

Приватне акціонерне товариство
«Приватний вищий навчальний заклад
«Запорізький інститут економіки та інформаційних технологій»

Кафедра економічної кібернетики та інженерії програмного забезпечення

“ЗАТВЕРДЖУЮ”
Проректор з навчальної роботи
Д.Є.Швець
“ 30 ” серпня 2021 року

СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

ОК 22 Основи програмної інженерії

(шифр і назва навчальної дисципліни)

Освітньо-професійна програма «ІНЖЕНЕРІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ»
(назва ОПП)

спеціальність 121 Інженерія програмного забезпечення
(шифр і назва спеціальності)

спеціалізація _____
(назва спеціалізації при наявності)

Рівень вищої освіти перший (бакалаврський)
(підкреслити вірне)

2021 – 2022 навчальний рік

Мова викладання _____ українська _____

Прізвище, ім'я та по-батькові викладача(ів)/розробника(ів)

К.т.н., доц. Резніченко Юлія Сергіївна,
ст. викл. Дережа Катерина Володимирівна
(вказати викладачів їхні посади, наукові ступені та вчені звання)

Електронна адреса викладача(ів) y.reznichenko@econom.zp.ua,
k.dereza@econom.zp.ua

Дні занять за розкладом згідно з розкладом

Графік консультацій (он-лайн консультації): _____ вівторок, четвер _____

Сторінка курсу в Moodle <http://moodle.zieit.zp.ua/course/view.php?id=883>

Силабус схвалено на засіданні кафедри
економічної кібернетики та
інженерії програмного забезпечення

Протокол від "30" серпня 2021 року № 1
Завідувач кафедри ЕКІПЗ
[підпис] (підпис) Левиньшина О. (прізвище та ініціали)

Силабус погоджено
Начальник навчального відділу
[підпис] О.В.Сташкевічус

1. Опис дисципліни

Анотація дисципліни (Призначення навчальної дисципліни)	Дана дисципліна викладається для вормування цілей навчання та розуміння студентами різних аспектів майбутньої професії є необхідним компонентом підготовки кваліфікованого інженера-програміста (Software Engineer), системного архітектора (System Architect), архітектора програмного забезпечення (Software Architect).
Мета вивчення	Досягнення студентами сучасного конструктивного мислення та системи спеціальних знань у галузі програмної інженерії, прийняття рішень, теорії ризиків та здатності їх використовувати при створенні програмних продуктів.
Завдання навчальної дисципліни	Здатність складати формулювати постановку задачі для рішення завдань програмної інженерії; технічні завдання на розробку програмного забезпечення, документів на методики управління процесами створення програмних продуктів, управління ІТ-проектами; життєвим циклом програмного продукту; роботі в команді; управління ресурсами ІТ-проекту; інтеграції програмних модулів; реалізації завдань кожного етапу життєвого циклу програмного продукту.
Пререквізити	Основи програмування, Економіко-математичні методи та моделі
Результати навчання	<p>ПР01. Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки.</p> <p>ПР03. Знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення.</p> <p>ПР04. Знати і застосовувати професійні стандарти і інші нормативно-правові документи в галузі інженерії програмного забезпечення.</p> <p>ПР06. Уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення.</p> <p>ПР07. Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.</p> <p>ПР08. Вміти розробляти людино-машинний інтерфейс.</p> <p>ПР09. Знати та вміти використовувати методи та засоби збору, формулювання та аналізу вимог до програмного забезпечення.</p> <p>ПР10. Проводити передпроектне обстеження предметної області, системний аналіз об'єкта проектування.</p> <p>ПР15. Мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення.</p> <p>ПР20. Знати підходи щодо оцінки та забезпечення якості програмного забезпечення.</p> <p>ПР21. Знати, аналізувати, вибирати, кваліфіковано застосовувати засоби забезпечення інформаційної безпеки (в тому числі кібербезпеки) і цілісності даних відповідно до розв'язуваних прикладних завдань та створюваних програмних систем.</p> <p>ПР22. Знати та вміти застосовувати методи та засоби управління проектами.</p> <p>ПР24. Вміти проводити розрахунок економічної ефективності програмних систем.</p> <p>ПР26. Вміти використовувати сучасні технології та інструментальні засоби для проектування і розробки WEB-додатків.</p>

2. Обсяг та ознаки навчальної дисципліни

Найменування показників	Галузь знань, спеціальність, рівень вищої освіти	Характеристика навчальної дисципліни	
		денна форма навчання	заочна та дистанційна форма навчання
Кількість кредитів – 5	Галузь знань 12-Інформаційні технології (шифр і назва) Спеціальність 121 Інженерія програмного забезпечення (шифр і назва) Рівень вищої освіти перший	Обов'язкова	
Модулів – 2		Рік підготовки	
Змістових модулів – 2		2-й	2-й
Індивідуальне науково-дослідне завдання - не передбачено		Семестр	
Загальна кількість годин – 150		4-й	4-й
		Лекції	
		10 год.	6 год.
		Практичні, семінарські	
		40 год.	10 год.
		Лабораторні	
	Самостійна робота		
	100 год.	134 год.	
	Індивідуальні завдання:		
	Вид контролю:		
екзамен	екзамен		

3. Дидактична карта дисципліни

Назви змістових модулів і тем	Кількість годин											
	денна форма						заочна форма					
	усього	у тому числі					усього	у тому числі				
		л	п	інд.	с.р.	бали		л	п	інд.	с.р.	бали
1	2	3	4	5	6	7	8	9	10	11	12	13
Модуль 1												
Змістовий модуль 1. Загальна інформація												
Тема 1. Програмні вимоги	16	2	4		10	6	12	2	0		10	6
Тема 2. Аналіз вимог	14	0	4		10	6	16	0	2		14	6
Тема 3. Етапи розробки програмного забезпечення	14	0	4		10	6	16	0	0		16	6
Тема 4. Поняття життєвого циклу	16	2	4		10	6	14	2	0		12	6
Тема 5. Поняття життєвого циклу програмного продукту	14	0	4		10	6	16	0	2		14	6
Разом за змістовим модулем 1	74	4	20		50	30	74	4	4		66	30
Змістовий модуль 2. Деталізація проектування програмного забезпечення												
Тема 1. Моделі розробки програмного забезпечення	16	2	4		10	6	12	2	0		10	6
Тема 2. CASE-засоби	14	0	4		10	6	16	0	2		14	6
Тема 3. UML-діаграми розробки програмного забезпечення	16	2	4		10	6	16	0	2		14	6
Тема 4. Управління ресурсами в життєвому циклі програмних засобів.	14	0	4		10	6	16	0	0		16	6
Тема 5. Тестування програмного продукту	16	2	4		10	6	16	0	2		14	6
Разом за змістовим модулем 1	76	6	20		50	30	76	2	6		68	30
Модуль 2												
ІНДЗ			-	-		-			-	-	-	
Усього годин	150	10	40		100		150	6	10		134	

Самостійна робота

№ з/п	Назва теми	Кількість годин
1	Тема 1. Програмні вимоги 1. На які типи діляться програмні продукти. 2. Фундаментальні про створення ПЗ. 3. Моделі технологічного процесу створення ПЗ.	20

	<p>4. Моделі розробки ПЗ.</p> <p>5. З чим зв'язані не функціональні вимоги?</p> <p>6. На які групи можна розділити не функціональні вимоги?</p> <p>7. Які кількісні показники використовуються для не функціональних вимог?</p>	
2	<p>Тема 2. Аналіз вимог</p> <p>1. Призначення CASE-засобів.</p> <p>2. Показники якісного ПЗ. Перелічити та охарактеризувати.</p> <p>3. Основні питання, що стоять перед спеціалістами з ПЗ.</p> <p>4. Що таке та для чого використовується пілотний проект.</p> <p>5. Що входить до плану переходу?</p> <p>6. Дати визначення поняттям функціональні та не функціональні вимоги, а також проектна системна специфікація.</p> <p>7. В чому різниця користувацьких вимог від системних?</p> <p>8. Що означає, що специфікація вимог повинна бути комплексною?</p>	20
3	<p>Тема 3. Етапи розробки програмного забезпечення</p> <p>1. Які існують основні принципи інженерії програмного забезпечення?</p> <p>2. Назвіть основні види діаграм в структурному аналізі.</p> <p>3. Чому неможливо керуватися тільки базовими принципами?</p> <p>4. Моделі і процеси управління проектами програмних засобів.</p> <p>5. Системне проектування програмних засобів.</p> <p>6. Техніко-економічне обґрунтування проектів програмних засобів.</p> <p>7. Розробка вимог до програмних засобів.</p>	20
4	<p>Тема 4. Поняття життєвого циклу</p> <p>1. Планування життєвого циклу програмних засобів.</p> <p>2. Об'єктно-орієнтоване проектування програмних засобів.</p> <p>3. Управління ресурсами в життєвому циклі програмних засобів.</p> <p>4. Дефекти, помилки та ризики в життєвому циклі програмних засобів.</p>	20
5	<p>Тема 5. Поняття життєвого циклу програмного продукту та тестування</p> <p>1. З чого складається структура ЖЦ?</p> <p>2. В чому різниця між верифікацією, тестуванням та перевіркою?</p> <p>3. Що в себе включає управління конфігурацією?</p> <p>4. Які фундаментальні процеси створення ПЗ існують?</p> <p>5. Що таке модель ЖЦ? Які бувають?</p> <p>6. В чому заключається основний недолік спіральної моделі?</p>	20
	Разом	100

4. Індивідуальні завдання

Відсутні

5. Методи навчання	<p>1. Словесні (пояснення, розповідь, лекція, бесіда), наочні (ілюстрація, демонстрація), практичні вправи.</p> <p>2. Проблемно-пошукові, дослідницькі</p>
6. Система контролю та	<p>1. Оцінювання правильності виконання практичних робіт</p>

оцінювання.	<p>2. Оцінка активності студента у процесі занять, внесених пропозицій, оригінальних рішень, уточнень і визначень, доповнень попередніх відповідей і т. ін.</p> <p>3. Модульна робота</p>
7. Технічне й програмне забезпечення/обладнання.	<p>Ноутбук, персональний комп'ютер, планшет з підключенням до Інтернет, для виконання практичних завдань; виконання завдань самостійної роботи; проходження тестування (підсумковий контроль). Програмне забезпечення для роботи з освітнім контентом дисципліни та виконання передбачених видів освітньої діяльності: графічні редактори</p>
8. Політика дисципліни.	<p>1. Курс передбачає індивідуальне виконання завдання</p> <p>2. Середовище в аудиторії є дружнім, творчим, відкритим до конструктивної критики.</p> <p>3. Самостійна робота включає в себе теоретичну обробку питань, що стосуються тем лекційних занять, які не ввійшли в теоретичний курс, або ж були розглянуті коротко, їх поглиблена проробка за рекомендованою літературою</p> <p>4. Усі завдання, передбачені програмою, мають бути виконані у встановлений термін.</p> <p>5. Здобувачі вищої освіти мають право скласти екзамен автоматично, у випадку, якщо впродовж семестру такі здобувачі набрали 60-100 балів.</p> <p>6. Під час роботи над завданнями не допустимо порушення академічної доброчесності: при використанні Інтернет ресурсів та інших джерел інформації студент повинен вказати джерело, використане під час виконання завдання.</p> <p>7. Списування під час контрольних робіт та екзаменів заборонені (в т.ч. із використанням мобільних пристроїв). Мобільні пристрої дозволяється використовувати лише під час он-лайн тестування.</p> <p>8. Присутність на заняттях, а у випадку хвороби - надання справки.</p>

9. Розподіл балів, які отримують студенти

Приклад для екзамену

Поточне тестування та самостійна робота					Сума	
Змістовий модуль 1					Підсумковий тест (МК)	50
T1	T2	T3	T4	T5		
6	6	6	6	6	20	

Поточне тестування та самостійна робота					Сума	
Змістовий модуль 2					Підсумковий тест (МК)	50
T1	T2	T3	T4	T5		
6	6	6	6	6	20	

Шкала оцінювання: національна та ЄКТС

Сума балів за всі види навчальної діяльності	Оцінка за національною шкалою	
	для екзамену, курсового проекту (роботи), практики	для заліку
90 – 100	відмінно	зараховано
82-89	добре	
74-81		
64-73	задовільно	
60-63		
35-59	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
0-34	незадовільно з обов'язковим повторним вивченням дисципліни	не зараховано з обов'язковим повторним вивченням дисципліни

10. Питання до заліку чи екзамену.

Питання до екзамену

1. На які типи діляться програмні продукти.
2. Фундаментальні про створення ПЗ.
3. Моделі технологічного процесу створення ПЗ.
4. Моделі розробки ПЗ.
5. З чим зв'язані не функціональні вимоги?
6. На які групи можна розділити не функціональні вимоги?
7. Які кількісні показники використовуються для не функціональних вимог?
8. Призначення CASE-засобів.
9. Показники якісного ПЗ. Перелічити та охарактеризувати.
10. Основні питання, що стоять перед спеціалістами з ПЗ.
11. Що таке та для чого використовується пілотний проект.
12. Що входить до плану переходу?
13. Дати визначення поняттям функціональні та не функціональні вимоги, а також проектна системна специфікація.
14. В чому різниця користувацьких вимог від системних?
15. Що означає, що специфікація вимог повинна бути комплексною?
16. Які існують основні принципи інженерії програмного забезпечення?
17. Назвіть основні види діаграм в структурному аналізі.
18. Чому неможливо керуватися тільки базовими принципами?
19. Моделі і процеси управління проектами програмних засобів.

20. Системне проектування програмних засобів.
21. Техніко-економічне обґрунтування проектів програмних засобів.
22. Розробка вимог до програмних засобів.
23. Планування життєвого циклу програмних засобів.
24. Об'єктно-орієнтоване проектування програмних засобів.
25. Управління ресурсами в життєвому циклі програмних засобів.
26. Дефекти, помилки та ризики в життєвому циклі програмних засобів.
27. З чого складається структура ЖЦ?
28. В чому різниця між верифікацією, тестуванням та перевіркою?
29. Що в себе включає управління конфігурацією?
30. Які фундаментальні процеси створення ПЗ існують?
31. Що таке модель ЖЦ? Які бувають?
32. В чому заключається основний недолік спіральної моделі?
33. Поняття інформаційної системи.
34. Класифікація інформаційних систем.
35. Складність програмних систем.
36. Необхідність інженерії програмних систем.
37. Поняття проекту.
38. Керування проектом.
39. Планування проекту.
40. Моніторинг проекту.
41. Технології проектування програмних систем.
42. Життєвий цикл програмного забезпечення.
43. Стадії життєвого циклу.
44. Стандарти проектування.
45. Процеси життєвого циклу розробки систем.
46. Моделі життєвого циклу програмних систем: касадна, інкрементна, еволюційна.
47. Ітеративна розробка програмних систем.
48. Поняття про уніфікований процес розробки систем (RUP).
49. Фази та дисципліни RUP.
50. Методології проектування програмних систем.
51. Концепції (парадигми) програмування.
52. Структурний підхід до проектування.
53. Технологія структурного проектування.
54. Висхідне та низхідне проектування.
55. Об'єктно-орієнтована методологія розробки систем.

56. Основні положення об'єктної моделі. Об'єкти і класи. Відношення між класами.
57. Принципи об'єктно-орієнтованого проектування програмних систем.
58. Засоби моделювання програмних систем.
59. Моделювання бізнес-процесів.
60. Нотація BPMN.
61. Діаграми функціональної декомпозиції FDD.
62. Діаграми потоків даних DFD.
63. Уніфікована мова моделювання UML. Історія виникнення та призначення UML.
64. Основні компоненти UML: сутності, відношення, діаграми. Класифікація діаграм.
65. Інженерія вимог до програмної системи.
66. Класифікація вимог.
67. Артефакти вимог.
68. Класифікація вимог FURPS+: функціональні та нефункціональні вимоги, обмеження.
69. Традиційні та сучасні способи виявлення вимог.
70. Керування вимогами: ідентифікація, організація та специфікація вимог.
71. Технічне завдання. Призначення та структура технічного завдання.
72. Вивчення та моделювання предметної області.
73. Діаграми випадків використання (Use Case).
74. Сценарії випадків використання.
75. Візуалізація (побудова діаграм) випадків використання системи.
76. Моделювання структури системи.
77. Діаграми класів UML. Властивості класів. Атрибути та операції класів.
78. Інтерфейси. Моделювання та візуалізація інтерфейсів.
79. Взаємозв'язки між класами. Асоціації, наслідування, залежність.
80. Взаємозв'язки між класами. Агрегація та композиція.
81. Діаграми об'єктів UML.
82. Моделювання та специфікація поведінки програмних систем.
83. Діаграми взаємодії.
84. Діаграми послідовності та комунікації.
85. Діаграми діяльності.
86. Діаграми станів.
87. Поняття архітектури програмної системи.
88. Логічна та фізична архітектура системи.
89. Архітектурні стилі.
90. Архітектурний аналіз.
91. Способи виявлення класів.

92. Концептуальні класи.
93. Вибір архітектури системи.
94. Архітектурні шаблони. Принцип Model-View Separation.
95. Архітектурний шаблон Модель-Подання-Контролер (Model-View-Controller).
96. Проектування на основі обов'язків.
97. Принципи розподілу обов'язків між об'єктами.
98. Шаблони розподілу обов'язків GRASP.
99. Проектування на основі шаблонів (патернів).
100. Шаблони проектування GoF.
101. Процес розробки програмного забезпечення
102. Перетворення моделей в програмний код.
103. Керування якістю програмного забезпечення.
104. Контроль та підтримка якості.
105. Стратегії тестування.
106. Тестування програмного забезпечення. Типи тестів.
107. Модульне та функціональне тестування.
108. Розробка програмних систем на основі тестування (TDD).
109. Фреймворки для модульного тестування. Поняття про рефакторинг коду.
110. Розгортання та супровід системи. Приймальні випробування.
111. Інноваційні технології проектування систем.
112. Технології швидкої розробки програм.
113. Гнучкі методи розробки програмних систем (Agile, Scrum, XP).

11. Рекомендована література

Базова

1. Табунщик Г. В., Каплієнко Т. І., Петрова О. А. Проектування та моделювання програмного забезпечення сучасних інформаційних систем. Запоріжжя : Дике Поле, 2016. 250 с.
2. Конспект лекцій (опорний) з дисципліни: «Основи програмної інженерії» освітньо-професійної програми першого (бакалаврського) рівня вищої освіти зі спеціальності 121 – «Інженерія програмного забезпечення» / Укладачі О.О.Жулковський, І.Жулковська.– Дніпродзержинськ: ДДТУ, 2017.– 46с.
3. Уонг, Х. Основні формули і дані по теплообміну для інженерів / Х. Уонг. - М .: [не вказано], 2017. - 653 с.
4. Фомін, В.Л. Механіка континууму для інженерів / В.Л. Фомін. - М .: [не вказано], 2016. - 974 с.
5. Гецці, Карло Основи інженерії програмного забезпечення / Карло Гецці, Мехді Джазайері, Діно Мандріолі. - М .: БХВ-Петербург, 2016. - 832 с.

Допоміжна

1. Software Engineering Tutorial. – http://www.tutorialspoint.com/software_engineering/
2. Mark Reed C++: The Ultimate Beginners Guide to Learn C++ Programming. – Kindle Edition, 2021. – 258 p
3. Джефф Сазерленд, Скрам. Революционный метод управления проектами.: Пер с англ., - Манн, Иванов і Фербер. 2020. - 288