

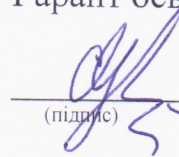
Міністерство освіти і науки України

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

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

ЗАТВЕРДЖУЮ

Гарант освітньої програми


(підпис)

О. В. Малєєва
(ініціали та прізвище)

« ____ » _____ 2021 р.

**РОБОЧА ПРОГРАМА ОБОВ'ЯЗКОВОЇ
НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

«Сучасні технології програмування»
(назва навчальної дисципліни)

Галузь знань: 12 «Інформаційні технології»

Спеціальність: 126 «Інформаційні системи та технології»

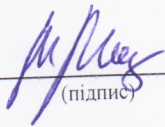
Освітня програма: «Розподілені інформаційні системи»

Форма навчання: денна

Рівень вищої освіти: перший (бакалаврський)

Харків 2021 рік

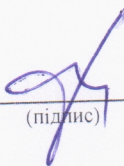
Розробник: Момот М.О., доцент, к.т.н., доцент
(прізвище та ініціали, посада, наукова ступінь та вчене звання)


(підпис)

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

Протокол № 634/08 від « 30 » серпня 2021 р.

Завідувач кафедри д.т.н., проф.
(науковий ступінь і вчене звання)


(підпис)

О.Є. 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 – 6,5	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		Навчальний рік
Кількість змістових модулів – 2		2021/2022
Індивідуальне завдання: РР		Семестр
Загальна кількість годин – 80/195		3
Тижневих годин для денної форми навчання: аудиторних – 5 годин самостійної роботи студента – 7,2 години	Рівень вищої освіти: перший (бакалаврський)	Лекції¹⁾
		48 годин
		Практичні, семінарські

		Лабораторні¹⁾
		32 годин
		Самостійна робота
		115 годин
		Вид контролю:
		модульний контроль, іспит

Співвідношення кількості годин аудиторних занять до самостійної роботи становить: для денної форми навчання – 80/115.

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

2. Мета та завдання навчальної дисципліни

Мета – дати концептуальні положення, а також вивчення методів та принципів об'єктно-орієнтованого програмування для створення сучасних програмних продуктів.

Завдання – навчити студентів використовувати в практичній діяльності можливості об'єктно-орієнтованого програмування при створенні комп'ютерних систем

Компетентності, які набуваються:

- здатність до 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 (ФКЗ);

- 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 (ФК7);

- здатність до інтелектуального багатовимірної аналізу даних та їхньої оперативної аналітичної обробки з візуалізацією результатів аналізу в процесі розв'язування прикладних задач у галузі комп'ютерних наук (ФК9);

- 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 (ФК12);

- здатність управляти якістю продуктів і сервісів як складових інформаційно-управляючих систем на основі використання сучасних підходів та інструментальних засобів тестування програмного забезпечення (ФК13).

Очікувані результати навчання:

- проектувати, розробляти та аналізувати алгоритми розв'язання обчислювальних та логічних задач, оцінювати ефективність та складність алгоритмів на основі застосування формальних моделей алгоритмів та обчислюваних функцій (ПРН4);

- 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 (ПРН7);

- Застосовувати знання методології та CASE-засобів проектування складних систем, методів структурного аналізу систем, об'єктно-орієнтованої методології проектування в процесі побудови і практичного застосування функціональних моделей організаційно-економічних і виробничо-технічних систем (ПРН11);

- володіти мовами системного програмування та методами розробки програм, що взаємодіють з компонентами комп'ютерних систем, демонструвати знання мережних технологій, архітектури комп'ютерних мереж і практичні навички технології адміністрування комп'ютерних мереж та їх програмного забезпечення (ПРН12);

- забезпечувати ефективне управління якістю продуктів і сервісів як складових інформаційно-управляючих систем (ПРН17);

- забезпечувати еволюційне удосконалення структури програмного забезпечення інформаційної системи з використанням методів рефакторингу (ПРН19);

- розробляти, використовувати та впроваджувати вбудовані системи, призначені для роботи в реальному часі (ПРН21).

Пререквізити:

- «Основи програмування»;

- «Структуризація інформації в управлінні».

Кореквізити

- «Розробка мобільних додатків»;

- «Основи тестування інформаційних систем»;

- «Управління створенням програмних продуктів».

3. Зміст навчальної дисципліни

Модуль 1

Змістовий модуль 1. Основи об'єктно-орієнтованого програмування

Тема 1. Вступ до навчальної дисципліни «Сучасні технології програмування».

Предмет, задачі та структура курсу. Зв'язок курсу з іншими дисциплінами спеціальності "Основи програмування та алгоритмічні мови", "Алгоритмізація та програмування",

"Програмування інформаційних управляючих систем", "Крос-платформне програмування". Список рекомендованої літератури.

Тема 2 . Роль декомпозиції, абстракції та ієрархії при розробці складного програмного забезпечення.

Складність, властива програмному забезпеченню. Організована та неорганізована складність. Канонічна форма складної програмної системи. Алгоритмічна та об'єктно-орієнтована декомпозиція. Роль декомпозиції, абстракції та ієрархії при розробці програмного забезпечення.

Тема 3 . Покоління мов програмування. Парадигми програмування.

Покоління мов програмування. Топологія мов різного покоління. Структурна та процедурна парадигми програмування. Об'єктно-орієнтована парадигма програмування. Класифікація об'єктно-орієнтованих мов програмування.

Тема 4. Структури та класи в C++/C#.

Класи. Структури як класи. Характерні особливості, відмінності структур в C, C++, C#, а також записів в Pascal\Delphi. Виділення та визволення пам'яті. Особливості об'яви конструкторів та деструкторів та перевантаження конструкторів в C++. Функції-члени та дані-члени.

Тема 5. Інтерфейси та реалізації класів в C++. Області видимості.

Показчик this. Приховання даних. Розділи класів. Друзі класу. Статичні члени. Вкладені класи.

Тема 6. Перевантаження операцій у мові C++/C#.

Бінарні й унарні операції. Наперед визначені значення операцій. Визначене перетворення типу. Обмеження на перевантаження операцій, особливості у мовах C++/C#. Перевантаження операцій в Delphi.

Модульний контроль.

Змістовий модуль 2. Взаємодія класів та об'єктів, шаблони

Тема 7. Похідні класи. Побудова ієрархій класів та поліморфізм.

Особливості об'яви похідних класів. Одиночне та множинне спадкування. Поля типу. Дерево та граф спадкування ієрархій класів. Віртуальні функції. Поліморфна поведінка класів, які зв'язані спадкуванням. Чисті віртуальні функції. Абстрактний базовий клас.

Тема 8. Обробка виняткових ситуацій.

Обробка помилок в стандартному C. Поняття винятку, виняткової ситуації стосовно програмного забезпечення. Оператори try, catch, throw. Вкладення блоків try-catch. Перехоплення виняткової ситуації. Оброблювач виняткової ситуації. Повторна генерація виняткової ситуації. Особливості обробки виняткових ситуацій у мовах C++/C#.

Тема 9. Шаблони функцій та класів.

Шаблони функцій, їх переваги та недоліки у порівнянні з макросами. Об'ява та виклик шаблонної функції. Шаблони класів. Об'ява та використання шаблонів класів. Параметри шаблонів. Інстаціювання. Нетипові параметри шаблонів. Особливості узагальнень у мові C#.

Тема 10. Узагальнені колекції в C# (System.Collections.Generic), бібліотека стандартних шаблонів STL (C++).

Огляд шаблонів стандартної бібліотеки STL (Standard Template Library), а також узагальнених колекцій C# (System.Collections.Generic). Контейнерні типи, колекції, ітератори, черги, стеки.

Тема 11. Об'єктний підхід. Об'єктно-орієнтовані аналіз, проектування та програмування.

Складові об'єктного підходу: абстрагування, інкапсуляція, модульність, ієрархія, слабка та сильна типізація, паралелізм, зберегаємість. Поняття об'єктно-орієнтованого аналізу (ООА), проектування (OOD) та програмування (OOP). Взаємозв'язок об'єктно-орієнтованого аналізу, проектування та програмування та їх використання при розробці програмного забезпечення.

Тема 12. Об'єкти. Відносини між об'єктами.

Об'єкт як основа ООП. Стан та поведінка об'єкта, ідентичність об'єктів. Зв'язок між об'єктами – передача повідомлень. Ролі об'єктів: актор, сервер та агент.

Тема 13. Класи. Відносини між класами.

Різні розуміння поняття інтерфейсу, розділи інтерфейсу класу. Взаємозв'язок класів та об'єктів.

Асоціація як семантичний зв'язок між класами. Потужність зв'язку. *Спадкування*. Одиночне та множинне спадкування. Графічне позначення асоціації та спадкування. Приклади асоціації та спадкування. *Використання*. Клієнт та сервер при використанні. *Агрегація*. Фізичне включення як складова форма агрегації. Графічне позначення агрегації та використання. Приклади агрегації та використання. *Інстанцування* при використанні шаблонів. Графічне позначення інстанцування та *метакласів*. Приклади інстанцування та метакласів. Поняття класифікації як виділення класів та об'єктів предметної області. Проблеми класифікації.

Тема 14. Якість програмного продукту.

Класичні міри якості програмного продукту. Залежність кількості знайдених помилок від часу. Щільність помилок. Якість класів та об'єктів. Об'єктно-орієнтовані міри якості програмного продукту. Зважена насиченість класу методами. Глибина дерева спадку. Число нащадків. Зачеплення класів. Відклик на клас. Нестача зв'язності у методах

Тема 15. Інструменти розробки програмного забезпечення.

Набір інструментів попередніх поколінь мов програмування – редактор, компілятор, компонувальник, завантажувач.

Набір інструментів об'єктно-орієнтованих мов програмування – графічний редактор, браузер класів, інкрементний компілятор, налагоджувач, компонувальник, завантажувач, інструменти управління конфігурацією та контролю версій проекту ПЗ, генератор графічного інтерфейсу користувача.

Модульний контроль.

Модуль 2

Індивідуальне завдання (РР)

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. Основи об'єктно-орієнтованого програмування					
Тема 1. Вступ до навчальної дисципліни «Сучасні технології програмування».	1	1	-	-	-
Тема 2. Роль декомпозиції, абстракції та ієрархії при розробці складного програмного забезпечення	13	3	-	-	10
Тема 3. Покоління мов програмування. Парадигми програмування	13	3	-	-	10
Тема 4. Структури та класи в C++/C#	21	6	-	5	10
Тема 5. Інтерфейси та реалізації класів в C++. Області видимості	18	3	-	5	10
Тема 6. Перевантаження операцій у мові C++/C#	22	6	-	6	10
Модульний контроль	2	2	-	-	-
Усього годин	90	24	-	16	50
Змістовий модуль 2. Взаємодія класів та об'єктів, шаблони.					
Тема 7. Похідні класи. Побудова ієрархій класів та поліморфізм	14	3	-	6	5
Тема 8. Обробка виняткових ситуацій	14	4	-	5	5
Тема 9. Шаблони функцій та класів	14	4	-	5	5
Тема 10. Узагальнені колекції в C# (System.Collections.Generic), бібліотека стандартних шаблонів STL (C++)	6	1	-	-	5
Тема 11. Об'єктний підхід. Об'єктно-орієнтовані аналіз, проектування та програмування	7	2	-	-	5
Тема 12. Об'єкти. Відносини між об'єктами	7	2	-	-	5
Тема 13. Класи. Відносини між класами	14	4	-	-	10
Тема 14. Якість програмного продукту	6	1	-	-	5
Тема 15. Інструменти розробки програмного забезпечення	6	1	-	-	5
Модульний контроль	2	2	-	-	-
Усього годин (модуль 2)	90	24	-	16	50
Модуль 2					
Індивідуальне завдання (РР)	15	-	-	-	15
Усього з дисципліни	195	48	-	32	115

5. Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
1	Використання структур у C++/C# та записів в Delphi.	5
2	Використання класів в C++/C# та Delphi. Конструювання і знищення об'єктів. Динамічне виділення та визволення пам'яті.	5
3	Перевантаження операторів.	6
4	Вивчення спадкування в ООП.	6
5	Обробка виняткових ситуацій.	5
6	Шаблони функцій та класів.	5
	Разом	32

6. Самостійна робота

№ з/п	Назва теми	Кількість годин
1	Організована та неорганізована складність. Канонічна форма складної програмної системи.	10
2	Структурна та процедурна парадигми програмування.	10
3	Характерні особливості, відмінності структур в C, C++,C#, а також записів в Pascal/Delphi .	10
4	Вкладені класи.	10
5	Обмеження на перевантаження операцій у мові C++/C#.	10
6	Чисті віртуальні функції. Абстрактний базовий клас.	5
7	Повторна генерація виняткової ситуації.	5
8	Нетипові параметри шаблонів.	5
9	Контейнерні типи, колекції, ітератори, черги, стеки	5
10	Взаємозв'язок об'єктно-орієнтованого аналізу (ООА), проектування (OOD) та програмування (ООР) та їх використання при розробці програмного забезпечення.	5
11	Ролі об'єктів: актор, сервер та агент.	5
12	Поняття класифікації як виділення класів та об'єктів предметної області. Проблеми класифікації.	10
13	Класичні міри якості програмного продукту. Залежність кількості знайдених помилок від часу. Щільність помилок.	5
14	Набір інструментів попередніх поколінь мов програмування – редактор, компілятор, компоновальник, завантажувач.	5
15	Виконання РР	15
	Разом	115

7. Індивідуальні завдання

Виконання РР за варіантами на тему «Побудова діаграм взаємодії класів і об'єктів».

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			
Робота на лекціях	0...1	5	0...5
Виконання і захист лабораторних робіт	0...10	3	0...30
Модульний контроль	0...10	1	0...10
Змістовний модуль 2			
Робота на лекціях	0...1	5	0...5
Виконання і захист лабораторних робіт	0...10	3	0...30
Модульний контроль	0...10	1	0...10
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 одержання позитивної оцінки:

- основи об'єктно-орієнтованого аналізу, об'єктно-орієнтованого проектування;
- основні положення об'єктно-орієнтованого програмування, методи, способи та засоби розробки програм у рамках цього напрямку;
- сучасні методи та засоби розробки алгоритмів та програм, способи запису алгоритму мовою високого рівня, способи налагодження, випробування та документування програм;
- методи проектування та виробництва програмного продукту, принципи побудови, структури і прийоми роботи з інструментальними засобами, що підтримують створення програмного забезпечення.

Необхідний обсяг вмінь для одержання позитивної оцінки:

- формулювати проблему, ставити задачі та цілі;
- робити об'єктно-орієнтований аналіз предметної області;
- робити декомпозицію задачі у вигляді окремих класів та об'єктів, а також взаємозв'язків між ними;
- реалізовувати спроектовані класи у вигляді закінченого програмного забезпечення;
- працювати з інструментальними засобами, що підтримують створення програмного забезпечення в рамках об'єктно-орієнтованої парадигми програмування;
- 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 застосовувати основи об'єктно-орієнтованого аналізу, знати основні положення об'єктно-орієнтованого програмування, методи, способи та засоби розробки програм у рамках цього напрямку. Вміти писати програмний код структур та класів.

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 робити об'єктно-орієнтований аналіз предметної області, робити декомпозицію задачі у вигляді окремих класів та об'єктів, а також взаємозв'язків між ними. Вміти пояснювати складні алгоритми, які реалізовані в програмах.

Відмінно (90 - 100). Повно знати основний та додатковий матеріал. Знати усі теми. Орієнтуватися у підручниках та посібниках. Досконально знати складові об'єктно-орієнтованого програмування, які використовуються при розробці складних програм. Вміти реалізовувати спроектовані ієрархії класів у вигляді закінченого програмного забезпечення; знати сучасні методи та засоби розробки алгоритмів та програм, способи запису алгоритму мовою високого рівня, способи налагодження, випробування та документування програм. 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. Кулик Ю.А., Момот М.А. Объектно-ориентированное программирование. Часть 1. Учебное пособие по лабораторному практикуму. Харьков: Нац. аэрокосмический ун-т "Харьк. авиац. ин-т", 2005. - 54 с.

2. Кулик Ю.А., Момот М.А. Объектно-ориентированное программирование. Часть 2. Учебное пособие по лабораторному практикуму. Харьков: Нац. аэрокосмический ун-т "Харьк. авиац. ин-т", 2006. - 54 с.

3. Навчально-методичне забезпечення дисципліни «Сучасні технології програмування». [Електронний ресурс] – Режим доступу: http://library.khai.edu/library/fulltexts/doc/Suchasni_Tehnologi_Programuvannya.pdf.

12. Рекомендована література

Базова

1. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений, 3-е изд. - М.: ООО "Вильямс", 2008. - 720 с.

2. Троелсен Э. - Язык программирования C# 5.0 и платформа .NET 4.5, 6-е изд. - М.: ООО "Вильямс", - 2013. - 1311 с.

3. Васильев А.Н. C#. Объектно-ориентированное программирование : учебный курс — М.; СПб. [и др.] : Питер, 2012. — 315 с.

4. Страуструп Б. Программирование. Принципы и практика с использованием C++. - М.: ООО "Вильямс", - 2015. - 1348 с.
5. Павловская Т.А. C#. Программирование на языке высокого уровня. - СПб.: Питер, 2014. - 432 с.

Допоміжна

1. Элиенс А. Принципы объектно-ориентированной разработки программ. М.: Вильямс, 2002. - 494 с.
2. Маклафлин Б, Поллайс Г, Уэст Д. Объектно-ориентированный анализ и проектирование. - СПб.: Питер, 2013. - 608 с.
3. Фридман А.Л. Основы ООП на C++ - М.: Радио и связь, Горячая линия-Телеком, 1999. - 208 с.
4. Коуд П., Норт Д., Мейфилд М. Объектные модели. Стратегии, шаблоны и приложения. - М.: Лори, 1999 - 434 с.

13. Інформаційні ресурси

1. Объектно ориентированное программирование (C#). [Электронный ресурс] – Режим доступа: <https://docs.microsoft.com/ru-ru/dotnet/csharp/programming-guide/concepts/object-oriented-programming>
2. Классы. Объектно-ориентированное программирование. [Электронный ресурс] – Режим доступа: <https://metanit.com/sharp/tutorial/3.1.php>