

Силабуси проходять валідацію у Kharkiv IT Cluster



Прізвище, ім'я та по батькові розробника, вчене звання, науковий ступінь, посада:
Гаманюк Ігор Михайлович, старший викладач

Назва закладу освіти: Державний університет інформаційно-комунікаційних технологій

Шифр та назва спеціальності	121 Інженерія програмного забезпечення
Назва освітньої програми	Інженерія програмного забезпечення
Назва дисципліни	Моделювання та проектування програмного забезпечення
Вид (основна, вибіркова)	Основна
Блок дисципліни (залишити блок, що відповідає дисципліні)	1. Алгоритмізація та програмування (зокрема, різні парадигми: структурне, ООП)
Кількість студентів (поточний рік)	100
Курс/Семестр	5, 6
Навчальний рік валідації	2025

ЗАГАЛЬНА ІНФОРМАЦІЯ ПРО ДИСЦИПЛІНУ

Анотація	Дисципліна Моделювання та проектування програмного забезпечення ознайомлює здобувачів освіти із моделюванням програмного забезпечення при створенні програмного забезпечення. На основі отриманих знань з моделювання програмного забезпечення, здобувач набуде практичних вмінь розробляти програмне забезпечення на основі його моделювання. Вивчення курсу передбачає теоретичну підготовку і практичне опрацювання матеріалу з використанням мережі Інтернет. Для проведення лабораторних та практичних занять використовуються комп'ютери з встановленим середовищем розробки програмного забезпечення Visual Studio та онлайн редактор UML. Онлайн редактор UML дозволяє моделювати програмне забезпечення. Середовище розробки програмного забезпечення Visual Studio дозволяє розробляти програмне забезпечення. Використовується наскрізний проект. В освітній компонент «Моделювання та проектування програмного забезпечення» імплементовано навчальний курс Крег Лармана «Введення в об'єктно-орієнтований аналіз та ітеративну розробку».
Мета навчальної дисципліни	Формування системи понять, знань, умінь і навичок з моделювання та проектування програмного забезпечення.

Силабуси проходять валідацію у Kharkiv IT Cluster

Типи занять та контролю	Лекції, практичні роботи, лабораторні роботи, самостійна робота. Підсумковий контроль – залік, іспит.								
Загальний обсяг (кредитів)	3	Лекції (занять)	0,5	Лабораторні (занять)	1	Практичні (занять)	1	Самостійна (годин)	0,5
Попередні дисципліни	Об'єктно-орієнтоване програмування C#								

МАТЕРІАЛЬНО-ТЕХНІЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДИСЦИПЛІНИ

ПК, Visual Studio, онлайн редактор UML Umletino, редактор електронних таблиць, текстовий редактор.

СТРУКТУРА ТА ЗМІСТ ДИСЦИПЛІНИ

№	Теоретична складова Назва, перелік питань або анотація лекції	Год ин	Практична складова Опис та приклад завдання, а також посилання на методичні матеріали	Годи н	Інструменти, засоби та технології
Тема 1 – Проектування інформаційних систем. Ітеративний, еволюційний та гнучкий процес					
1	Об'єктно-орієнтовний аналіз	2	Створення програмного модуля на основі типів зв'язків між сутностями один-до-одного, один-до-багатьох, багато-до-багатьох. Створення класів моделей за допомогою навігаційних властивостей.	2	
2	UML		Створення програмного модуля на основі моделі реляційних таблиць. Створення класів моделей за допомогою використання зовнішніх ключів.	2	
3	Уніфікований процес. Ітеративність та гнучкість уніфікованого процесу.				
Тема 2 – Модель предметної галузі					
4	Модель предметної галузі. Побудова моделі предметної галузі: визначення концептуальних класів, визначення атрибутів та асоціацій.	2	Скласти словник термінів з атрибутами: синоніми, опис, формат (тип, довжина, одиниці виміру), взаємозв'язок з іншими елементами, діапазон значень, правила перевірки коректності. Скласти модель предметної галузі, пов'язаної з елементом словника термінів. Створення програмного модуля на основі словника термінів з урахуванням того, що атрибути словника термінів можуть стати класом, властивістю, типом властивості в залежності від предметної галузі.	2	

Силабуси проходять валідацію у Kharkiv IT Cluster

5	Концептуальні класи. Визначення концептуальних класів методом виділення іменників з опису прецедентів та іншими методами.		Створення програмного модуля на основі моделі предметної галузі.	2	
6	Асоціації і атрибути. Визначення асоціацій та атрибутів методом виділення іменників з опису прецедентів та іншими методами, а також шляхом урахування зв'язків між сутностями.				
7	Агрегація і композиція				
8	Словник термінів				
Тема 3 – Модель прецедентів					
9	Опис прецедентів. Ролі (акторів), типи прецедентів (основні, розширення, включення).	2	Створення програмного модуля на основі високорівневих властивостей системи. Реалізація окремих сценаріїв використання як методів класів, або створення шаблонів для контролерів/сервісів.	2	
10	Шаблон прецеденту		Створення програмного модуля на основі моделі прецедентів та UML діаграми прецедентів.	2	
11	Діаграми прецедентів				
12	Списки високорівневих властивостей системи				
Тема 4 – Моделювання нефункціональних вимог					
13	Бачення системи	2	Розробка архітектурних підходів з урахуванням нефункціональних вимог. Адаптація попередньо реалізованої функціональності під нефункціональну вимогу.	2	
14	Класифікація нефункціональних вимог. Додаткова специфікація. Опис нефункціональних вимог: наскрізна функціональність, безпека, зручність інтерфейсу, надійність, продуктивність, масштабованість, доступність та інших.		Створення програмного модуля на основі документів: бачення системи, контекстної UML діаграми та додаткової специфікації.	2	
Тема 5 – Моделювання бізнес-процесів					

Силабуси проходять валідацію у Kharkiv IT Cluster

15	Бізнес-правила	2	Створення програмного модуля на основі UML діаграми діяльності.	2	
16	Діаграми видів діяльності.				
17	Діаграма станів. Порівняння реалізації процесів за діаграмою діяльності та діаграмою станів.	2	Створення програмного модуля на основі UML діаграми станів.	2	
Тема 6 – Багаторівневі застосунки. Моделювання підсистем та частин систем					
18	Роль багаторівневої архітектури (presentation, business logic, data access). Моделювання архітектури UML-діаграмами. Архітектурні шаблони. Логічна архітектура. Діаграма пакетів.	2	Створення програмного модуля на основі UML діаграми пакетів. Реалізація структури просторів імен.	2	
19	Моделювання підсистем та частин систем. Діаграми пакетів, компонентів, кооперації, композитної структури.	2	Створення програмного модуля на основі UML діаграми компонентів. Реалізація компонента у вигляді бібліотеки. Опис структури програмного модуля за допомогою діаграми композитної структури.	2	
Тема 7 – Моделювання динамічних аспектів взаємодії об'єктів					
18	Основи поведінкового моделювання систем. Взаємодія об'єктів у часі.	2	Скласти UML діаграму послідовності відповідно до варіанту завдання. Створення програмного модуля на основі складеної діаграми.	2	
19	Діаграми послідовності. Системні діаграми послідовності.				
20	Опис операцій.				
21	Діаграма комунікацій.	2	Скласти UML діаграму комунікацій відповідно до варіанту завдання. Створення програмного модуля на основі складеної діаграми.	2	
Тема 8 – Моделювання статичних аспектів взаємовідносин класів та об'єктів					
22	Моделювання статичних аспектів взаємовідносин класів.	2	Створення програмного модуля на основі UML діаграми класів.	2	
23	Моделювання статичних аспектів взаємовідносин об'єктів.	2	Створення програмного модуля на основі UML діаграми об'єктів.	2	
Тема 9 – Моделювання реалізації інформаційної системи					

Силабуси проходять валідацію у Kharkiv IT Cluster

24	Моделювання реалізації інформаційної системи. Діаграми артефактів. Моделювання коду програми. Моделювання виконуємих програм та бібліотек.	2	Створення програмного модуля на основі UML діаграм артефактів, діаграми трасування.	2	
25	Діаграма трасування. Моделювання взаємовідносин артефактів проекту.		Створення програмного модуля на основі UML діаграми розгортання.	2	
26	Діаграма розгортання. Моделювання розміщення файлів на вузлах системи.				
27	Діаграма прояву. Опис, як фізичні артефакти втілюють (реалізують) логічні компоненти системи.				
28	Діаграма мережі. Моделювання мережевої інфраструктури системи.				
Тема 10 – Представлення архітектури					
36	Представлення архітектури (набір проєкцій моделі системи) (логічне, процесів, розгортання, даних, безпеки, реалізації, розробки, прецедентів).	2	Створення програмного модуля на основі представлень архітектури (логічне, процесів, розгортання, даних, безпеки, реалізації, розробки, прецедентів) відповідно до варіанту завдань.	2	
Тема 11 – Основні принципи проєктування					
29	Основні принципи проєктування SoC, Modularity, Abstraction, Information Hiding, DRY, KISS.	2	Реалізація принципів проєктування SoC, Modularity, Abstraction, Information Hiding.	2	
30	Основні принципи проєктування DDD, TDD.	2	Реалізація принципів проєктування DRY, KISS, DDD, TDD.	2	
31	Основні принципи проєктування SOLID: SRP, OCP.	2	Реалізація принципів проєктування SOLID: SRP, OCP.	2	
32	Основні принципи проєктування SOLID: LSP, ISP, DIP.	2	Реалізація принципів проєктування LSP, ISP.	2	
			Реалізація принципу проєктування DIP.	2	
Тема 12 – Шаблони проєктування					
29	Шаблони проєктування. Породжувальні шаблони проєктування. Шаблони проєктування: Фабричний метод	2	Створення програмного модуля на основі породжувального шаблону проєктування Фабричний метод відповідно до варіанту завдання.	2	

Силабуси проходять валідацію у Kharkiv IT Cluster

	(інкапсуляція логіки створення об'єктів та забезпечення гнучкості й розширюваності системи), Абстрактна фабрика (узгоджене створення об'єктів, які мають працювати разом, зберігаючи інкапсуляцію та гнучкість архітектури).				
			Створення програмного модуля на основі породжувального шаблону проектування Абстрактна фабрика відповідно до варіанту завдання.	2	
30	Породжувальні шаблони проектування Будівельник (ізоляція логіки створення об'єкта, створення різних варіантів одного об'єкта, забезпечення контролю над послідовністю побудови об'єкта), Одинак (забезпечення існування лише одного екземпляра класу, надання глобальної точки доступу), Прототип (створення нових об'єктів шляхом копіювання вже існуючих, замість створення їх "з нуля").	2	Створення програмного модуля на основі породжувального шаблону проектування Будівельник відповідно до варіанту завдання.	2	
			Створення програмного модуля на основі породжувального шаблону проектування Прототип відповідно до варіанту завдання.	2	
			Створення програмного модуля на основі породжувального шаблону проектування Одинак відповідно до варіанту завдання.	2	
31	Структурні шаблони проектування Адаптер (узгодження інтерфейсу одного класу з інтерфейсом, який очікує інший клас), Компонувальник (організація об'єктів у деревоподібні структури), Міст (відокремлення абстракції від її реалізації).	2	Створення програмного модуля на основі структурного шаблону проектування Адаптер, Компонувальник відповідно до варіанту завдання.	2	
			Створення програмного модуля на основі структурного шаблону проектування Міст відповідно до варіанту завдання.	2	
32	Структурні шаблони проектування Декоратор (динамічне додавання нової функціональності об'єктам, не змінюючи	2	Створення програмного модуля на основі структурного шаблону проектування Декоратор, Легковаговик відповідно до варіанту завдання.	2	

Силабуси проходять валідацію у Kharkiv IT Cluster

	їхнього коду), Легковаговик (оптимізація використання пам'яті при роботі з великою кількістю однотипних об'єктів), Фасад (спрощення доступу до складних систем, об'єднуючи кілька підсистем або компонентів в один загальний інтерфейс).				
			Створення програмного модуля на основі структурного шаблону проектування Фасад відповідно до варіанту завдання.	2	
33	Поведінкові шаблони проектування Ланцюжок обов'язків (динамічне делегування запиту між обробниками, де кожен обробник вирішує: обробити запит чи передати далі), Ітератор (послідовний обхід складених об'єктів (колекцій) без розкриття їх внутрішньої структури), Посередник (спрощення архітектури шляхом централізації взаємодії між об'єктами, щоб вони не взаємодіяли напряму, а через спеціальний посередник), Відвідувач (додавання нової поведінки до об'єктів без зміни їхніх класів, реалізуючи при цьому принцип відкритості/закритості).	2	Створення програмного модуля на основі поведінкового шаблону проектування Ланцюжок обов'язків, Ітератор, Посередник, Відвідувач відповідно до варіанту завдання.	2	
34	Поведінкові шаблони проектування Команда (інкапсуляція запитів як об'єктів, що дозволяє розділити відправника запиту від його виконавця), Знімок (збереження та відновлення стану об'єкта без порушення його інкапсуляції), Стан (зміна поведінки об'єкта залежно від його внутрішнього стану).	2	Створення програмного модуля на основі поведінкового шаблону проектування Команда відповідно до варіанту завдання.	2	
35	Поведінкові шаблони проектування Шаблонний метод (визначення скелету алгоритму в базовому класі, але деякі кроки залишити змінними і реалізувати їх у похідних класах), Спостерігач (організація зв'язку "один до багатьох" між об'єктами, коли зміна стану одного об'єкта повинна	2	Створення програмного модуля на основі поведінкового шаблону проектування Знімок, Стан відповідно до варіанту завдання.	2	

Силабуси проходять валідацію у Kharkiv IT Cluster

	спричинити автоматичне оновлення всіх залежних об'єктів), Стратегія (інкапсуляція різних варіантів поведінки (алгоритмів) і динамічно вибирати потрібну стратегію під час виконання).			
			Створення програмного модуля на основі поведінкового шаблону проектування Шаблонний метод відповідно до варіанту завдання.	2
			Створення програмного модуля на основі поведінкового шаблону проектування Спостерігач відповідно до варіанту завдання.	2
			Створення програмного модуля на основі поведінкового шаблону проектування Стратегія відповідно до варіанту завдання.	2

ТЕМИ ТА ЗАВДАННЯ ДЛЯ САМОСТІЙНОЇ РОБОТИ

№	Назва теми	Опис завдання	Методи контролю та критерії оцінювання	Годин
1	Процес розробки програмного забезпечення Waterfall.	Реферат, презентація, захист	0,75 бали	6
2	Модель сутність-зв'язок.	Реферат, презентація, захист	0,75 бали	6
3	Техніки моделювання IDEF.	Реферат, презентація, захист	0,75 бали	7
4	Взаємодія об'єктів в об'єктно-орієнтованому програмуванні.	Реферат, презентація, захист	0,75 бали	7
5	Компонентне програмування.	Реферат, презентація, захист	0,75 бали	7
6	Пряме та зворотне проектування артефактів.	Реферат, презентація, захист	0,75 бали	7
7	Моделювання вбудованої системи.	Реферат, презентація, захист	0,75 бали	7
8	Графічна мова моделювання загального призначення для застосувань в системній інженерії SysML	Реферат, презентація, захист	0,75 бали	7
9	Принципи проектування DRY, KISS.	Реферат, презентація, захист	2.5	10

Силабуси проходять валідацію у Kharkiv IT Cluster

10	Шаблони проектування GRASP.	Реферат, презентація, захист	2.5	14
11	Рефакторинг.	Реферат, презентація, захист	2	10

ПРОЄКТ (за наявністю)

№	Назва та опис завдання	Метод контролю та захисту	Строки виконання
1	Завдання з практичної складової	Контроль та захист здійснюється на практичних заняттях	

РЕКОМЕНДОВАНІ ДЖЕРЕЛА ІНФОРМАЦІЇ ТА НАВЧАЛЬНІ МАТЕРІАЛИ

Основні

№	Назва	До теми (вказати номер)
1	G. Booch, J. Rumbaugh, I. Jacobson The Unified Modeling Language. User Guide / Addison-Wesley, Upper Saddle River, NJ. - 493 p.	
2	Craig Larman. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development / Publisher: Prentice Hall PTR Upper Saddle River, NJ United States. – 736 p.	
3	Патерни проектування - https://refactoring.guru/uk/design-patterns	
4	SOLID принципи - https://uk.wikipedia.org/wiki/SOLID_(об'єктно-орієнтоване_програмування)	
5	R. Martin, M. Martin. Agile Principles, Patterns, and Practices in C# / Пер. з англ. – Prentice Hall. – 768 p.	

Додаткові

№	Назва	До теми (вказати номер)
1	Уніфікована мова моделювання UML - https://www.uml-diagrams.org/	
2	OMG® Unified Modeling Language® (OMG UML®). Specification. - https://www.omg.org/spec/UML	

КОНТРОЛЬНІ ЗАХОДИ

№	Назва та опис	Методи контролю та критерії оцінювання
1	залік	60% - виконані практичні роботи + 40% - залік

Силабуси проходять валідацію у Kharkiv IT Cluster

2	екзамен	60% - виконані практичні роботи + 40% - екзамен
РЕЗУЛЬТАТИ НАВЧАННЯ		

ЗВ'ЯЗОК ІЗ РИНКОМ ПРАЦІ	
Спеціальність/професія, підготовці до діяльності в якій читається курс	Інженерія програмного забезпечення
Посилання на вакансії (понад 3), та їх скриншоти, в яких вказані компетенції, що формуються на курсі, як вимоги до кандидатів	- PNN Вакансія від 24 червня 2025 Full Stack C#/.Net Developer - CRM Genesis Full-stack Developer (C#, .NET) - Novotest C#, .NET-програміст для розробки інтерфейсу вимірювального приладу (Avalonia, Raspberry Pi) - Progresia Middle C#/.NET Developer (Microsoft Dynamics 365)
Перелік компетентностей із вказаних як вимоги до вакансії, які набувають студенти, в процесі проходження дисципліни.	- моделювання програмного забезпечення - використання патернів проектування

Інструменти оцінювання результатів навчання за дисципліною*

№	Об'єкт оцінювання (знання методів та принципів, практичні навички,	Методи контролю (тести, виконання поточних практичних завдань та їх форма: написання коду,	Інструмент оцінювання (доступ до результатів тесту, гостьова лекція, посилання
---	---	---	---

Силабуси проходять валідацію у Kharkiv IT Cluster

	командна робота тощо)	створення діаграми Гантта, створення прототипу тощо)	на виконанні завдання, посилання на проєкт, присутність на захисті проєктів, доступ до запису захисту тощо)
1	Практична робота	Наявність артефакту, що необхідно було зробити.	посилання на виконанні завдання в Google Class
2		Наявність коду, який зазнав впливу із-за цього артефакту	

***Блок. Інструменти оцінювання результатів навчання за дисципліною** — цей блок має на меті показати можливі варіанти проведення валідації за участі експертів від ІТ-компаній. Основна умова проведення валідації це попереднє погодження всіх аспектів з експертом та викладачем. Інструменти, завдяки яким це можна зробити:

- тести за темою (аналогічно технічному інтерв'ю);
- поточні завдання впродовж курсу (посилання на код на GitHub-репозитарію студента тощо) ;
- підсумковий проєкт (код, презентація, скринкаст презентації тощо);
- гостьові активності (гостьові лекції, відвідування контрольних активностей тощо).