

Міністерство освіти і науки України

Національний аерокосмічний університет ім. М.Є. Жуковського  
«Харківський авіаційний інститут»

Кафедра комп'ютерних наук та інформаційних технологій (№ 302)

**ЗАТВЕРДЖУЮ**

Гарант освітньої програми

 Ольга МАЛІСОВА  
(підпис) (ініціали та прізвище)

« 29 » \_\_\_\_\_ 08 \_\_\_\_\_ 2023 р.

**РОБОЧА ПРОГРАМА ОБОВ'ЯЗКОВОЇ  
НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

**«Управління створенням програмних продуктів»**  
(назва навчальної дисципліни)

**Галузь знань:** 12 «Інформаційні технології»

**Спеціальність:** 126 «Інформаційні системи та технології»

**Освітня програма:** «Розподілені інформаційні системи»

**Форма навчання:** денна

**Рівень вищої освіти:** перший (бакалаврський)

Харків 2023 рік

Розробник: Мирослав МОМОТ, доцент, к.т.н., доцент  
(прізвище та ініціали, посада, наукова ступінь та вчене звання)

  
(підпис)

Робочу програму розглянуто на засіданні кафедри комп'ютерних наук та інформаційних технологій

Протокол № 659/09 від « 29 » 08 2023 р.

Завідувач кафедри д.т.н., проф.



Олег ФЕДОРОВИЧ

## 1. Опис навчальної дисципліни

Найменування показників	Галузь знань, спеціальність, освітня програма, рівень вищої освіти	Характеристика навчальної дисципліни	
		денна форма навчання	
Кількість кредитів – 4,5	Галузь знань: <u>12 «Інформаційні технології»</u>  Спеціальність: <u>126 «Інформаційні системи та технології»</u>  Освітня програма: <u>«Розподілені інформаційні системи»</u>	обов'язкова	
Кількість модулів – 2		<b>Навчальний рік</b>	
Кількість змістових модулів – 2		2023/2024	
Індивідуальне завдання – РР «Побудова діаграм UML»		<b>Семестр</b>	
Загальна кількість годин – 48/135		6	<b>Лекції<sup>1)</sup></b>
Тижневих годин для денної форми навчання: аудиторних – 3; самостійної роботи студента – 5,4.	Рівень вищої освіти: перший (бакалаврський)	24 години	
		<b>Практичні, семінарські</b>	
		----	
		<b>Лабораторні<sup>1)</sup></b>	
		24 години	
		<b>Самостійна робота</b>	
		87 година	
<b>Вид контролю:</b>			
		модульний контроль, іспит	

Співвідношення кількості годин аудиторних занять до самостійної роботи становить:  
для денної форми навчання – 48/87.

<sup>1)</sup> Аудиторне навантаження може бути зменшене або збільшене на одну годину в залежності від розкладу занять.

## 2. Мета та завдання навчальної дисципліни

**Мета вивчення** – дати знання з основних напрямків та технологій комп'ютерного проектування та створення програмних продуктів.

**Завдання** – вивчити методи та інструментальні засоби комп'ютерного проектування та створення програмних продуктів з використанням сучасних програмних платформ.

### **Компетентності, які набуваються:**

КЗ 3. Здатність до розуміння предметної області та професійної діяльності.

КЗ 4. Здатність спілкуватися іноземною мовою.

КЗ 6. Здатність до пошуку, оброблення та узагальнення інформації з різних джерел.

КС 4. Здатність проектувати, розробляти та використовувати засоби реалізації інформаційних систем, технологій та інфокомунікацій (методичні, інформаційні, алгоритмічні, технічні, програмні та інші).

КС 8. Здатність управляти якістю продуктів і сервісів інформаційних систем та технологій протягом їх життєвого циклу.

КС 14. Здатність формувати нові конкурентоспроможні ідеї й реалізовувати їх у проектах (стартапах).

### **Очікувані результати навчання:**

ПР 2. Застосовувати знання фундаментальних і природничих наук, системного аналізу та технологій моделювання, стандартних алгоритмів та дискретного аналізу при розв'язанні задач проектування і використання інформаційних систем та технологій.

ПР 3. Використовувати базові знання інформатики й сучасних інформаційних систем та технологій, навички програмування, технології безпечної роботи в комп'ютерних мережах, методи створення баз даних та інтернет-ресурсів, технології розроблення алгоритмів і комп'ютерних програм мовами високого рівня із застосуванням об'єктно-орієнтованого програмування для розв'язання задач проектування і використання інформаційних систем та технологій.

ПР 5. Аргументувати вибір програмних та технічних засобів для створення інформаційних систем та технологій на основі аналізу їх властивостей, призначення і технічних характеристик з урахуванням вимог до системи і експлуатаційних умов; мати навички налагодження та тестування програмних і технічних засобів інформаційних систем та технологій.

ПР 6. Демонструвати знання сучасного рівня технологій інформаційних систем, практичні навички програмування та використання прикладних і спеціалізованих комп'ютерних систем та середовищ з метою їх запровадження у професійній діяльності.

ПР 7. Обґрунтувати вибір технічної структури та розробляти відповідне програмне забезпечення, що входить до складу інформаційних систем та технологій.

ПР 8. Застосовувати правила оформлення проектних матеріалів інформаційних систем та технологій, знати склад та послідовність виконання проектних робіт з урахуванням вимог відповідних нормативно-правових документів для запровадження у професійній діяльності.

ПР 12. Виконувати розробку програмних та апаратних засобів для створення розподілених

інформаційних систем в аерокосмічній галузі та розподіленом виробництві складної техніки (автомобілебудування, судобудування тощо).

**Пререквізити:**

- OK2. Основи програмування;
- OK4. Вступ до спеціальності;
- OK7. Створення візуальних інтерфейсів;
- OK11. Сучасні технології програмування;
- OK13. Веб-технології в розподілених інформаційних системах;
- OK14. Компонентна технологія проектування інформаційних систем;
- OK15. Крос-платформне програмування;
- OK16. Тестування інформаційних систем;
- OK18. Мобільні та хмарні технології (КР);
- OK20. Архітектура ІТ-інфраструктури підприємств;
- OK23. Системне уявлення та інтеграція інформаційних систем;
- OK24. Крос-платформне програмування (КР).

**Кореквізити:**

- OK25. Розробка веб-застосунків в інформаційних системах;
- OK26. Моделювання процесів та систем;
- OK27. Бази даних та знань в інформаційних системах;
- OK31. Створення систем штучного інтелекту та машинне навчання;
- OK32. Проектно-орієнтоване управління створенням інформаційних систем;
- OK33. Бази даних та знань в інформаційних системах (КР);
- OK34. Екологія та техногенна безпека;
- OK35. Технології захисту інформації;
- OK36. Інформаційні технології Інтернету-речей;
- OK37. Проектно-орієнтоване управління створенням інформаційних систем (КР);
- OK38. Дипломне проектування.

### 3. Зміст навчальної дисципліни

#### Модуль 1

#### Змістовий модуль 1. Технології проектування інформаційних систем, тестування ПЗ

**Тема 1. Вступ до навчальної дисципліни «Управління створенням програмних продуктів».**

Предмет, задачі та структура курсу. Зв'язок курсу з іншими дисциплінами спеціальності 126 «Інформаційні системи та технології». Список рекомендованої літератури.

**Тема 2 . Методології та технології проектування інформаційних систем.**

Складові технології проектування інформаційних систем. Вимоги до технології проектування. Представлення технологічної операції проектування.

Методологія RAD (Rapid Application Development).

Структурний підхід до проектування інформаційних систем: SADT (Structured Analysis and Design Technique), діаграми потоків даних DFD (Data Flow Diagrams), діаграми "сутність - зв'язок" ERD (Entity-Relationship Diagrams).

### **Тема 3 . Технологія клієнт/сервер. Багатоланкові клієнт/серверні системи.**

Етапи розвитку технології клієнт/сервер. Термінальні комплекси. Автономні системи. Файл-серверні мережі. Клієнт-серверні мережі. Клієнт. Сервер. Типи серверів. Переваги технології клієнт/сервер.

Спеціальна складова багатоланкових клієнт/серверні систем – сервер додатку. Створення багатоланкових інформаційних систем за допомогою C#/C++. Клієнтські додатки та їхні види в багатоланкових клієнт/серверних системах.

### **Тема 4. Вбудова контекстно-чуттєвої довідкової системи у програмне забезпечення.**

Створення спливаючої підказки за допомогою C#/C++. Процес створення вбудованої контекстно-чуттєвої довідкової системи у програмне забезпечення. Налаштування контекстно-чуттєвої довідкової системи у програмному забезпеченні.

### **Модульний контроль.**

## **Змістовий модуль 2. Діаграми UML, CASE – засоби створення ПЗ**

### **Тема 5. Системи контролю версій проекту ПЗ.**

Методи організації роботи в колективі розробників ПЗ. Системи контролю версій проекту ПЗ (Version Control System – VCS). Керування релізами програмного забезпечення.

Локальні системи контролю версій. Централізовані системи контролю версій: TeamSource, Microsoft Visual SourceSafe, Team Foundation Server. Розподілені системи контролю версій (Distributed Version Control System — DVCS): Git, GitHub.

### **Тема 6. Основи UML. Діаграми варіантів використання (use case diagrams) в UML.**

Історія створення мови UML. Огляд літератури з UML. Основні поняття та діаграми мови UML. Призначення діаграм варіантів використання. Дійова особа або актор. Варіант використання або прецедент. Позначення на діаграмах варіантів використання.

Використання діаграм варіантів використання для опису вимог до програмного забезпечення.

### **Тема 7. Діаграми класів (class diagrams).**

Призначення діаграм класів. Атрибути. Операції. Області видимості.

Відносини між класами: асоціація, спадкування, використання, агрегація, інстанцування, бета класи. Асоціація як семантичний зв'язок між класами. Потужність зв'язку. Спадкування. Одиночне та множинне спадкування. Використання. Клієнт та сервер при використанні. Агрегація. Фізичне включення як складова форма агрегації. Інстанцування при використанні шаблонів. Графічні позначення на діаграмах класів.

### **Тема 8. Дінамічні діаграми в UML**

Діаграми поведінки системи (behavior diagrams): діаграми взаємодії: діаграми

послідовності (sequence diagrams) та діаграми кооперації/ комунікації (collaboration/communication diagram) в UML.

Призначення діаграм взаємодії об'єктів. Графічні позначення на діаграмах послідовності дій. Графічні позначення на кооперативних діаграмах. Порівняння діаграм послідовності дій та кооперативних діаграм.

Діаграми взаємодії UML 2.0: діаграми огляду взаємодії (interaction overview diagram), часові діаграми або діаграми синхронізації (timing diagrams).

#### **Тема 9. Діаграми становищ та діаграми діяльностей в UML.**

Діаграми поведінки системи (behavior diagrams): діаграми становищ (statechart diagrams) та діаграми діяльностей (activity diagrams) в UML.

Графічні позначення на діаграмах становищ та діаграмах діяльностей. Відмінності діаграм становищ від інших діаграм поведінки системи: діаграм послідовності та кооперативних діаграм.

Відмінності діаграм діяльностей від іншої діаграми поведінки системи - діаграми становищ.

#### **Тема 10. Діаграми реалізації в UML.**

Діаграми реалізації (implementation diagrams): діаграми компонентів (component diagrams) і діаграми розміщення (deployment diagrams) в UML.

Графічні позначення на діаграмах компонентів. Компоненти, зв'язки між ними.

Графічні позначення на діаграмах розміщення. Прилади, процесори, зв'язки між ними, розміщення процесів між приладами та процесами.

Структурні діаграми UML 2.0: діаграми композитної/складеної структури (composite structure diagram).

Приклади діаграм реалізацій складних систем.

#### **Тема 11. CASE – засіб Rational Rose для створення програмного продукту.**

Поняття CASE – засоби створення програмних продуктів. Класифікація CASE – засобів для створення ПЗ. UpperCASE. MiddleCASE. Інтегровані CASE – засоби.

Огляд CASE – засобів. Основні особливості CASE – засобів, область застосування і порівняльна характеристика. Основні характеристики Rational Rose. Особливості застосування Rational Rose для різних учасників процесу проектування ПЗ.

#### **Тема 12. Зворотне проектування в Rational Rose.**

Застосування CASE – засобу Rational Rose для створення складних і дуже складних програмних систем та проектів. Утиліта Rational Rose C++ Analyzer, її настройки.

Перспективи розвитку технологій створення програмних продуктів.

**Модульний контроль.**

## **Модуль 2**

**Індивідуальне завдання (PP)**

## 4. Структура навчальної дисципліни

Назви змістових модулів і тем	Кількість годин				
	денна форма				
	усього	у тому числі			
л		п	Лаб	с.р.	
<b>Модуль 1</b>					
<b>Змістовий модуль 1. Технології проектування інформаційних систем, тестування ПЗ</b>					
Тема 1. Вступ до навчальної дисципліни «Управління створенням програмних продуктів».	2	2	-	-	-
Тема 2. Методології та технології проектування інформаційних систем.	16	4	-	2	10
Тема 3. Технологія клієнт/сервер. Багатоланкові клієнт/серверні системи.	17	4	-	3	10
Тема 4. Вбудова контекстно-чутливої довідкової системи у програмне забезпечення.	15	2	-	3	10
<b>Модульний контроль</b>	2	2	-	-	-
<b>Усього годин</b>	<b>52</b>	<b>14</b>	<b>-</b>	<b>8</b>	<b>30</b>
<b>Змістовий модуль 2. Діаграми UML, CASE – засоби створення ПЗ</b>					
Тема 5. Системи контролю версій проекту ПЗ.	11	1	-	6	4
Тема 6. Основи UML. Діаграми варіантів використання (use case diagrams) в UML.	5	1	-	-	4
Тема 7. Діаграми класів (class diagrams).	5	1	-	-	4
Тема 8. Дінамічні діаграми в UML	5	1	-	-	4
Тема 9. Діаграми становищ та діаграми діяльності в UML.	5	1	-	-	4
Тема 10. Діаграми реалізації в UML.	5	1	-	-	4
Тема 11. CASE – засіб Rational Rose для створення програмного продукту.	11	1	-	6	4
Тема 12. Зворотне проектування в Rational Rose.	8	1	-	4	3
<b>Модульний контроль</b>	2	2	-		
<b>Усього годин</b>	<b>57</b>	<b>10</b>	<b>-</b>	<b>16</b>	<b>31</b>
<b>Модуль 2</b>					
Індивідуальне завдання (PP)	26	-	-	-	26
<b>Усього з дисципліни</b>	<b>135</b>	<b>24</b>	<b>-</b>	<b>24</b>	<b>87</b>



### 5. Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
1	Створення інформаційних систем підтримки БД	2
2	Створення багатоланкових інформаційних систем підтримки БД за допомогою C#/C++	3
3	Створення контекстно-чутливої вбудованої довідкової системи програмного продукту)	3
4	Вивчення веб-сервісу GitHub – DVCS для спільної розробки в колективі розробників ПЗ	6
5	Побудова діаграм UML засобами Rational Rose	6
6	Зворотне проектування в Rational Rose	4
	Разом	<b>24</b>

### 6. Самостійна робота

№ з/п	Назва теми	Кількість годин
1	Діаграми "сутність - зв'язок" ERD (Entity-Relationship Diagrams)	10
2	Клієнтські додатки та їхні види в багатоланкових клієнт/серверних системах	10
3	Налагодження контекстно-чутливої довідкової системи у програмному забезпеченні	10
4	Особливості централізованої системи контролю версій проекту ПЗ TeamSource	4
5	Використання діаграм варіантів використання для опису вимог до програмного забезпечення	4
6	Графічні позначення на діаграмах класів	4
7	Порівняння діаграм послідовності дій та кооперативних діаграм	4
8	Відмінності діаграм діяльностей від іншої діаграми поведінки системи - діаграми становищ	4
9	Приклади діаграм реалізацій складних систем	4
10	Особливості застосування Rational Rose для різних учасників процесу проектування ПЗ	4
11	Утиліта Rational Rose C++ Analyzer, її настройки	3
12	Виконання РР	26
	Разом	<b>87</b>

### 7. Індивідуальні завдання

Виконання РР за варіантами на тему «Побудова діаграм UML».

### 8. Методи навчання

Проведення аудиторних лекцій, лабораторних занять, індивідуальні консультації (при необхідності), самостійна робота студентів за матеріалами, опублікованими кафедрою (методичні посібники).

### 9. Методи контролю

Проведення поточного контролю, письмового модульного контролю, фінальний контроль у вигляді іспиту.

## 10. Критерії оцінювання та розподіл балів, які отримують здобувачі

### 10.1. Розподіл балів, які отримують студенти (кількісні критерії оцінювання)

Складові навчальної роботи	Бали за одне заняття (завдання)	Кількість занять (завдань)	Сумарна кількість балів
<b>Змістовний модуль 1</b>			
Виконання і захист лабораторних робіт	0...10	3	0...30
Модульний контроль	0...15	1	0...15
<b>Змістовний модуль 2</b>			
Виконання і захист лабораторних робіт	0...10	3	0...30
Модульний контроль	0...15	1	0...15
Виконання і захист РР	0...10	1	0...10
<b>Усього за семестр</b>			<b>0...100</b>

Семестровий контроль (іспит) проводиться у разі відмови студента від балів поточного тестування й за наявності допуску до іспиту. Під час складання семестрового іспиту студент має можливість отримати максимум 100 балів.

Білет для іспиту складається з трьох запитань: двох теоретичних (максимальна кількість балів за повну та правильну відповідь на одне запитання - 30) та одного практичного (максимальна кількість балів - 40).

### 10.2. Якісні критерії оцінювання

Необхідний обсяг знань для отримання позитивної оцінки:

- апарат об'єктно-орієнтованої методології розробки програмного забезпечення (ПЗ);
- основи UML мови визначення, представлення, проектування та документування

програмних систем, бізнес-систем та інших систем різної природи;

- інтегровані середовища програмування, що дозволяють створювати програмне забезпечення в рамках об'єктно-орієнтованої парадигми програмування;
- методи реалізації задачі з використанням об'єктно-орієнтованої парадигми програмування;
- методи організації роботи в колективах розробників складного ПЗ.

Необхідний обсяг вмінь для отримання позитивної оцінки:

- вибрати технологію та інструментальні засоби розробки, складання, налагодження, випробування і документування програм на об'єктно-орієнтованих мовах для задач обробки числової та символічної інформації;
- створювати якісний програмний продукт;
- робити тестування, налагодження та документування програм.

### 10.3 Критерії оцінювання роботи студента протягом семестру

**Задовільно (60-74).** Показати мінімум знань та умінь. Відпрацювати та захистити всі лабораторні роботи та РР. Вміти читати UML-діаграми, які описують програмні системи, будувати недеталізовані UML-діаграми. Знати методології та технології проектування інформаційних систем, процеси управління створенням програмного забезпечення (ПЗ).

**Добре (75 - 89).** Твердо знати мінімум знань, виконати усі завдання. Показати вміння виконувати та захищати всі лабораторні роботи в обумовлений викладачем строк з обґрунтуванням рішень та заходів, які запропоновано у роботах. Вміти застосовувати мову UML для визначення, представлення, проектування та документування програмних систем, а також робити тестування, налагодження та документування програм. Знати можливості сучасних CASE-засобів для створення ПЗ.

**Відмінно (90 - 100).** Повно знати основний та додатковий матеріал. Знати усі теми. Орієнтуватися у підручниках та посібниках. Досконально знати системи контролю версій (СКВ), застосовувати розподілені СКВ для організації роботи в колективах розробників складного ПЗ. Вміти застосовувати CASE-засоби для створення ПЗ. Безпомилково виконувати та захищати всі лабораторні роботи в обумовлений викладачем строк з докладним обґрунтуванням рішень та заходів, які запропоновано у роботах.

### Шкала оцінювання: бальна і традиційна

Сума балів	Оцінка за традиційною шкалою	
	Іспит, диференційований залік	Залік
90 – 100	Відмінно	Зараховано
75 – 89	Добре	
60 – 74	Задовільно	

0 – 59	Незадовільно	Не зараховано
--------	--------------	---------------

### 11. Методичне забезпечення

1. Момот М.О., Кулик Ю.О. Технологія створення програмних продуктів. Частина 1. Навчальний посібник до лабораторного практикуму. Харків : Нац. аерокосм. ун-т ім. М. Є. Жуковського «Харків. авіац. ін-т», 2021. – 88 с.
2. Момот М.О., Кулик Ю.О. Технологія створення програмних продуктів. Частина 2. Навчальний посібник до лабораторного практикуму. Харків : Нац. аерокосм. ун-т ім. М. Є. Жуковського «Харків. авіац. ін-т», 2022. – 60 с.
3. Навчально-методичне забезпечення дисципліни «Управління створенням програмних продуктів. [Електронний ресурс] – Режим доступу: [http://library.khai.edu/library/fulltexts/doc/Upravlinnya\\_Sstvorenyam%20.pdf](http://library.khai.edu/library/fulltexts/doc/Upravlinnya_Sstvorenyam%20.pdf)
4. Електронні матеріали «Управління створенням програмних продуктів» [Електронний ресурс] – Режим доступу: <https://mentor.khai.edu/course/view.php?id=1308>

### 12. Рекомендована література

#### Базова

1. Alan Dennis, David Paul Tegarden, Barbara Haley Wixom. Systems Analysis and Design: An Object-Oriented Approach with UML– Wiley, 2015. – 544 p.
2. Grady Booch, James Rumbaugh, Ivar Jacobson. The Unified Modeling Language User Guide: 2nd edition. – Addison-Wesley Professional, 2017. – 504 p.
3. Martin Fowler. UML Distilled: A Brief Guide to the Standard Object Modeling Language. - 3rd edition. – Addison-Wesley Professional, 2003. – 208 p.
4. І.М.Дудзяний. Об'єктно-орієнтоване моделювання програмних систем: Навчальний посібник. – Львів: Видавничий центр ЛНУ імені Івана Франка, 2007. - 108 с.
5. Grady Booch, James Rumbaugh, Ivar Jacobson. The Unified Modeling Language User Guide: 2nd edition. – Addison-Wesley Professional, 2017. – 504 p.

#### Допоміжна

1. Bjarne Stroustrup. Programming: Principles and Practice Using C++: 2nd edition. – Addison-Wesley Professional, 2014 –1312 p.
2. Raul Sidnei Wazlawick. Object-Oriented Analysis and Design for Information Systems: Modeling with UML, OCL, and IFML. – Morgan Kaufmann, 2014. – 376 p.
3. Alan Dennis, David Paul Tegarden, Barbara Haley Wixom. Systems Analysis and Design: An Object-Oriented Approach with UML– Wiley, 2015. –? 544 p.
4. Grady Booch. Object-Oriented Analysis and Design with Applications. – Addison-Wesley Professional, 2011. – 608 p.

### 13. Інформаційні ресурси

1. Основы UML. [Электронный ресурс] – Режим доступа: <https://docs.kde.org/trunk5/uk/umbrello/umbrello/uml-basics.html>
2. The Unified Modeling Language. [Электронный ресурс] – Режим доступа: <https://www.uml-diagrams.org/>
3. GitHub с GitHub Desktop. [Электронный ресурс] – Режим доступа: <https://betacode.net/10283/use-github-with-github-desktop>