

**ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«ПРИДНІПРОВСЬКА ДЕРЖАВНА АКАДЕМІЯ БУДІВНИЦТВА ТА АРХІТЕКТУРИ»**

Кафедра комп'ютерних наук, інформаційних технологій та прикладної математики

«ЗАТВЕРДЖУЮ»
Проректор з науково-педагогічної
та навчальної роботи
Р. Б. Папірник

« 01 » _____ 2020 ____ року



СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Структура та проектування програмного забезпечення

спеціальність	122 «Комп'ютерні науки»
освітньо-професійна програма	«Комп'ютерні науки»
освітній ступінь	бакалавр
форма навчання	денна
розробник	Семенець Сергій Миколайович

1. АНОТАЦІЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Дисципліна «Структура та проектування програмного забезпечення» є варіативною компонентою циклу професійної підготовки бакалаврів за спеціальністю 122 «Комп'ютерні науки». Предметом вивчення дисципліни є структура та методи проектування програмного забезпечення (ПЗ). Вивчення даної дисципліни дає знання, що використовуються на етапі проектування ПЗ при розробці програмного дизайну.

Згідно з навчальною програмою дисципліни розглядаються наступні основні питання: структура ПЗ; системне, прикладне, сервісне, службове та інструментальне ПЗ; етапи та моделі життєвого циклу програмного продукту; управління процесом проектування ПЗ; класифікація вимог до ПЗ; атрибути якості; аналіз вимог та документування; стратегії і методи проектування ПЗ; об'єктно-орієнтоване, функціонально-орієнтоване та компонентне проектування ПЗ; архітектурні фреймворки; case- технології проектування ПЗ; принципи об'єктно-орієнтованого проектування; аналіз та оцінка якості програмного дизайну; уніфікована мова моделювання UML; діаграми UML; патерни проектування ПЗ; підвищення проектної надійності інформаційних систем.



2. ЗМІСТ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

	Години	Кредити	Семестр
			IV
Всього годин за навчальним планом, з них:	135	4,5	135
Аудиторні заняття, у т.ч:	52		52
лекції	22		22
лабораторні роботи			
практичні заняття	30		30
Самостійна робота, у т.ч:	83		83
підготовка до аудиторних занять	10		10
підготовка до контрольних заходів	3		3
виконання курсового проекту	30		30
опрацювання розділів програми, які не викладаються на лекціях	10		10
підготовка до екзамену	30		30
Форма підсумкового контролю			Екзамен

3. СТИСЛИЙ ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Мета дисципліни - теоретична та практична підготовка фахівців з питань структури і проектування програмного забезпечення інформаційних систем.

Завдання дисципліни - опанування студентами сучасними методами проектування програмного забезпечення; отримання знань про архітектуру, методи аналізу, побудови та тестування програмних засобів; отримання практичних навичок з розробки програмних продуктів.

Пререквізити дисципліни. «Інформатика», «Алгоритмізація і програмування»

Постреквізити дисципліни. «Об'єктно-орієнтоване програмування», «Проектування інформаційних систем».

Компетентності.

Інтегральна компетентність:

Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми у галузі комп'ютерних наук, або у процесі навчання, що передбачає застосування теорій та методів інформаційних технологій і характеризується комплексністю та невизначеністю умов.

Загальні компетентності:

ЗК2. Здатність застосовувати знання у практичних ситуаціях.

ЗК3. Знання та розуміння предметної області та розуміння професійної діяльності.

Спеціальні (фахові, предметні) компетентності:

СК8. Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: структурного, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами та алгоритмами обчислень, структурами даних і механізмами управління.

Заплановані результати навчання:

ПР1. Застосовувати знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу, обробки та синтезу інформації в предметній області комп'ютерних наук.

ПР2. Використовувати сучасний математичний апарат неперервного та дискретного аналізу, лінійної алгебри, аналітичної геометрії, в професійній діяльності для розв'язання задач теоретичного та прикладного характеру в процесі проектування та реалізації об'єктів інформатизації.

ПР3. Використовувати знання закономірностей випадкових явищ, їх властивостей та операцій над ними, моделей випадкових процесів та сучасних програмних середовищ для розв'язування задач статистичної обробки даних і побудови прогнозних моделей.

ПР4. Використовувати методи обчислювального інтелекту, машинного навчання, нейромережевої та нечіткої обробки даних, генетичного та еволюційного програмування для розв'язання задач розпізнавання, прогнозування, класифікації, ідентифікації об'єктів керування тощо.

ПР5. Проектувати, розробляти та аналізувати алгоритми розв'язання обчислювальних та логічних задач, оцінювати ефективність та складність алгоритмів на основі застосування формальних моделей алгоритмів та обчислюваних функцій.

ПР6. Використовувати методи чисельного диференціювання та інтегрування функцій, розв'язання звичайних диференціальних та інтегральних рівнянь, особливостей чисельних методів та можливостей їх адаптації до інженерних задач, мати навички програмної реалізації чисельних методів.

ПР7. Розуміти принципи моделювання організаційно-технічних систем і операцій; використовувати методи дослідження операцій, розв'язання одно-та багатокритеріальних оптимізаційних задач лінійного, цілочисельного, нелінійного, стохастичного програмування.

ПР8. Використовувати методологію системного аналізу об'єктів, процесів і систем для задач аналізу, прогнозування, управління та проектування динамічних процесів в макроекономічних, технічних, технологічних і фінансових об'єктах.

ПР9. Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук.

ПР10. Використовувати інструментальні засоби розробки клієнт-серверних застосувань, проектувати концептуальні, логічні та фізичні моделі баз даних, розробляти та оптимізувати запити до них, створювати розподілені бази даних, сховища та вітрини даних, бази знань, у тому числі на хмарних сервісах, із застосуванням мов веб-програмування.

ПР11. Володіти навичками управління життєвим циклом програмного забезпечення, продуктів і сервісів інформаційних технологій відповідно до вимог і обмежень замовника, вміти розробляти проектну документацію (техніко-економічне обґрунтування, технічне завдання, бізнес-план, угоду, договір, контракт).

ПР12. Застосовувати методи та алгоритми обчислювального інтелекту та інтелектуального аналізу даних в задачах класифікації, прогнозування, кластерного аналізу, пошуку асоціативних правил з використанням програмних інструментів підтримки багатовимірного аналізу даних на основі технологій DataMining, TextMining, WebMining.

ПР13. Володіти мовами системного програмування та методами розробки програм, що взаємодіють з компонентами комп'ютерних систем, знати мережні технології, архітектури комп'ютерних мереж, мати практичні навички технології адміністрування комп'ютерних мереж та їх програмного забезпечення

ПР14. Володіти мовами системного програмування та методами розробки програм, що взаємодіють з компонентами комп'ютерних систем, знати мережні технології, архітектури комп'ютерних мереж, мати практичні навички технології адміністрування комп'ютерних мереж та їх програмного забезпечення

ПР15. Застосовувати знання методології та CASE-засобів проектування складних систем, методів структурного аналізу систем, об'єктно-орієнтованої методології проектування при розробці і дослідженні функціональних моделей організаційно-економічних і виробничо-технічних систем.

ПР16. Розуміти концепцію інформаційної безпеки, принципи безпечного проектування програмного забезпечення, забезпечувати безпеку комп'ютерних мереж в умовах неповноти та невизначеності вихідних даних.

ПР17. Виконувати паралельні та розподілені обчислення, застосовувати чисельні методи та алгоритми для паралельних структур, мови паралельного програмування при розробці та експлуатації паралельного та розподіленого програмного забезпечення.

У результаті вивчення навчальної дисципліни студент повинен знати:

- структуру ПЗ;
- моделі життєвого циклу програмного продукту;
- вимоги до програмного забезпечення;
- стратегії і методи проектування програмного забезпечення;
- архітектурні фреймворки та CASE- технології проектування;
- принципи об'єктно-орієнтоване проектування;
- уніфіковану мову моделювання UML;
- критерії оцінювання якості дизайну програмного забезпечення;
- патерни проектування програмного забезпечення;
- методи підвищення проектної надійності інформаційних систем.

вміти:

- орієнтуватися в структурі сучасного програмного забезпечення;
- оцінювати якість програмного дизайну;
- аналізувати вимоги та розробляти відповідний дизайн програмного забезпечення;
- розробляти проектну документацію для опису та підтримки програмного забезпечення;
- управляти процесом проектування програм;
- проектувати програмні продукти згідно з наданим технічним завданням.

Методи навчання: словесні методи (лекція); наочні методи (ілюстрація, демонстрація); практичні методи (вправа, практична робота).

Форми навчання: фронтальні, групові, аудиторні, позааудиторні.

4. СТРУКТУРА (ТЕМАТИЧНИЙ ПЛАН) ДИСЦИПЛІНИ

Назва змістових модулів і тем	Кількість годин, у тому числі				
	усього	л	п	лаб	с/р
Змістовий модуль 1. Структура, етапи розробки та вимоги до ПЗ					
Структура програмного забезпечення.	6	4			2
Етапи розробки програмного продукту.	12	2	8		2
Вимоги до ПЗ.	12	2	8		2
Разом за змістовим модулем 1	30	8	16		6
Змістовий модуль 2. Проектування ПЗ					
Стратегії і методи проектування ПЗ.	7	2			5
Принципи об'єктно-орієнтованого проектування.	5	2			3
Уніфікована мова моделювання UML.	7	4			3
Патерни проектування.	15	4	8		3
Підвищення проектної надійності ІС .	11	2	6		3
Разом за змістовим модулем 2	45	14	14		17
Змістовий модуль 3. Курсовий проект					
Аналіз вимог та декомпозиція системи.	4				4
Обрання та освоєння патерну проектування з урахуванням вимог до системи.	4				4
Побудова діаграми варіантів використання UML.	2				2
Побудова діаграми класів UML.	4				4
Побудова діаграми послідовності UML.	4				4
Аналіз та оцінка якості програмного дизайну.	2				2

Розрахунок показників надійності системи.	2				2
Розробка моделі оптимального резервування системи з урахуванням надійності.	2				2
Розробка оптимальної структурної схеми надійності системи.	2				2
Оформлення пояснювальної записки.	4				4
Разом за змістовим модулем 3	30				30
Підготовка до екзамену	30				30
Усього годин	135	22	30		83

5. ЛЕКЦІЙНИЙ КУРС

№ занять	Тема занять	Кількість годин
1,2	Структура програмного забезпечення. Предмет та задачі дисципліни. Програмне забезпечення (ПЗ) комп'ютера. Системне ПЗ. Операційна система. Драйвери. Планування процесів. Синхронізація та обмін даними між процесами. Керування пам'яттю. Керування файлами. Системи керування вводом-виводом. Переривання. Огляд операційних систем Windows, Linux, Macintosh. Прикладне ПЗ. Браузери. Текстові і графічні редактори. Пакети презентацій. Табличні процесори, системи управління базами даних, мультимедійні програми та САПР. Службове і сервісне ПЗ. Мережеві протоколи. Утиліти. Архіватори та антивірусні програми. Інструментальне ПЗ. Редактори, компілятори та бібліотеки мов програмування. Асемблери.	4
3	Етапи розробки програмного продукту. Життєвий цикл програмного продукту (ПП). Основні етапи життєвого циклу ПП. Планування. Розробка технічного завдання (формування вимог до системи). Аналіз вимог та проектування. Етап конструювання. Інтеграція і тестування. Супровід і експлуатація. Моделі життєвого циклу ПП. Каскадна, спіральна та інкрементна моделі. Стороння розробка ПП. Заміна версії та управління конфігурацією системи.	2
4	Вимоги до ПЗ. Класифікація вимог до програмного забезпечення. Бізнес вимоги та бізнес правила. Вимоги користувача. Функціональні і не функціональні вимоги. Ефективність, надійність та безпека системи. Атрибути якості. Вимоги до даних. Вимоги до інтерфейсів. Аналіз вимог та документування.	2
5	Стратегії і методи проектування ПЗ. Загальні стратегії проектування ПЗ. Функціонально-орієнтоване (структурне) проектування ПЗ. Проектування «зверху-вниз» і «знизу-вгору». Компонентне проектування ПЗ. Об'єктно-орієнтоване проектування ПЗ. Архітектурні фреймворки. CASE - технології проектування. Нотації проектування.	2
6	Принципи об'єктно-орієнтованого проектування. Принципи SOLID як базові принципи об'єктно-орієнтованого проектування ПЗ. Принцип єдиної відповідальності. Принцип відкритості / закритості. Принцип підстановки. Принцип поділу інтерфейсу. Принцип інверсії залежностей. Інкапсуляція, абстракція, поліморфізм, спадкування та принципи SOLID. Критерії якісної та неякісної архітектури. Розширюваність, гнучкість, масштабованість, тестованість та ремонтпридатність програми. Можливість перевикористання модулів. Жорсткість, крихкість, нерухомість. Аналіз та оцінка якості програмного дизайну.	2
7,8	Уніфікована мова моделювання UML. Мова UML як засіб моделювання	4

	ПЗ. Основні будівельні блоки UML. Структурні та поведінкові сутності. Сутності, що групують та сутності, що анотують. Діаграми UML. Діаграма класів як засіб моделювання статичної структури програмної системи в термінах класів та відношень між ними. Графічне зображення класів. Атрибути та операції (методи) класу. Абстрактні класи. Інтерфейси. Умовне графічне зображення відносин залежності, асоціації, композиції, агрегації, узагальнення та реалізації. Приклади складання діаграм класів. Діаграми варіантів використання, послідовності, станів та діяльності. Інші діаграми.	
9,10	Патерни проектування. Призначення та класифікація патернів проектування. Породжуючи, структурні та поведінкові дизайн - патерни. Огляд дизайн - патернів Builder, Facade, Adapter, Decorator, Observer, Mediator.	4
11	Підвищення проектної надійності інформаційних систем. Визначення надійності ІС. Розрахунок показників надійності систем з логічно послідовним, логічно паралельним та змішаним з'єднанням елементів. Метод структурного резервування для підвищення надійності складних систем. Оцінка надійності резервованої системи.	2
	Усього годин	22

6. ТЕМИ ПРАКТИЧНИХ ЗАНЯТЬ

№ занять	Тема занять	Кількість годин
1-4	Розробка UML-діаграм.	8
5-8	Проектування уніфікованого програмного інтерфейсу.	8
9-12	Розробка програмного дизайну.	8
13-15	Проектування ІС з урахуванням надійності.	6
	Усього годин	30

7. ТЕМИ ЛАБОРАТОРНИХ ЗАНЯТЬ

Лабораторні заняття навчальним планом не передбачені.

8. САМОСТІЙНА РОБОТА

№ занять	Вид роботи / Назва теми	Кількість годин
1	Підготовка до аудиторних занять.	10
2	Підготовка до контрольних заходів.	3
3	Опрацювання розділів програми, що не викладаються на лекціях: - Мова UML як засіб моделювання програмного забезпечення. - Архітектура веб- додатків. - 4+1 View Model of Software Architecture. - Шаблони проектування.	10 2 2 3 3
4	Виконання курсового проекту: - Аналіз вимог та декомпозиція системи. - Обрання та освоєння патерну проектування з урахуванням вимог до системи. - Побудова діаграми варіантів використання UML. - Побудова діаграми класів UML. - Побудова діаграми послідовності UML. - Аналіз та оцінка якості програмного дизайну. - Розрахунок показників надійності системи. - Розробка моделі оптимального резервування системи з урахуванням	30 4 4 2 4 4 2 2 2

	надійності. - Розробка оптимальної структурної схеми надійності системи. - Оформлення пояснювальної записки.	2 4
5	Підготовка до екзамену.	30

9. МЕТОДИ КОНТРОЛЮ

Основними методами контролю знань студентів є усний, письмовий і графічний контроль, а також методи самоконтролю та самооцінки.

10. ПОРЯДОК ТА КРИТЕРІЇ ОЦІНЮВАННЯ

Структура оцінювання видів навчальної роботи студента у кожному змістовому модулі

Змістовий модуль 1. Структура, етапи розробки та вимоги до ПЗ

№ п/п	Вид навчальної роботи студента	Максимальна кількість балів
1	Виконання практичних робіт:	80 (40 балів × 2 практичні роботи)
	1. Розробка UML-діаграм.	
	2. Проектування уніфікованого програмного інтерфейсу.	
2	Контрольна робота	20 (10 балів × 2 питання)
Разом		100

Змістовий модуль 2. Проектування ПЗ

№ п/п	Вид навчальної роботи студента	Максимальна кількість балів
1	Виконання практичних робіт:	80 (40 балів × 2 практичні роботи)
	1. Розробка програмного дизайну.	
	2. Проектування ІС з урахуванням надійності.	
2	Контрольна робота	20 (10 балів × 2 питання)
Разом		100

Змістовий модуль 3. Курсовий проект

№ п/п	Вид навчальної роботи студента	Максимальна кількість балів
1	Виконання курсового проекту	
	1-й проміжний контроль:	
	Аналіз вимог та декомпозиція системи.	6
	Обрання та освоєння патерну проектування з урахуванням вимог до системи.	6
	Побудова діаграми варіантів використання UML:	6
	Побудова діаграми класів UML:	6
	Побудова діаграми послідовності UML.	6
	Усього	30
	2-й проміжний контроль:	
	Аналіз та оцінка якості програмного дизайну.	6
	Розрахунок показників надійності системи.	6
	Розробка моделі оптимального резервування системи з урахуванням надійності.	6

	Розробка оптимальної структурної схеми надійності системи.	6
	Оформлення пояснювальної записки.	6
	Усього	30
2	Захист курсового проекту	40
	Разом	100

Критерії оцінювання практичних робіт

Максимальна кількість балів за виконання однієї практичної роботи – 40. Загальна кількість практичних робіт – 4.

Кількість балів «40» – ставиться, якщо студент у відведений час повністю виконав обсяг робіт відповідно до передбаченого варіанта. Розв'язання задач виконано логічно послідовно, отримано правильні результати. Робота оформлена охайно.

Кількість балів «30–39» – ставиться, якщо студент у відведений час повністю виконав обсяг робіт відповідно до передбаченого варіанта. Розв'язання задач виконано логічно послідовно, отримано в цілому правильні результати, однак мають місце несуттєві (з точки зору програмної інженерії) помилки, робота оформлена не досить охайно.

Кількість балів «20–29» – ставиться, якщо студент у відведений час повністю виконав обсяг робіт відповідно до передбаченого варіанта. Більшість отриманих результатів є правильними, однак при розв'язанні деяких задач мають місце суттєві (з точки зору програмної інженерії) помилки, робота оформлена не досить охайно.

Кількість балів «10–19» – ставиться, якщо студент у відведений час неповністю виконав обсяг робіт відповідно до передбаченого варіанта, не всі отримані результати є правильними, робота оформлена неохайно.

Кількість балів «0–9» – ставиться, якщо студент у відведений час не виконав обсяг робіт відповідно до передбаченого варіанта, при розв'язанні задач мають місце суттєві (з точки зору програмної інженерії) помилки.

Критерії оцінювання контрольної роботи

Контрольна робота складається з 2 запитань. Максимальна кількість балів за відповідь на одне запитання кожного змістового модуля – 10 балів.

Кількість балів «10» – ставиться студенту за повну, змістовну, логічно послідовну, правильну відповідь у письмово-графічній формі на питання контрольної роботи.

Кількість балів «7–9» – ставиться студенту за логічно послідовну, загалом правильну відповідь в письмово-графічній формі на питання контрольної роботи. Але окремі пункти відповідей не повністю розкривають суть питання.

Кількість балів «3–6» – ставиться студенту за відповідь в письмово-графічній формі на питання контрольної роботи, в якій не повністю розкривається суть поставлених питань. В розв'язанні задач наявні суттєві (з точки зору програмної інженерії) помилки, що свідчать про недостатнє засвоєння студентом теоретичного та практичного матеріалу. Представлена відповідь має фрагментарний характер, слабо пов'язана з суттю поставленого питання, оформлена недбало і не дає повного уявлення про правильність кінцевих результатів.

Кількість балів «0–2» – ставиться студенту за відсутність конкретної відповіді в письмово-графічній формі на питання контрольної роботи. Відповідь носить поверхневий безсистемний характер, відсутня теоретична база у висвітленні поставленого питання, наявні грубі (з точки зору програмної інженерії) помилки, що свідчать про відсутність у студента мінімуму знань з дисципліни.

Критерії оцінювання курсового проекту

Максимальна оцінка за виконання і захист курсового проекту – 100 балів, у т.ч. виконання курсового проекту – 60 балів; захист курсового проекту – 40 балів.

Виконання курсового проекту.

Максимальна кількість балів за виконання курсового проекту – 60 балів. Кількість задач, що містить завдання на курсовий проект – 10. Максимальна оцінка за розв'язання однієї задачі – 6 балів.

Кількість балів «6» – ставиться, якщо задача розв'язана правильно і в повному обсязі.

Кількість балів «3-5» – ставиться, якщо задача розв'язана в повному обсязі. Отримано в цілому правильні результати, однак мають місце несуттєві (з точки зору програмної інженерії) помилки.

Кількість балів «0-2» – ставиться, якщо задача розв'язана неправильно або в неповному обсязі, мають місце суттєві (з точки зору програмної інженерії) помилки.

Захист курсового проекту.

Максимальна кількість балів при захисті курсового проекту – 40 балів.

Кількість балів «40» – ставиться, якщо під час захисту студент демонструє системний характер знань сутностей, принципів та методів розробки програмного дизайну, що використовувалися в проекті. Основні результати викладаються чітко і логічно-послідовно. На поставлені викладачем питання надаються повні, глибокі, обґрунтовані відповіді.

Кількість балів «30-39» – ставиться, якщо під час захисту студент показує досить високий рівень знань сутностей, принципів та методів розробки програмного дизайну, що використовувалися в проекті. Основні результати викладаються чітко і логічно-послідовно. На поставлені викладачем питання надаються в цілому правильні, обґрунтовані відповіді, однак мають місце несуттєві помилки або неточності.

Кількість балів «20-29» – ставиться, якщо під час захисту студент не показує глибоке розуміння усіх сутностей, принципів та методів розробки програмного дизайну, що використовувалися в проекті. Основні результати викладаються достатньо чітко і логічно-послідовно, але на поставлені викладачем питання надаються неповні або необґрунтовані відповіді.

Кількість балів «10-19» – ставиться, якщо під час захисту студент показує поверхневе знання сутностей, принципів та методів розробки програмного дизайну, що використовувалися в проекті. Основні результати викладаються нечітко і невпевнено. На поставлені викладачем питання надаються помилкові або необґрунтовані відповіді.

Кількість балів «0-9» – ставиться, якщо під час захисту студент не демонструє необхідне розуміння сутностей, принципів та методів розробки програмного дизайну, що використовувалися в проекті. Основні результати викладаються безсистемно, відсутня логічна послідовність викладання. Студент неспроможний надати правильні відповіді на поставлені викладачем питання.

Критерії оцінювання знань студентів на екзамені

Максимальна кількість балів на екзамені – 100 балів.

В екзаменаційному білеті 4 питання.

Максимальна кількість балів за відповідь на кожне питання – 25.

25 балів – вичерпна відповідь на запитання.

16-24 балів – ставиться за змістовну, логічно послідовну, в цілому правильну відповідь в письмовій формі на питання екзаменаційного білета. Відповідь охайно оформлено, але окремі підпункти питання розкриті не в повному обсязі, мають місце незначні (з точки зору програмної інженерії) помилки.

10-15 балів – ставиться за відповідь в письмовій формі на питання екзаменаційного білета, якщо студент надав поверхову відповідь на питання екзаменаційного білета. Допущені суттєві (з точки зору програмної інженерії) помилки, відсутня логічна послідовність відповіді.

0-9 балів – ставиться студенту при відсутності конкретної відповіді в письмово-графічній формі на екзаменаційне питання. Відповідь носить безсистемний характер,

відсутня теоретична база у висвітленні поставленого питання, наявні грубі (з точки зору програмної інженерії) помилки, що свідчить про відсутність у студента відповідних теоретичних знань.

Підсумкова оцінка з дисципліни визначається як середньоарифметичне між оцінками змістових модулів 1 і 2 та екзамену.

11. ПОЛІТИКА КУРСУ

Порядок зарахування пропущених занять:

- пропущена лекція відпрацьовується підготовкою реферату відповідно до теми пропущеного заняття та його захистом;
- пропущені практичні заняття відпрацьовуються студентами виконанням відповідної практичної роботи самостійно та її захистом.

Зміни в нарахуванні балів у випадках несвоєчасного виконання завдань не відбувається.

Дотримання академічної доброчесності студента передбачає:

- самостійне та добросовісне виконання завдань, в тому числі поточного та підсумкового контролю;
- відповідальне ставлення до своїх обов'язків;
- повага до честі й гідності інших осіб;
- посилання на джерела інформації у разі запозичення ідей, розробок, тверджень, відомостей;
- використання при виконанні завдань лише перевірених та достовірних джерел інформації.

За порушення академічної доброчесності студент може бути притягнутий до академічної відповідальності (повторне проходження оцінювання).

12. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Основна

1. Руководство Microsoft по проектированию архитектуры приложений. 2-е издание. - Корпорация Microsoft, 2012. - 529с.
2. Фаулер М. Архитектура корпоративных программных приложений. - М: Вильямс, 2016. - 541с.
3. Басс Л. Архитектура программного обеспечения на практике / Л. Басс, П. Клементс, Р. Кацман. - СПб: Питер, 2012. - 574с.
4. Гагарина Л. Г. Введение в архитектуру программного обеспечения. Учебное пособие. - М.: Инфра-М, 2016. - 320 с.
5. Панюкова Т. А. Проектирование программных средств. - М.: Гостехиздат, 2012. - 364 с.
6. Соммервилл И. Инженерия программного обеспечения / И. Соммервилл - М: Вильямс, 2012. - 467с.
7. Назаров С.В. Архитектура и проектирование программных систем. - М.: Инфра-М, 2013. - 351 с.
8. Гамма Э., Хелм Р. Приемы объектно-ориентированного программирования. Паттерны проектирования. СПб: Питер, 2014. - 366с.

Допоміжна

1. Коуд П., Норт Д., Мейфилд М. Объектные модели. Стратегии, шаблоны и приложения. - М: Лори, 2015. - 230с.
2. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на C++. - М: Бином, 2010, 560с.
3. Якобсон А. Унифицированный процесс разработки программного обеспечения. -
4. Лавріщева К.М. Програма інженерія. Підручник. - К: BHV, 2011. - 435с.

5. Мацяшек Л. Анализ требований и проектирование систем. Разработка информационных систем с использованием UML.- М: Вильямс, 2014. - 432с.
6. Gamma E. Design Patterns. СПб.: Питер, 2014. -511 с.

13. INTERNET-РЕСУРСИ

1. Компонентный подход в программировании [Электронный ресурс]. – Режим доступа: http://e.lanbook.com/books/element.php?pl1_cid=25&pl1_id=2778
2. Архитектурное проектирование программного обеспечения [Электронный ресурс]. – Режим доступа: <http://www.lib.tpu.ru/fulltext2/m/2011/m416.pdf>
3. Обзор паттернов проектирования [Электронный ресурс]. – Режим доступа: <http://www.lib.tpu.ru/fulltext2/m/2011/m310.pdf>

Розробник

(С. М. Семенець)

Гарант освітньої програми

(Н. О. Вельмагіна)

Силабус затверджено на засіданні кафедри
комп'ютерних наук, інформаційних технологій та прикладної математики

Протокол від «31» серпня 2020 року № 2