

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДВНЗ «ПРИКАРПАТСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАСИЛЯ СТЕФАНІКА»**

Фізико-технічний факультет
Кафедра комп'ютерної інженерії та електроніки

СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Об'єктно-орієнтоване проектування

Освітня програма «Комп'ютерне проектування інтегральних схем»

Спеціальність 171 Електроніка

Галузь знань 17 Електроніка та комунікації

Затверджено на засіданні кафедри
Протокол № 1 від “30” серпня 2021 р.

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	
Назва дисципліни	Об'єктно-орієнтоване проектування
Рівень вищої освіти	Перший рівень вищої освіти
Викладач (-і)	доцент, кандидат технічних наук Голота Віктор Іванович
Контактний телефон викладача	+38(0342) 59-60-07
Е-mail викладача	victor.holota@pnu.edu.ua
Формат дисципліни	Семестровий
Обсяг дисципліни	6 кредитів
Посилання на сайт дистанційного навчання	http://www.d-learn.pu.if.ua/
Консультації	відповідно до графіку індивідуальних консультацій, який розміщений на інформаційному стенді кафедри комп'ютерної інженерії та електроніки
2. Анотація до курсу	
<p>Дисципліна “Об'єктно-орієнтовне проектування” належить до переліку вибіркових навчальних дисциплін за освітнім рівнем “бакалавр”, що пропонуються в рамках циклу загальної підготовки студентів за освітньою програмою “Комп'ютерне проектування інтегральних схем” спеціальності 171 “Електроніка” на третьому році навчання. Вона забезпечує формування у студентів практичних професійно-орієнтованих компетенцій. Предметом вивчення навчальної дисципліни є методи об'єктно-орієнтованого проектування програмних систем з використанням шаблонів проектування.</p> <p>Силабус навчальної дисципліни “Об'єктно-орієнтовне проектування” складений відповідно до освітньо-професійної програми “Комп'ютерне проектування інтегральних схем” підготовки бакалавра спеціальності 171 “Електроніка”.</p>	
3. Мета та цілі курсу	
<p>Метою викладання навчальної дисципліни є набуття студентами необхідних теоретичних знань та практичних навиків для використання технологій об'єктно-орієнтованого проектування (ООП), мови моделювання UML і шаблонів проектування при розробці програмних проектів на мові Java.</p> <p>Основними завданнями вивчення дисципліни є:</p> <ul style="list-style-type: none"> - знайомство з сучасним станом, тенденціями та перспективами розвитку ООП і використання шаблонів проектування; - вивчення різних типів шаблонів проектування і використання їх при реалізації прикладних програм; - поглиблення студентами теоретичних знань та практичних навиків з принципів об'єктно-орієнтованої декомпозиції складних об'єктів; - застосування мови UML для описання об'єктів і їх взаємозв'язків у заданій предметній області у середовищі розробки Visual Paradigm. - отримання практичних навиків з побудови різних типів UML-діаграм; - вивчення об'єктно-орієнтованих і розширених можливостей мови Java; - отримання практичних навиків автоматичної генерації з UML-діаграм кодів програм на мові Java. <p>У результаті вивчення навчальної дисципліни студент повинен знати:</p> <ul style="list-style-type: none"> - основні принципи ООП; - застосування шаблонів при проектуванні програмних систем; 	

- технології об'єктно-орієнтованого моделювання з використанням UML-діаграм;
- призначення UML-діаграм різних типів;
- типи відношень між класами і об'єктами;
- технологію автоматичної генерації з UML-діаграм класів кодів програм на мові Java;
- об'єктно-орієнтовані і розширені можливості мови Java;

вміти:

- аналізувати задану предметну область, проводити її об'єктно-орієнтовану декомпозицію, визначати класи та об'єкти;
- будувати концептуальну модель і UML діаграми в середовищі Visual Paradigm;
- встановлювати і визначати типи зв'язків між класами та об'єктами;
- автоматично генерувати з UML-діаграми класів коди програм на мові Java;
- використовувати в програмах на мові Java породжуючі і структурні шаблони проектування.

4. Компетентності

Спеціальні (фахові) компетентності

СК5. Здатність застосовувати відповідні математичні, наукові й технічні методи, сучасні інформаційні технології і комп'ютерне програмне забезпечення, навички роботи з комп'ютерними мережами, базами даних та Інтернет-ресурсами для вирішення інженерних задач в галузі електроніки.

Додаткові компетентності

Здатність розробляти програмне забезпечення на мові Java з використанням технологій об'єктно-орієнтованого проектування і шаблонів проектування.

5. Результат навчання

Р5. Використовувати інформаційні та комунікаційні технології, прикладні та спеціалізовані програмні продукти для вирішення задач проектування та налагодження електронних систем, демонструвати навички програмування, аналізу та відображення результатів вимірювання та контролю.

6. Організація навчання курсу

Обсяг курсу

Вид заняття	Загальна кількість годин
лекції	30
семінарські заняття / практичні / лабораторні	40
самостійна робота	110

Ознаки курсу

Семестр	Спеціальність	Курс (рік навчання)	Нормативний/ вибірковий
5	171 Електроніка	3	Нормативний

Тематика курсу

Тема, план	Форма заняття	Література	К-ть годин, год	Вага оцінки	Термін виконання
------------	---------------	------------	-----------------	-------------	------------------

Змістовий модуль 1. Проектування і реалізація UML-діаграм

Тема 1. Особливості ООП стилю в мові Java. Інкапсуляція, абстракція, успадкування, поліморфізм.	лекція	[6-8,11, 12,14,15, 17,18,21-23]	2	1	Згідно розкладу
Тема 2. Абстрактні класи та інтерфейси.	лекція	[6-8,11, 12,14,15,	2	1	Згідно розкладу

		17,18,21-23]			
Тема 3. Сучасний стан і засоби ООП технологій	лекція	[1-5,9,10,13,16,19,20,24-29]	2	1	Згідно розкладу
Тема 4. Типи відношень в UML-діаграмах між класами і об'єктами.	лекція	[1-5,9,10,13,16,19,20,24-29]	2	1	Згідно розкладу
Тема 5. Усунення залежностей між класами.	лекція	[1-5,9,10,13,16,19,20,24-29]	2	1	Згідно розкладу
Тема 6. Реалізація відношень між класами і об'єктами на мові Java.	лекція	[6-8,11,12,14,15,17,18,21-23]	2	1	Згідно розкладу
Модульний контроль 1			2		Згідно розкладу
Змістовий модуль 2. Шаблони проектування					
Тема 7. Типи відношень в UML-діаграмах класів.	лекція	[4,5,9,10,13,16,19,20,24-29]	2	1	Згідно розкладу
Тема 8. Шаблони проектування GoF	лекція	[4,5,9,10,13,16,19,20,24-29]	2	1	Згідно розкладу
Тема 9. Породжуючі шаблони проектування і їх реалізація на мові Java	лекція	[4,5,9,10,13,16,19,20,24-29]	2	2	Згідно розкладу
Тема 10. Структурні шаблони проектування і їх реалізація на мові Java.	лекція	[1-5,9,10,13,16,19,20,24-29]	2	2	Згідно розкладу
Тема 11. Поведінкові шаблони проектування і їх реалізація на мові Java.	лекція	[4,5,9,10,13,16,19,20,24-29]	2	2	Згідно розкладу
Тема 12. Шаблони проектування GRASP	лекція	[4,5,9,10,13,16,19,20,24-29]	4	2	Згідно розкладу
Модульний контроль 2			2		Згідно розкладу
Лабораторні заняття					
1. Інтегровані середовища розроблення Java програм IntelliJ, NetBeans, Eclipse.	Лаб. робота	[30,31,32]	2	2	Згідно розкладу
2. ООП в мові Java	Лаб. робота	[8,11,12,14,15,17]	2	2	Згідно розкладу
3. Шаблон проектування Абстрактна Фабрика/Abstract Factory.	Лаб. робота	[4,5,9,10,13,16,19]	4	2	Згідно розкладу
4. Шаблон проектування Будівельник/Builder.	Лаб. робота	[4,5,9,10,13,16,19]	4	2	Згідно розкладу
5. Шаблон проектування фабричний метод/Factory Method.	Лаб. робота	[4,5,9,10,13,16,19]	4	2	Згідно розкладу
6. Шаблон проектування	Лаб.	[4,5,9,10,	4	2	Згідно

Прототип/Prototype.	робота	13,16,19]			розкладу
7. Шаблон проектування Одинак/Singleton	Лаб. робота	[4,5,9,10, 13,16,19]	4	2	Згідно розкладу
8. Шаблон проектування Адаптер/Adapter.	Лаб. робота	[4,5,9,10, 13,16,19]	4	2	Згідно розкладу
9. Шаблон проектування Міст/Bridge	Лаб. робота	[4,5,9,10, 13,16,19]	4	2	Згідно розкладу
10. Шаблон проектування Компонувальник/Composite	Лаб. робота	[4,5,9,10, 13,16,19]	4	2	Згідно розкладу
11. Шаблон проектування Декоратором (Decorator)	Лаб. робота	[4,5,9,10, 13,16,19]	4	2	Згідно розкладу
12. Шаблон проектування Заступник (Proxy)	Лаб. робота	[4,5,9,10, 13,16,19]	4	2	Згідно розкладу
Самостійна робота студентів					
Тема 1. Винятки та їх оброблення у Java.	Само- стійна робота	[6-8,11, 12,14,15, 17,18,21- 23]	8	1	Впродовж семестру
Тема 2. Родові (generic) методи та класи Java.	Само- стійна робота	[6-8,11, 12,14,15, 17,18,21- 23]	8	1	Впродовж семестру
Тема 3. Проектування шаблонів у Java.	Само- стійна робота	[6-8,11, 12,14,15, 17,18,21- 23]	10	1	Впродовж семестру
Тема 4. Подійно орієнтоване програмування в Java.	Само- стійна робота	[6-8,11, 12,14,15, 17,18,21- 23]	8	1	Впродовж семестру
Тема 5. Потоки у Java.	Само- стійна робота	[6-8,11, 12,14,15, 17,18,21- 23]	10	1	Впродовж семестру
Тема 6. Реалізація шаблонів у Java.	Само- стійна робота	[6-8,11, 12,14,15, 17,18,21- 23]	10	1	Впродовж семестру
Контроль самостійної роботи			2		Згідно розкладу
Тема 7. Огляд програмних засобів для побудови UML-діаграм.	Само- стійна робота	[1-5,9,10, 13,16,19, 20,24-29]	6	1	Впродовж семестру
Тема 8. Переваги і недоліки ООП.	Само- стійна робота	[1-5,9,10, 13,16,19, 20,24-29]	8	1	Впродовж семестру
Тема 9. Ідентифікація об'єктів під час виконання.	Само- стійна робота	[1-5,9,10, 13,16,19, 20,24-29]	10	1	Впродовж семестру
Тема 10. Тестування програм.	Само- стійна	[9,10]	10	1	Впродовж семестру

	робота				
Тема 11. Рефакторинг і масштабування програм.	Само-стійна робота	[1-5,9,10,13,16,19,20,24-29]	10	1	Впродовж семестру
Тема 12. Моделювання програм з використанням машини станів.	Само-стійна робота	[1-5,9,10,13,16,19,20,24-29]	10	1	Впродовж семестру
Контроль самостійної роботи			2		Згідно розкладу
Підсумковий контроль (екзамен)				50	

7. Система оцінювання курсу

Загальна система оцінювання курсу	<p><i>Поточний контроль</i> здійснюється під час проведення лекційних, лабораторних та 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 («зараховано», «незараховано»), отримані студентами, виставляються у журналах обліку відвідування та успішності академічної групи.</p> <p><i>Модульний контроль</i> (сума балів за окремий змістовий модуль) проводиться (виставляється) на підставі оцінювання результатів знань студентів після вивчення матеріалу з логічно завершеної частини дисципліни – змістового модуля.</p> <p>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.</p> <p><i>Семестровий (підсумковий) контроль</i> проводиться у формі екзамену.</p> <p><i>Екзамен</i> – форма підсумкового контролю, яка передбачає перевірку розуміння студентом теоретичного та практичного програмного матеріалу з усієї дисципліни, здатності творчо використовувати здобуті знання та вміння, формувати власне ставлення до певної проблеми тощо.</p> <p>Екзамен здійснюється в письмовій формі за підсумковим тестовим завданням, що дає можливість здійснити оцінювання знань студента з усієї дисципліни або у тестовій формі з використанням комп'ютерного автоекзаменатора.</p>				
	Шкала оцінювання: національна та ECTS				
	Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою		
			для екзамену, курсового проєкту (роботи)	для заліку	
90-100	A	відмінно	зараховано		

	80-99	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 робота виконується у формі тестових завдань з вибором правильної відповіді. Кількість тестових завдань – 25.			
Лабораторні заняття	<p>Після узагальнення (вступного слова) викладач дає відповіді на окремі теоретичні запитання, які виникли в студентів у процесі підготовки до заняття. Зазвичай з кожної теми лекційного курсу на лабораторні заняття виносять індивідуалізовані теми комплексного характеру, які дають змогу студенту ширше застосувати здобуті знання та підготуватися до самостійного виконання домашнього завдання.</p> <p>Для перевірки рівня засвоєння навчального матеріалу студенти виконують тестові завдання.</p> <p>До початку лабораторної роботи студент має отримати допуск на основі усної співбесіди. На лабораторній роботі кожен студент отримує інструкцію до виконання. Після завершення роботи студент оформляє і захищає звіт з результатами роботи.</p>			
Умови допуску до підсумкового контролю	<p>Студент допускається до підсумкового контролю (екзамену) за наявності виконаних лабораторних завдань, результатів тестування за тематикою лекційних завдань та самостійної роботи.</p> <p>Також є можливість перезарахування результатів навчання в інших закладах вищої освіти чи результатів неформальної освіти згідно Положення про порядок зарахування результатів неформальної освіти у ДВНЗ “Прикарпатський національний університет імені Василя Стефаника” (затверджено вченою радою університету 27.11.2019 р. протокол № 10 та введено в дію наказом ректора № 819 від 29.11.2019 р.).</p> <p>Студент не допускається до підсумкового контролю, якщо впродовж семестру він набрав менше 50 балів із перерахованих вище категорій занять. 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</p>			

індивідуальних завдань тощо) для підвищення оцінок за змістові модулі.

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, 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.

Політика академічної поведінки та етики

Студент повинен бути толерантним і поважати думку інших.

Заперечення повинні формулюватися тільки в коректній формі.

Плагіат та академічна недоброчесність несумісні з принципами діяльності ЗВО.

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

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

9. Рекомендована література

Основна

1. Навчальний посібник з дисципліни «Технології розробки програмного забезпечення» для студентів спеціальності 123 «Комп'ютерна інженерія». – Полтава: ПолНТУ, 2017. – 218с.

2. К.Є. Золотко, Д.В. Красношарпа, С.Ф.Сірик, Методичні рекомендації до виконання лабораторних робіт з дисципліни «Проектування програмних систем». – Дніпро, 2018. – 27с.

3. Методичні вказівки до лабораторного практикуму з дисципліни „Конструювання програмного забезпечення” для студентів напряму підготовки 6.050103 - “Програмна інженерія” / Укл. А. М. Акименко, І. В. Богдан — Чернігів: ЧПБіП, 2016. — 34с.

Додаткова

4. Mark Seemann, Steven van Deursen. Dependency injection Principles, Practices, and Patterns. - Shelter Irland: “Manning”, 2019. - 552 p.

5. Vaskaran Sarcar. Java Design Patterns: A Hands-On Experience with Real-World Examples, 2nd Edition. - “Apress”, 2019. - 500 p.

6. Jouce Farrell. Java programming. 9 ed. – Boston: Cengage. – 2019 – 898 p.

7. Николай Прохоренко. Основы Java. – СПб.: БХВ-Петербург, 2019. – 768 с.

8. Horstman Cay. Core Java SE9 for the Impatient. – Second edition. – Addison Wesley, 2018.– 1818 p.

9. John F. Dooley. Software Development, Design and Coding: With Patterns, Debugging, Unit Testing, and Refactoring, 2nd Edition. - “Apress”, 2017. - 320 p.

10. Bipin Joshi. Beginning SOLID Principles and Design Patterns for ASP.NET Developers. “Apress”, 2016. - 415 p.

11. Брюс Эккель. Философия Java. – СПб.: «Питер», 2016. – 1168 с.

12. Джошуа Блох. Java. Эффективное программирование. Изд. «Лори», 2016. – 440 с.

13. Scott Millett and Nick Tune. Patterns, Principles, and Practices of Domain-Driven Design. -

“Wrox”, 2015.

14. Paul Deitel, Harvey Deitel. Java. How to program. 10 ed. – New Jersey: Pearson Education, 2015. – 1245 p.

15. Арнольд К., Гослинг Д. Язык программирования Java. – Санкт-Петербург: «Издательский дом Питер», 2013.– 745 с.

16. Rudolf Pecinovsky. OOP – Learn Object Oriented Thinking and Programming. – “Tomas Bruckner Publishing”, 2013. - 502 p.

17. Хабибуллин И. Самоучитель Java. – Санкт-Петербург: «БХВ-Петербург», 2012. – 750 с.

18. Аккуратов Е.Е. Самоучитель. Знакомьтесь: Java. – М.: Издательский дом «Вильямс», 2012. – 250 с.

19. Мартин Фаулер. UML. Основы. Краткое руководство по стандартному языку объектного моделирования. – СПб.: Символ-Плюс, 2011. – 192 с.

20. Э. Гамма. Приемы объектно-ориентированного проектирования. / Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес. – СПб.: Питер, 2011. – 366 с.

21. Wu, C. Thomas. An introduction in object-oriented programming with Java. 5 ed. – New York: McGraw-Hill, 2010. – 1009 p.

22. Флэнаган Д. Java в примерах. Справочник.– Санкт-Петербург– Москва: «Издательский дом Питер», 2009.– 642 с.

23. Хорстман К.С., Корнелл Г. Java 2. – Том 1 Основы. – М.: Издательский дом «Вильямс», 2007. – 896 с.

24. Арлоу Д. UML 2 и унифицированный процесс. Практический объектно-ориентированный анализ и проектирование, 2-е издание / Арлоу Д., Нейштадт И.– СПб: Символ Плюс, 2007. – 624 с.

25. Буч Г., Якобсон А., Рамбо Дж. UML. Классика CS. 2-е изд. / Пер. с англ.; Под общей редакцией проф. С. Орлова – СПб.: Питер, 2006. – 736 с.

26. Грейди Буч, Джеймс Рамбо, Айвар Джекобсон. Язык UML. Руководство пользователя. – М., СПб.: ДМК Пресс, Питер, 2004. – 432 с.

27. Кватрани Е. Rational Rose 2000 и UML. Визуальное моделирование. Издательский дом «Вильямс». М., СПб, К.2003. 176с.

28. Леоненков А.В. Самоучитель UML. – СПб: БХВ-Петербург, 2001.

29. Буч, Гради. Объектно-ориентированный анализ и проектирование. Пер. с англ. 2-е изд. М.: «Бином». 2000.-560с.

Інформаційні ресурси

30. <https://www.jetbrains.com/> - інтегроване середовище для розроблення Java програма

31. <https://netbeans.apache.org/> - інтегроване середовище для розроблення програм

32. <https://www.eclipse.org/> - інтегроване середовище для розроблення програм

33. <https://umbrello.kde.org> – розробник системи Umbrello.

34. institute.cn.ua – підручник з Umbrello UML Modeller.

35. <https://docs.oracle.com/javase/tutorial/> – Oracle. The Java™ Tutorials.

36. <https://www.tutorialspoint.com/java/> – Java Tutorial.

Викладач

