

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

Кафедра прикладної математики та інформаційних технологій



«ЗАТВЕРДЖУЮ»

Проректор з науково-педагогічної

та навчальної роботи

Р. Б. Папірник

« 15 »

2019 __ 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:	60		60
лекції	30		30
лабораторні роботи			
практичні заняття	30		30
Самостійна робота, у т.ч:	75		75
підготовка до аудиторних занять	18		18
підготовка до контрольних заходів	12		12
виконання курсового проекту або роботи			
опрацювання розділів програми, які не викладаються на лекціях	15		15
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:

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

Загальні компетентності:

ЗК-1. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК-2. Здатність застосовувати знання у практичних ситуаціях.

ЗК-3. Знання та розуміння предметної області та розуміння професійної діяльності.

ЗК-6. Здатність вчитися і оволодівати сучасними знаннями.

ЗК-7. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

ЗК-12. Здатність приймати обґрунтовані рішення.

Спеціальні (фахові, предметні) компетентності:

СК-8. 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:

РН-1. Здобувати систематичні знання в галузі комп'ютерних наук, аналізувати проблеми з точки зору сучасних наукових парадигм, осмислювати і робити обґрунтовані висновки з наукової і навчальної літератури та результатів експериментів

РН-20. 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, створювати надійне та ефективне програмне забезпечення .

РН-27. Використовувати технології проектування складних систем, вибирати CASE- засоби; формулювати техніко-економічні вимоги, розробляти інформаційні та програмні системи з використанням шаблонів та засобів автоматизованого проектування.

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.	9	6			6
Етапи розробки програмного продукту.	7	4			6
Вимоги до ПЗ.	15	4	8		6
Стратегії і методи проектування ПЗ.	14	2	8		10
Разом за змістовим модулем 1	60	16	16		28
Змістовий модуль 2. Проектування ПЗ					
Принципи об'єктно - орієнтованого проектування.	2	2			2
Уніфікована мова моделювання UML.	6	6			3
Патерни проектування.	13	4	8		6
Підвищення проектної надійності ІС .	9	2	6		6
Разом за змістовим модулем 2	45	14	14		17
Підготовка до екзамену	30				30
Усього годин	135	30	30		75

5. ЛЕКЦІЙНИЙ КУРС

№ заняття	Тема занять	Кількість годин
1,3	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.	6
4,5	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.	4
6,7	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.	4
8	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
9	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
10-12	Уніфікована мова моделювання UML. Мова UML як засіб моделювання 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.	6
13,14	Патерни проектування. Призначення та класифікація патернів проектування. Породжуючі, структурні та поведінкові дизайн-патерни.	4

	Огляд дизайн - патернів Builder, Facade, Adapter, Decorator, Observer, Mediator.	
15	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	30

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.	18
2	Підготовка до контрольних заходів.	12
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.	15 3 3 4 5
4	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

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 свідчить про відсутність у студента мінімуму знань з дисципліни.

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.

Порядок зарахування пропущених занять. Пропущені лекції та практичні заняття, незалежно від причини пропуску, студент відпрацьовує викладачеві (презентація) згідно з графіком консультацій.

11. 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

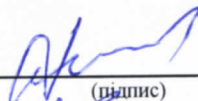
1. Коуд П., Норт Д., Мейфилд М. Объектные модели. Стратегии, шаблоны и приложения. - М: Лори, 2015. - 230с.
2. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на C++. - М: Бином, 2010, 560с.
3. Якобсон А. Унифицированный процесс разработки программного обеспечения. -
4.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

12. INTERNET-РЕСУРСИ

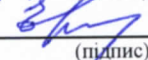
1. Компонентный подход в программировании [Электронный ресурс]. – Режим доступа: http://e.lanbook.com/books/element.php?pl1_cid=25&pl1_id=2778
2. Архитектурное проектирование программного обеспечения [Электронный ресурс]. – Режим доступа: <http://www.lib.tpu.ru/fulltext2/m/2011/m416.pdf>
3. Обзор паттернов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

Розробник


_____ (підпис)

(С. М. Семенець)

Гарант освітньої програми


_____ (підпис)

(Н.М. Єршова)

Силабус затверджено на засіданні кафедри
прикладної математики та інформаційних технологій

Протокол від «12» жовтня 2019 року № 3