Контактний телефон викладача	(0342) 59-60-07
Е-mail викладача	mykhaylo.kotyk@pnu.edu.ua
Формат дисципліни	Семестровий
Обсяг дисципліни	6 кредитів
Посилання на сайт дистанційного навчання	http://www.d-learn.pnu.edu.ua
Консультації	Відповідно до графіку індивідуальних консультацій, який розміщений на інформаційному стенді кафедри комп'ютерної інженерії та електроніки

2. Анотація до курсу

Дисципліна “Об'єктно-орієнтоване проектування” належить до переліку вибіркових навчальних дисциплін підготовки за освітнім рівнем «бакалавр», що пропонуються в рамках циклу вільного вибору підготовки студентів за освітньо-професійною програмою “Комп'ютерна інженерія”. Вона забезпечує формування у студентів загальних і спеціальних (професійно-орієнтованих) компетенцій. Предметом вивчення навчальної дисципліни є методологія та технології об'єктно-орієнтованого проектування програмних систем. Силабус дисципліни “Об'єктно-орієнтоване проектування” складений відповідно до освітньо-професійної програми підготовки бакалавра спеціальності 123 “Комп'ютерна інженерія”.

3. Мета та цілі курсу

Мета: сформувати у студентів необхідні теоретичні знання з принципів об'єктно-орієнтованого проектування (ООП), шаблонів проектування GoF, побудови UML діаграм класів, побудови автономних тестів, проведення рефакторингу програмного коду, реалізації проектів на мові Java.

Завдання:

- знайомство з сучасним станом, тенденціями та перспективами розвитку ООП;
- шаблонів проектування GoF;
- набуття студентами теоретичних знань та практичних навиків з принципів об'єктно-орієнтованої декомпозиції складних об'єктів;
- вивчення мови UML і середовища розробки Umbrello;
- застосування мови UML для описання об'єктів і їх взаємозв'язків у заданій предметній області.
- набуття практичних навиків з побудови основних типів UML-діаграм;
- набуття практичних навиків з побудови автономних тестів;
- набуття практичних навиків з рефакторингу програмного коду;

- набуття практичних навиків автоматичної генерації з UML-діаграм кодів програми на мові Java.

У результаті вивчення навчальної дисципліни студент повинен

знати:

- фундаментальні концепції та основні принципи ООП;
- шаблони проектування GoF;
- мову об'єктно-орієнтованого проектування UML;
- призначення різних UML-діаграм;
- структуру UML класів і типи відношень між класами;
- технологію автоматичної генерації з UML-діаграм класів кодів програм на мові Java;
- механізми об'єктно-орієнтованого програмування – інкапсуляція, успадкування, поліморфізм.

- побудову автономних тестів;
- рефакторинг програмного коду;

вміти:

- аналізувати задану предметну область, проводити її об'єктно-орієнтовану декомпозицію, визначати класи та об'єкти;
- застосовувати шаблони проектування GoF;
- будувати концептуальну модель UML в середовищі Umbrello;
- встановлювати зв'язки між класами та об'єктами і будувати UML діаграми;
- автоматично генерувати з UML-діаграми класів коди програм на мові Java;
- будувати автономні тести;
- виконувати рефакторинг програмного коду.

4. Компетентності

- Загальні

- ЗКЗ. Здатність застосовувати знання у практичних ситуаціях.

Спеціальні (фахові)

- Р2. Здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення.
- Р3. Здатність створювати системне та прикладне програмне забезпечення комп'ютерних систем та мереж.
- *Здатність використовувати методологію і технологію об'єктно-орієнтованого проектування при розробленні програмних систем.*

5. Результати навчання

- N6. Вміти застосовувати знання для ідентифікації, формулювання і розв'язування технічних задач спеціальності, використовувати методи, що є найбільш придатними для досягнення поставлених цілей.

6. Організація навчання курсу

Обсяг курсу

Вид заняття		Загальна кількість годин			
лекції		26			
семінарські заняття / практичні / лабораторні		38			
самостійна робота		116			
Ознаки курсу					
Семестр	Спеціальність	Курс (рік навчання)		Нормативний/ вибірковий	
8	123 Комп'ютерна інженерія	4		вибірковий	
Тематика курсу					
Тема	Форма заняття, год.	Література	Кількість годин	Вага оцінки	Термін виконання
Змістовий модуль 1. Теорія ООП і шаблони GoF.					
Тема 1. SOLID принципи ООП.	лекція	20	2	2	Згідно розкладу
Тема 2. Основні принципи ООП. Шаблони проектування GoF та Rasp.	лекція	5, 7, 14	2	2	Згідно розкладу
Тема 3. Шаблони проектування і їх реалізація.	лекція	5, 7, 14, 16	2	2	Згідно розкладу
Тема 4. Мова візуального проектування UML.	лекція	4, 7, 8, 9, 10, 13	2	2	Згідно розкладу
Тема 5. Синтаксис і семантика основних об'єктів UML.	лекція	4, 7, 8, 9, 10, 13	2	2	Згідно розкладу
Тема 6. Усунення залежностей між класами.	лекція	8, 15	2	2	Згідно розкладу
Модульний контроль 1			2	2	
Змістовий модуль 2. Технології ООП і мова програмування Java.					
Тема 7. Огляд технологій JAVA і IDE.	лекція	1, 2, 3, 18	2	2	Згідно розкладу
Тема 8. Основи мови Java.	лекція	2, 17, 22	2	2	Згідно розкладу
Тема 9. Можливості мов програмування з реалізації механізмів ООП.	лекція	1, 2 3, 21, 22	2	2	Згідно розкладу
Тема 10. Класи стандартної бібліотеки Java.	лекція	1, 2 3, 21, 22	2	2	Згідно розкладу
Тема 11. Генерація Java програмного коду в системі Umbrello.	лекція	4, 28	2	2	Згідно розкладу

Тема 12. Побудова автоматичних тестів.	лекція	22, 23, 24	2	2	Згідно розкладу
Тема 13. Рефакторинг програмного коду.	лекція	22, 23, 24	2	2	Згідно розкладу
Модульний контроль 2			2		
Практичний модуль					
1. Графічне середовище Umbrello UML modeler.	лаб. робота	27, 28	2	2	Згідно розкладу
2. Сутності, діаграми і відношення в UML. Діаграми варіантів використання (use case).	лаб. робота	7, 8, 27, 28	2	2	Згідно розкладу
3. Діаграма класів (class diagram).	лаб. робота	7, 8, 27, 28	2	2	Згідно розкладу
4. Діаграма послідовності (sequency diagram).	лаб. робота	2, 3, 6, 11, 16, 17, 18	4	3	Згідно розкладу
5. Діаграма взаємодії (interaction diagram).	лаб. робота	28, 29	2	3	Згідно розкладу
6. Діаграма діяльності (activity diagram).	лаб. робота	14, 15	4	3	Згідно розкладу
7. Діаграма станів автомату (state machine).	лаб. робота	4, 5, 7, 11, 19, 21	4	3	Згідно розкладу
Діаграма пакетів і діаграма компонентів (package diagram, component diagram).	лаб. робота	4, 5, 7, 11, 19, 21	4	3	Згідно розкладу
9. Реалізація відношення залежності.	лаб. робота	4, 5, 7, 11, 19, 21	4	3	Згідно розкладу
10. Реалізація відношення асоціації на мовах програмування.	лаб. робота	3, 11, 12, 17, 18, 19	4	3	Згідно розкладу
11. Реалізація відношення агрегації та композиції на мовах програмування.	лаб. робота	17, 18, 19	4	3	Згідно розкладу
12. Реалізація відношення успадкування на мовах програмування.		22, 24	2	3	Згідно розкладу
Самостійна робота					
Тема 1. Засоби проектування UML діаграм.	самостійна робота	4, 27, 28	5	3	Впродовж семестру
Тема 2. Переваги і недоліки ООП.	самостійна робота	5, 8, 17, 21	5	3	Впродовж семестру
Тема 3. Об'єктно-орієнтована	самостійна робота	7, 8, 14,	6	3	Впродовж

інженерія програм (OOSE).	на робота	20, 21			семестру
Тема 4. Особливості реалізації породжуючих шаблонів проектування засобами мови C/C++, Python, Java.	самостій на робота	14, 19, 20, 24	10	3	Впродовж семестру
Тема 5. Особливості реалізації структурних шаблонів проектування засобами мови C/C++, Python, Java.	самостій на робота	14, 19, 20, 24	10	3	Впродовж семестру
Тема 6. Особливості реалізації поведінкових шаблонів проектування засобами мови C/C++, Python, Java.	самостій на робота	14, 19, 20, 24	10	3	Впродовж семестру
Тема 7. Елементарні шаблони проектування – iterator, singleton, adapter.	самостій на робота	19, 20	10	3	Впродовж семестру
Тема 8. Моделювання програм з використанням машини станів.	самостій на робота	21, 22, 23	10	3	Впродовж семестру
Тема 9. Реалізація ін'єкції залежностей у мовах C/C++, Python, Java.	самостій на робота	15, 20	10	3	Впродовж семестру
Тема 10. Автоматизація тестів засобами JUNIT.	самостій на робота	22, 24	10	3	Впродовж семестру
Тема 10. Автоматизація тестів засобами Mockito.	самостій на робота	22, 24	10	3	Впродовж семестру
Тема 12. Рефакторинг і масштабування програм в IntelliJ	самостій на робота	24, 31, 32, 34	10	3	Впродовж семестру
Тема 13. Рефакторинг і масштабування програм в NetBeans	самостій на робота	24, 31, 32, 33	10	3	Впродовж семестру
Контроль самостійної роботи			2	3	Згідно розкладу
Підсумковий контроль (залік)				100	Згідно розкладу

7. Система оцінювання курсу

Загальна система оцінювання курсу

Поточний контроль здійснюється під час проведення лекційних, лабораторних, індивідуальних занять і має на меті перевірку знань студентів з окремих тем навчальної дисципліни та рівня їх підготовленості до виконання конкретної роботи. Оцінки у національній шкалі («відмінно» - 5, «добре» - 4, «задовільно» - 3, «незадовільно» - 2), отримані студентами, виставляються у журналах обліку відвідування та успішності академічної групи.

Модульний контроль (сума балів за окремих змістовий модуль) проводиться (виставляється) на підставі оцінювання результатів знань студентів після вивчення матеріалу з логічно завершеної частини дисципліни – змістового модуля.

Завданням модульного контролю є перевірка розуміння та засвоєння певного матеріалу (теми), вироблення навичок проведення розрахункових робіт, вміння вирішувати конкретні ситуативні задачі, самостійно опрацьовувати тексти, здатності осмислювати зміст даної частини дисципліни, уміння публічно чи письмово подати певний матеріал.

Семестровий (підсумковий) контроль проводиться у формі заліку.

Залік – форма підсумкового контролю, що полягає в оцінюванні засвоєння здобувачем навчального матеріалу з певної дисципліни, і складається із зданих залікових змістових модулів, виконаних тестових завдань, ситуаційних робіт, опрацювання завдань робочих зошитів, тематичних рефератів, лабораторних робіт тощо, передбачених навчальною програмою.

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
80-89	B	добре	
70-79	C		
60-69	D	задовільно	
50-59	E		
26-49	FX	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
0-25	F	незадовільно з обов'язковим повторним вивченням дисципліни	не зараховано з обов'язковим повторним вивченням дисципліни

Вимоги до письмової роботи

Підсумкова письмова роботи виконується у формі тестових завдань. Кількість тестових завдань 25.

Лабораторні заняття

До початку лабораторної роботи студент має отримати допуск на основі усної співбесіди з викладачем. На лабораторній роботі кожний студент отримує варіант завдання до виконання на занятті і домашнього завдання для самостійної роботи. Після завершення роботи студент захищає поточне завдання і домашні завдання попередніх лабораторних робіт оформленні як звіти.

Умови допуску до підсумкового контролю

Студент допускається до складання заліку, якщо впродовж семестру він за змістові модулі набрав сумарно 25 балів і вище.

Студент не допускається до складання заліку, якщо впродовж семестру він за змістові модулі набрав менше 25 балів. У цьому випадку студенту у відомості робиться запис “не допущений” і виставляється набрана кількість балів. Допускається, як виняток, з дозволу декана факультету за заявою, погодженою з відповідною кафедрою, одноразове виконання студентом додаткових видів робіт з навчальної дисципліни (відпрацювання пропущених занять, перескладання змістових модулів, виконання індивідуальних завдань тощо) для підвищення оцінок за змістові модулі.

Напередодні заліку викладач подає доповідну декану про недопуск студентів академічної групи (груп). Відмітка про недопуск у відомості робиться при наявності розпорядження декана.

8. Політика курсу

Студент зобов'язаний відвідувати заняття відповідно до встановленого розкладу, не запізнюватися, мати відповідний зовнішній вигляд. У разі відсутності через хворобу надається відповідна довідка.

Пропущена лекція відпрацьовується студентом самостійно і оформляється як короткий конспект за темою заняття.

Пропущена лабораторна робота виконується студентом самостійно вдома або в комп'ютерному класі, результати оцінюються викладачем.

У випадку, коли студент приймав участь у програмі мобільності, можливе врахування отриманих оцінок в іншому навчальному закладі за умови відповідності навчальних планів.

Можливе зарахування результатів неформальної освіти згідно з Положенням про порядок зарахування результатів неформальної освіти у Прикарпатському національному університеті імені Василя Стефаника.

Політика академічної поведінки і етики

Студент повинен бути толерантним і поважати думку інших.

Заперечення повинні формулюватися тільки в коректній формі.

Академічна доброчесність здобувачами вищої освіти передбачає:

- самостійне виконання навчальних завдань, завдань поточного та підсумкового контролю результатів навчання;

- посилення на джерела інформації у разі використання ідей, розробок, тверджень, відомостей;

- дотримання норм законодавства про авторське право і суміжні права;
- надання достовірної інформації про результати власної навчальної (наукової, творчої) діяльності, використанні методики досліджень і джерелі інформації.

Не допускається підказування та списування під час здачі будь-яких робіт поточного, семестрового, підсумкового контролю.

Не допускається користування телефонами та будь-якими іншими електронними засобами під час здачі будь-яких робіт поточного, семестрового, чи підсумкового контролю.

9 Рекомендована література

Основна

1. Mark Seemann, Steven van Deursen. Dependency injection Principles, Practices, and Patterns. - Shelter Island: "Manning", 2019. – 552 p.
2. Vaskaran Sarcar. Java Design Patterns: A Hands-On Experience with Real-World Examples, 2nd Edition. - "Apress", 2019. – 500 p.
3. Jouce Farrell. Java programming. 9 ed. – Boston: Cengage. – 2019 – 898 p.
4. Horstman Cay. Core Java SE9 for the Impatient. – Second edition. – Addison Wesley, 2018.– 1818 p.
5. John F. Dooley. Software Development, Design and Coding: With Patterns, Debugging, Unit Testing, and Refactoring, 2nd Edition. - "Apress", 2017. – 320 p.
6. Bipin Joshi. Beginning SOLID Principles and Design Patterns for ASP.NET Developers. "Apress", 2016. – 415 p.
7. Scott Millett and Nick Tune. Patterns, Principles, and Practices of Domain-Driven Design. – "Wrox", 2015.
8. Paul Deitel, Harvey Deitel. Java. How to program. 10 ed. – New Jersey: Pearson Education, 2015. – 1245 p.
9. Rudolf Pecinovsky. OOP – Learn Object Oriented Thinking and Programming. - "Tomas Bruckner Publishing", 2013. – 502 p.
10. Wu, C. Thomas. An introduction in object-oriented programming with Java. 5 ed. – New York: McGraw-Hill, 2010. – 1009 p.

Інтернет ресурси

11. <http://lib.ru.if.ua/> – наукова бібліотека Прикарпатського національного університету імені Василя Стефаника.
12. <http://www.nbuv.gov.ua/> – Національна бібліотека України імені В.І. Вернадського.
13. <https://umbrello.kde.org> – розробник системи Umbrello.
14. institute.cn.ua – підручник з Umbrello UML Modeller.
15. <https://martinfowler.com/articles/injection.html> – рефакторинг і шаблони.
16. <https://docs.oracle.com/javase/tutorial/> – Oracle. The Java™ Tutorials.
17. <https://www.tutorialspoint.com/java/> – курси Java.
18. <https://www.bestprog.net> – теорія Java
19. <https://netbeans.org> – інтегроване середовище розробки Netbeans
20. <https://www.jetbrains.com> - інтегроване середовище розробки IntelliJ

Викладач



Голога В.І.