

МІЖНАРОДНИЙ ЄВРОПЕЙСЬКИЙ УНІВЕРСИТЕТ
Навчально-науковий інститут «Європейська школа бізнесу»
Кафедра інформаційних технологій

ЗАТВЕРДЖЕНО
Директор ННІ
«Європейська школа бізнесу»



Юлія РЕМИГА

від « 11 » 09 2023 р.

М.П.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

**КОНСТРУЮВАННЯ ПРОГРАМНОГО
ЗАБЕЗПЕЧЕННЯ**

Рівень вищої освіти перший (бакалаврський)

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма «Інженерія програмного забезпечення»

Робоча програма навчальної дисципліни «Конструювання програмного забезпечення» складена на основі освітньо-професійної програми 121 «Інженерія програмного забезпечення» для першого (бакалаврського) рівня спеціальності 121 «Інженерія програмного забезпечення», затвердженої Вченою радою Університету «30» травня 2023 року, протокол № 4.

Укладач програми: Фаловський Олександр Олександрович, к.т.н.

Рецензент: Нестеренко Олександр Васильович, доктор технічних наук, доцент

Гарант освітньої програми:  Олександр НЕСТЕРЕНКО

Робочу програму навчальної дисципліни розглянуто та схвалено на засіданні кафедри інформаційних технологій, протокол від 31.08.2023 р. № 1.

ВСТУП

Програма вивчення навчальної дисципліни «Конструювання програмного забезпечення» складена відповідно до Стандарту вищої освіти України (далі – Стандарт) галузі знань 12 «Інформаційні технології» спеціальності 121 «Інженерія програмного забезпечення».

Опис навчальної дисципліни (анотація). Дана навчальна дисципліна є однією з дисциплін фахової підготовки майбутніх розробників програмного забезпечення.

Таблиця 1

Найменування показників	Галузь знань, напрям підготовки, освітній рівень	Характеристика навчальної дисципліни	
		денна форма навчання	заочна форма навчання
Кількість кредитів – 4	Галузь знань, 12 «ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ»	Нормативна	
Розділів – 2	Спеціальність: 121 «ІНЖЕНЕРІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ»	Рік підготовки	
Змістових розділів – 2		3	3
Індивідуальне науково-дослідне завдання:		Семестр	
		6	6
Загальна кількість годин – 120		Лекції	
		14	6
Тижневе навантаження: аудиторних – 3 самостійної роботи студента – 6	Практично-лабораторні		
	28	4	
	Самостійна робота		
	78	110	
	Освітній рівень: бакалавр	Вид контролю:	
		Екзамен	Екзамен

Предметом вивчення навчальної дисципліни є сучасні та ефективні принципи детального конструювання програмного забезпечення та формальні методи і підходи до опису процесу конструювання програмного забезпечення, а також відповідні інструментальні програмні засоби.

Міждисциплінарні зв'язки: Навчальна дисципліна пов'язана з дисциплінами «Архітектура та проектування програмного забезпечення», «Аналіз вимог до програмного забезпечення», «Алгоритми та структури даних», «Об'єктно-орієнтоване програмування», «Основи програмної інженерії».

Знання, отримані після опанування даної навчальної дисципліни, можуть бути використані при вивченні дисциплін «Моделювання програмного забезпечення», «Людино-машинна взаємодія», «Програмування інтернет-застосувань», «Якість програмного забезпечення та тестування», «Менеджмент проектів програмного забезпечення».

1. МЕТА ТА ЗАВДАННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

1.1. **Метою** викладання навчальної дисципліни «Конструювання програмного забезпечення» є отримання студентами ґрунтовної теоретичної підготовки, знань основних методичних засад та ознайомлення з інструментальними програмними засобами, що використовуються в діяльності розробників програмних засобів, забезпечення теоретичної та інженерної підготовки фахівців у галузі проектування та розробки інформаційних систем.

1.2. Основними **завданнями** вивчення дисципліни «Конструювання програмного забезпечення» є:

- оволодіння різними методологіями конструювання програмного забезпечення, з урахуванням їх переваг і недоліків;
- оволодіння прийомами конструювання програмних об'єктів;
- вивчення принципів організації взаємодії програмних об'єктів;
- сформулювати вміння працювати з конкретними інструментальними засобами для опису структури і моделювання поведінки програмного забезпечення;
- розвинути навички використання отриманих знань для розв'язання задач конструювання інформаційних систем.

1.3. **Компетентності та результати навчання**, формуванню яких сприяє дисципліна (взаємозв'язок з нормативним змістом підготовки здобувачів вищої освіти, сформульованим у термінах результатів навчання у Стандарті).

Згідно з вимогами стандарту дисципліна забезпечує набуття студентами **компетентностей**:

Таблиця 2

Інтегральна компетентність	Здатність розв'язувати складні спеціалізовані завдання або практичні проблеми інженерії програмного забезпечення, що характеризуються комплексністю та невизначеністю умов, із застосуванням теорій та методів інформаційних технологій.
Спеціальні (фахові, предметні) компетентності	СК3. Здатність розробляти архітектури, модулі та компоненти програмних систем. СК5. Здатність дотримуватися специфікацій, стандартів, правил і рекомендацій в професійній галузі при реалізації процесів життєвого циклу програмного забезпечення. СК12. Здатність здійснювати процес інтеграції системи, застосовувати стандарти і процедури управління змінами для підтримки цілісності, загальної функціональності і надійності програмного забезпечення. СК13. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення.

Інтегративні кінцеві програмні результати навчання, формуванню яких сприяє навчальна дисципліна:

Програмні результати навчання	ПР04. Знати і застосовувати професійні стандарти і інші нормативно-правові документи в галузі інженерії програмного забезпечення. ПР13. Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань. ПР17. Вміти застосовувати методи компонентної розробки програмного забезпечення.
--------------------------------------	--

Після опанування дисципліни студент повинен

знати:

- принципи конструювання програмного забезпечення;
- методи опису поведінки програмного забезпечення у відповідності до
- вимог проекту;
- методи, технології і інструментальні засоби створення програмного
- коду;
- методи побудови програмного коду, як набору сценаріїв

уміти:

- правильно виконувати процес формалізації вимог до розроблюваного
- ПЗ з боку користувача/замовника;
- перетворювати набір вимог до ПЗ у множину сценаріїв;
- розуміти процес проектування ПЗ, як повторювану сукупність кроків
- уточнення сценаріїв через моделювання поведінки ПЗ, та
- використовувати для цього інструментальні програмні засоби;
- формувати опис ПЗ, як сукупності взаємопов'язаних сценаріїв
- будувати за допомогою інструментальних програмних засобів
- прототипи для дослідження поведінки і функціональних можливостей ПЗ відповідним вимогам користувача/замовника ;
- застосовувати набуті навички конструювання ПЗ до практичних задач у прикладних сферах.

2. ІНФОРМАЦІЙНИЙ ОБСЯГ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

На вивчення навчальної дисципліни «Конструювання програмного забезпечення» відводиться 120 годин 4 кредити ЄКТС.

Розділ 1. Методи та засоби для конструювання ПЗ

ТЕМА 1. Загальна модель функціонування ПЗ

ТЕМА 2. Моделювання поведінки ПЗ.

ТЕМА 3. Сценарії поведінки. ПЗ, як множина пов'язаних сценаріїв

ТЕМА 4. Базис для опису сценаріїв поведінки ПЗ

Розділ 2. Практичне використання методів та засобів для конструювання ПЗ

ТЕМА 5. BDD-підхід для опису поведінки ПЗ

ТЕМА 6. BPMN-підхід до побудови ПЗ

ТЕМА 7. Інструментальні засоби для проектування ПЗ

3. СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Назви тем	Кількість годин				
	Всього	у тому числі			
		Лекції	Практичні/ семінари	Лабораторні	СРС
ТЕМА 1. Загальна модель функціонування ПЗ	11	1			10
ТЕМА 2. Моделювання поведінки ПЗ.	13	1		2	10
ТЕМА 3. Сценарії поведінки. ПЗ, як множина пов'язаних сценаріїв	13	1		2	10
ТЕМА 4. Базис для опису сценаріїв поведінки ПЗ	21	3		6	12
ТЕМА 5. BDD-підхід для опису поведінки ПЗ	20	2		6	12
ТЕМА 6. BPMN-підхід до побудови ПЗ	20	2		6	12
ТЕМА 7. Інструментальні засоби для проектування ПЗ	20	2		6	12
<i>Разом</i>	120	14		28	78
<i>Консультації</i>					
<i>Екзамен</i>					
Всього годин	120	14		28	78

4. ТЕМИ ЛЕКЦІЙ

№ теми	Назва теми лекції та перелік основних питань
1	Загальна модель функціонування ПЗ 1.1. Як описувати функціонування ПЗ. 1.2. Frontend та Backend ПЗ 1.3. Архітектура ПЗ. Концепції MVS, MVC, MVP. 1.4. Безкодове програмування 1.5. Функціонування ПЗ як бізнес-процесів.
2	Моделювання поведінки ПЗ. 2.1. Модель «Події»-«Реакції»-«Зв'язки» 2.2. Зв'язок по управлінню.

	2.3. Зв'язок по даних. 2.4. TDD, BDD, DDD, FDD, MDD - підходи до моделювання
3	Сценарії поведінки. ПЗ, як множина пов'язаних сценаріїв 3.1. Сценарії, як модель функціонування ПЗ. 3.2. Принцип «Подія-реакція-вплив(наслідок)» 3.3. Події користувача 3.4. Події даних 3.5. Проектування ПЗ у вигляді набору сценаріїв. Прототип ПЗ. 3.6. Задача швидкої побудови прототипу ПЗ. Перевірка коректності функціонування ПЗ на основі прототипу.
4	Базис для опису сценаріїв поведінки ПЗ 4.1. Повний цикл виконання сценаріїв. Загальні характеристики. Набір об'єктів для сценаріїв. 4.2. Повний цикл виконання сценаріїв. Загальні характеристики. Властивості об'єктів. 4.3. Повний цикл виконання сценаріїв. Функції над значеннями властивостей об'єктів 4.4. Повний цикл виконання сценаріїв. Умови виконання функцій 4.5. Формування прототипу ПЗ за методом «згори до низу» 4.6. Формування прототипу ПЗ за методом «знизу до гори»
5	BDD-підхід для опису поведінки ПЗ 5.1. Конструювання ПЗ, засноване на аналізі подій. 5.2. Об'єкти. Їх ролі, типи і взаємодія. 5.3. Швидка побудова прототипу ПЗ. 5.4. Приклад: конструювання комп'ютерної гри «Збирач коштовностей» 5.5. Приклад: конструювання додатку «Автомат із продажу напоїв»
6	ВPMN-підхід до опису ПЗ. 6.1. Business Process Model and Notation, як середовище моделювання. Загальний огляд. 6.2. Об'єкти. Зв'язки. Ролі. 6.3. Приклади опису процесу функціонування ПЗ. 6.4. Програмна реалізація
7	Інструментальні засоби для проектування ПЗ 7.1. Середовища для моделювання поведінки ПЗ. Загальний огляд. 7.2. SAMUNDA Modeler. Інтерфейс. Засоби опису бізнес-процесу. Приклади використання. Java API SAMUNDA.Інтерфейс JavaDelegate 7.3. Visio. Інтерфейс. Типи об'єктів системи. Діаграми. Приклади використання. 7.4. Twine. Інтерфейс. Графічна оболонка. Реалізація сценаріїв. Приклади використання. 7.5. Unity. Інтерфейс. Оболонка. Реалізація сценаріїв. Приклади використання. 7.6. Behat. Інтерфейс. Оболонка. Реалізація сценаріїв. Приклади використання.

5. ТЕМИ ЛАБОРАТОРНО-ПРАКТИЧНИХ ЗАНЯТЬ

№ з/п	Назва лабораторної роботи (комп'ютерного практикуму)	Кількість ауд. годин
1	Опис процесу навчання в учбовому закладі із використанням BDD-	2

	підходу.	
3	Функціональні можливості середовища розробки комп'ютерних ігор UNITY.	2
2	Розробка комп'ютерної гри «Збирач коштовностей» засобами UNITY. Загальний огляд інтерфейсу.	2
3	Розробка комп'ютерної гри «Збирач коштовностей» засобами UNITY. Опис середовища.	4
4	Розробка комп'ютерної гри «Збирач коштовностей» засобами UNITY. Поведінка героя.	2
5	Опис функціонування інтернет-магазину із використанням програмних засобів Visio	4
6	Опис комп'ютерної гри «Збирач коштовностей» засобами TWINE	4
7	Інтерактивна текстова гра «Відгадай слово». Реалізація прототипу засобами TWINE	2
9	Опис комп'ютерної гри «Збирач коштовностей» вигляді діаграми Visio	2
10	Опис комп'ютерної гри «Збирач коштовностей» як BPMN-моделі у середовищі SAMUNDA	4
	Всього	28

6. САМОСТІЙНА РОБОТА

№ з/п	Назва теми, що виноситься на самостійне опрацювання	Кількість годин СРС
1. Загальна модель функціонування ПЗ		
1	Створити модель функціонування банкомату, використовуючи MVS, MVC та MVP-підходи	10
2. Моделювання поведінки ПЗ.		
2	Створити модель функціонування банкомату, як набір стандартних сценаріїв	10
3. Сценарії поведінки. ПЗ, як множина пов'язаних сценаріїв.		
3	Розробити набір сценаріїв для актора 2d комп'ютерної гри із переміщенням .	10
4. Базис для опису сценаріїв поведінки ПЗ		
4	Розробити набір сценаріїв для відстежування змін даних для 2d комп'ютерної гри із переміщенням	12
5. BDD-підхід для опису поведінки ПЗ		
6	Побудувати засобами TWINE моделю навчання студента у ВНЗ	12
6. BPMN-підхід до побудови ПЗ		
7	Побудувати засобами BPMN модель навчання студента у ВНЗ	12
7. Інструментальні засоби для проектування ПЗ		
8	Реалізувати у засобами TWINE гру «хрестики-нолики» на полі 5x5	12
		78

7. МЕТОДИ НАВЧАННЯ

При викладанні навчальної дисципліни «Конструювання програмного забезпечення» застосовуються інформаційні та практичні методи навчання: класичні лекції та лабораторно-практичні заняття з використанням лабораторних практикумів, а також консультації з виконання самостійної роботи студентів, письмові завдання.

Методи навчально-пізнавальної діяльності: пояснювально-ілюстративний метод, репродуктивний метод, метод проблемного викладу, частково-пошуковий або евристичний метод, дослідницький метод.

Методи стимулювання й мотивації навчально-пізнавальної діяльності: індуктивні і дедуктивні методи навчання, методи стимулювання і мотивації навчання.

8. МЕТОДИ КОНТРОЛЮ

Відповідно до плану вивчення дисципліни «Конструювання програмного забезпечення» передбачається проведення поточного та підсумкового контролю.

Поточний контроль – оцінювання рівня знань, умінь та навичок осіб, які навчаються, що здійснюється в ході навчального процесу шляхом проведення письмового опитування по закінченню розділів (модульний колоквиум). Підсумковий контроль проводиться у формі екзамену.

9. ФОРМА ПІДСУМКОВОГО КОНТРОЛЮ УСПІШНОСТІ НАВЧАННЯ

Формою підсумкового контролю є **екзамен**, який складається очно (при особливій ситуації – у формі комп'ютерного тесту) в період призначений деканатом або за індивідуальним графіком, який затверджується навчальним планом.

10. СХЕМА НАРАХУВАННЯ ТА РОЗПОДІЛ БАЛІВ, ЯКІ ОТРИМУЮТЬ СТУДЕНТИ

Накопичення балів протягом семестру

№ з/п	Вид діяльності	Кількість балів за дидактичну одиницю	Кількість	Загальна кількість балів
1	Відвідування лекційних та	4	7	28

	дабораторних робіт			
2	Виконання завдань практично-лабораторних робіт	3	10	30
3	Виконання самостійних робіт	1	2	2
Максимальна оцінка				60

Загальна оцінка знань студентів за поточним контролем

Результати поточного контролю знань студентів в цілому оцінюються в діапазоні від **0** до **60** балів.

Студент допускається до підсумкового контролю за умови виконання вимог навчальної програми та у разі, якщо за поточну навчальну діяльність він набрав не менше **36** балів.

Підсумкове оцінювання знань студентів

Підсумкове оцінювання знань студентів проводиться у формі **екзамену**.

Розподіл балів оцінювання при підсумковому контролі з навчальної дисципліни

Оцінка в балах за поточне оцінювання	Оцінка за національною шкалою
35-40	Відмінно
21-34	Добре
10-20	Задовільно
менше 10	Незадовільно

Під час оцінювання відповіді на окреме питання під час екзамену додатково враховуються допущені недоліки та помилки, якими вважаються:

- неохайне оформлення роботи (не загальноприйняті скорочення, незрозумілий почерк, використання олівців замість чітких чорнил) (мінус **2** бали);
- неточності в назвах окремих термінів та понять (мінус **4** бали).

Критерії оцінювання відповіді на теоретичні питання:

1. Повна відповідь на питання, яка оцінюється **«відмінно»**, повинна відповідати таким вимогам:

- розгорнутий, вичерпний виклад змісту даної у питанні проблеми;
- повний перелік необхідних для розкриття змісту питання фізичних термінів та законів;
- здатність здійснювати порівняльний аналіз різних теорій, концепцій, підходів та самостійно робити логічні висновки й узагальнення;
- уміння користуватись методами наукового аналізу фізичних явищ, процесів і характеризувати їхні риси та форми виявлення;
- демонстрація здатності висловлення та аргументування власного ставлення до альтернативних поглядів на дане питання;

2. Відповідь на питання оцінюється **«добре»**, якщо:

– відносно відповіді на найвищий бал не зроблено розкриття хоча б одного з пунктів, вказаних вище (якщо він явно потрібний для вичерпного розкриття питання) або, якщо:

– при розкритті змісту питання в цілому правильно за зазначеними вимогами зроблені окремі помилки під час: використання формул.

3. Відповідь на питання оцінюється **«задовільно»**, якщо:

– відносно відповіді на найвищий бал не зроблено розкриття чотирьох чи більше пунктів, зазначених у вимогах до нього (якщо вони явно потрібні для вичерпного розкриття питання);

– одночасно присутні чотири чи більше типів недоліків, які окремо характеризують критерій оцінки питання;

– висновки, зроблені під час відповіді, не відповідають правильним чи загально визначеним при відсутності доказів супротивного аргументами, зазначеними у відповіді;

– характер відповіді дає підставу стверджувати, що особа, яка складає іспит, не зовсім правильно зрозуміла зміст питання чи не знає правильної відповіді і тому не відповіла на нього по суті, допустивши грубі помилки у змісті відповіді.

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
82-89	B	добре	
74-81	C		
66-73	D	задовільно	
60-65	E		
30-59	FX	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
1-29	F	незадовільно з обов'язковим повторним вивченням дисципліни	не зараховано з обов'язковим повторним вивченням дисципліни

Загальна підсумкова оцінка в балах, за національною шкалою та за шкалою ECTS заноситься до заліково-екзаменаційної відомості, навчальної картки та залікової книжки студента.

11. МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ

- робоча навчальна програма дисципліни;
- електронний курс на платформі дистанційного навчання;
- плани лекцій, практичних занять та самостійної роботи студентів;

- тези лекцій з дисципліни;
- методичні вказівки до лабораторно-практичних занять для студентів;
- методичні матеріали, що забезпечують самостійну роботу студентів;
- тестові завдання до тем лекцій;
- перелік питань до екзамену.

12. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Основна:

1. Ерік Фрімен, Елізабет Робсон Head First. Патерни проектування, 2020. 672с.
2. Ірина Бородкіна, Георгій Бородкин Інженерія програмного забезпечення. Посібник для студентів вищих навчальних закладів, 2018. 230с.
3. Браунде Э. Технология разработки программного обеспечения. Пер. з англ. Питер, 2004. 655 с.
4. Орлов С.А. Технологии разработки программного обеспечения. (Разработка сложных программных систем):учеб. пособие. Пер. з англ. Питер. 2002. 464с.
5. Мацяшек Л.А. Анализ требований и проектирование систем. Разработка информационных систем с использованием UML Пер. с англ. Изд. дом "Вильямс". 2002. 432с.
6. Соммервилл, Иан. Инженерия программного обеспечения, 6-е издание.: Пер. с англ. Издательский дом «Вильямс», 2002. 624 с.
7. Вендеров. Проектирование ПО экономических ИС. Пер. з англ. Питер, 2004. 655 с..
8. Макконнелл Стив. Совершенный код: практическое руководство по разработке программного обеспечения: пер. с англ. 2013. 869 с.

Додаткова:

9. Bohm, Corrado; and Giuseppe Jacopini (May 1966). "Flow Diagrams, Turing Machines and Languages with Only Two Formation Rules". *Communications of the ACM* 9 (5): 366–371. doi:10.1145/355592.365646
10. Dijkstra, E. W. (Aug 1972). "The Humble Programmer". *Communications of the ACM* 15 (10): 859–866. doi:10.1145/355604.361591.
<http://www.cs.utexas.edu/~EWD/transcriptions/EWD03xx/EWD340.html>. (EWD340) PDF, 1972 ACM Turing Award lecture
11. Dijkstra, E.W., "Structured Programming," *Software Engineering Techniques*, Buxton, J.N., and Randell, B., eds. Brussels, Belgium, NATO Science Committee, 1969.
12. B. Meyer, *Object-Oriented Software Construction*, second ed., Prentice Hall, 1997, Chap. 6, 10, 11.
13. K. Beck, *Test-Driven Development: By Example*, Addison-Wesley, 2002.

14. Russell Gold, Thomas Hammell, Tom Snyder. Test Driven Development: A J2EE Example.- Apress, 2005.- 296 pages.

Інформаційні ресурси :

1. Guide to the Software Engineering Body of Knowledge (SWEBOK).

CHAPTER 4. SOFTWARE CONSTRUCTION. Режим доступу:

[SWEBOK Guide - Chapter 4 \(archive.org\)](#)

2. The Camunda Platform 7 Manual

Режим доступу: [User Guide | docs.camunda.org](#)

3. A guide to the Twine user interface. Режим доступу до ресурсу:

<https://twinery.org/reference/en>

4. Unity users manual. Режим доступу:

[Unity - Manual: IMGUI Basics \(unity2d.com\)](#)

4. Behat documentation . Режим доступу:

[Behat documentation](#)