

Міністерство освіти і науки України

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

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

ЗАТВЕРДЖУЮ

Гарант освітньої програми


(підпис)
Олександр ПРОХОРОВ
(підписати та прийняти)

« 29 » 08 2023 р.

**РОБОЧА ПРОГРАМА ОБОВ'ЯЗКОВОЇ
НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

«Технологія створення програмних продуктів»
(назва навчальної дисципліни)

Галузь знань: 12 «Інформаційні технології»

Спеціальність: 122 «Комп'ютерні науки»

Освітня програма: «Комп'ютеризація обробки інформації та управління»

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.
(науковий ступінь і вчене звання)



(підпис)

Олег ФЕДОРОВИЧ
(ініціали та прізвище)

1. Опис навчальної дисципліни

| Найменування показників | Галузь знань, спеціальність, освітня програма, рівень вищої освіти | Характеристика навчальної дисципліни | |
|--|--|--------------------------------------|--|
| | | денна форма навчання | |
| Кількість кредитів – 4,5 | <p>Галузь знань: <u>12 «Інформаційні технології»</u></p> <p>Спеціальність: <u>122 «Комп'ютерні науки»</u></p> <p>Освітня програма: <u>«Комп'ютеризація обробки інформації та управління»</u></p> | обов'язкова | |
| Кількість модулів – 2 | | Навчальний рік | |
| Кількість змістових модулів – 2 | | 2023/2024 | |
| Індивідуальне завдання – РР «Побудова діаграм UML» | | Семестр | |
| Загальна кількість годин – 48/135 | | 6 | |
| Тижневих годин для денної форми навчання: аудиторних – 3; самостійної роботи студента – 5,4. | Рівень вищої освіти: перший (бакалаврський) | 24 години | |
| | | Практичні, семінарські | |
| | | ---- | |
| | | Лабораторні¹⁾ | |
| | | 24 години | |
| | | Самостійна робота | |
| | | 87 годин | |
| Вид контролю: | | | |
| модульний контроль, іспит | | | |

Співвідношення кількості годин аудиторних занять до самостійної роботи становить:

для денної форми навчання – 48/87.

¹⁾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:

ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК4. Здатність спілкуватися державною мовою як усно, так і письмово.

ЗК5. Здатність спілкуватися іноземною мовою.

ЗК6. Здатність вчитися й оволодівати сучасними знаннями.

ЗК12. Здатність оцінювати та забезпечувати якість виконуваних робіт.

СК3. Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв'язності та нерозв'язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем.

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

СК9. Здатність реалізувати багаторівневу обчислювальну модель на основі архітектури клієнт-сервер, включаючи бази даних, знань і сховища даних, виконувати розподілену обробку великих наборів даних на кластерах стандартних серверів для забезпечення обчислювальних потреб користувачів, у тому числі на хмарних сервісах.

СК10. Здатність застосовувати методології, технології та інструментальні засоби для управління процесами життєвого циклу інформаційних і програмних систем, продуктів і сервісів інформаційних технологій відповідно до вимог замовника.

СК13. Здатність до розробки мережевого програмного забезпечення, що функціонує на основі різних топологій структурованих кабельних систем, використовує комп'ютерні системи і мережі передачі даних та аналізує якість роботи комп'ютерних мереж.

СК14. Здатність застосовувати методи та засоби забезпечення інформаційної безпеки, розробляти й експлуатувати спеціальне програмне забезпечення захисту інформаційних ресурсів об'єктів критичної інформаційної інфраструктури.

СК16. Здатність реалізовувати високопродуктивні обчислення на основі хмарних сервісів і технологій, паралельних і розподілених обчислень при розробці й експлуатації розподілених систем паралельної обробки інформації.

СК17. Здатність до розробки програмного забезпечення для задач управління об'єктами та процесами у реальному часі.

СК 18. Здатність розробляти й експлуатувати спеціальне програмне забезпечення для об'єктів та процесів аерокосмічної галузі.

Очікувані результати навчання:

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.

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).

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.

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.

ПР18. Виконувати розробку інструментальних засобів та програмного забезпечення для управління складними системами та процесами у реальному часі.

Пререквізити:

- ОК2. «Основи програмування»;
- ОК11. «Об'єктно-орієнтоване програмування»;
- ОК14. «Компонентна технологія проектування комп'ютерних систем»;
- ОК15. «Крос-платформне програмування»;
- ОК16. «Тестування програмних систем».

Кореквізити:

- ОК25. «Розробка веб-застосунків»;
- ОК26. «Моделювання систем»;
- ОК27. «Розробка баз даних та знань».

3. Зміст навчальної дисципліни**Модуль 1****Змістовий модуль 1. Технології проектування інформаційних систем, тестування ПЗ****Тема 1. Вступ до навчальної дисципліни «Технологія створення програмних продуктів».**

Предмет, задачі та структура курсу. Зв'язок курсу з іншими дисциплінами спеціальності. 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. Области видимості.

Відносини між класами: асоціація, спадкування, використання, агрегація, інстанцування, 7ета 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. | |
| Модуль 1 | | | | | |
| Змістовий модуль 1. Технології проектування інформаційних систем, тестування ПЗ | | | | | |
| Тема 1. Вступ до навчальної дисципліни «Технологія створення програмних продуктів». | 2 | 2 | - | - | - |
| Тема 2. Методології та технології проектування інформаційних систем. | 16 | 4 | - | 2 | 10 |
| Тема 3. Технологія клієнт/сервер. Багатоюанкові клієнт/серверні системи. | 17 | 4 | - | 3 | 10 |
| Тема 4. Вбудова контекстно-чуттєвої довідкової системи у програмне забезпечення. | 15 | 2 | - | 3 | 10 |
| Модульний контроль | 2 | 2 | - | - | - |
| Усього годин | 52 | 14 | - | 8 | 30 |
| Змістовий модуль 2. Діаграми UML, CASE – засоби створення ПЗ | | | | | |
| 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 |
| Модульний контроль | 2 | 2 | - | | |
| Усього годин | 57 | 10 | - | 16 | 31 |
| Модуль 2 | | | | | |
| Індивідуальне завдання (РР) | 26 | - | - | - | 26 |
| Усього з дисципліни | 135 | 24 | - | 24 | 87 |

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-чуттєвої 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 | 24 |

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-чуттєвої 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 | 87 |

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 |
|---------------------------------------|---------------------------------|----------------------------|-------------------------|
| Змістовний модуль 1 | | | |
| Виконання і захист лабораторних робіт | 0...10 | 3 | 0...30 |
| Модульний контроль | 0...15 | 1 | 0...15 |
| Змістовний модуль 2 | | | |
| 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 |
| Усього за семестр | | | 0...100 |

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.

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 програмного забезпечення (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.
2. Навчально-методичне забезпечення дисципліни «Технологія створення програмних продуктів. [Електронний ресурс] – Режим доступу: http://library.khai.edu/library/fulltexts/doc/TehnoL_Stvorennya.pdf

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