

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА

ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ

Кафедра інформаційних систем

«ЗАТВЕРДЖУЮ»

Заступник декана
з навчальної роботи

_____ Кашпур О.Ф.

«___» _____ 2018 року

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
ОСНОВИ ОБ'ЄКТНО-ОРІЄНТОВАНОГО
ПРОГРАМУВАННЯ

для студентів

галузь знань **12 «Інформаційні технології»**
(шифр і назва)
спеціальність **121 «Інженерія програмного забезпечення»**
(шифр і назва спеціальності)
освітній рівень **бакалавр**
(молодший бакалавр, бакалавр, магістр)
освітня програма **«Програмна інженерія»**
(назва освітньої програми)

вид дисципліни **обов'язкова**

Форма навчання **денна**
Навчальний рік **2018/2019**

Семестр **3, 4**

Кількість кредитів ECTS **8**

Мова викладання, навчання
та оцінювання **українська**

Форма заключного контролю **екзамен**

Викладачі: **к.ф.-м.н., доц. Жереб К.А.** (лекції, лабораторні заняття),
к.ф.-м.н., доц. Петрушенко А.М. (лабораторні заняття),
к.ф.-м.н., асистент Терлецький Д.О. (лабораторні заняття)

Пролонговано: на 20__/20__ н.р. _____ (_____) «__» 20__ р.
(підпис, ПІБ, дата)

на 20__/20__ н.р. _____ (_____) «__» 20__ р.
(підпис, ПІБ, дата)

КИЇВ – 2018

Розробник: Жереб Костянтин Анатолійович, к.ф.-м.н., доцент кафедри «Інформаційних систем»

ЗАТВЕРДЖЕНО

В.о. зав. кафедри «Інформаційних систем»

_____ (Іванов Є.О.)
(підпис) (прізвище та ініціали)

Протокол № 8 від «22» травня 2018 р.

Схвалено науково-методичною комісією факультету комп'ютерних наук та кібернетики

Протокол від «18» червня 2018 року №__

Голова науково-методичної комісії _____ (Хусаїнов Д.Я.)
(підпис) (прізвище та ініціали)

«_____» _____ 20__ року

1. Мета дисципліни – формування у студентів основ знань, необхідних для розуміння принципів організації, функціонування і проектування об'єктно-орієнтованих програмних систем; опанування навичками проектування об'єктно-орієнтованих систем, зокрема з використанням графічної нотації UML; практичне опанування навичками програмування на мові C/C++, зокрема робота з функціями та вказівниками, об'єктами та класами, шаблонами, використання можливостей стандартних бібліотек.

2. Попередні вимоги до опанування або вибору навчальної дисципліни (за наявності):

1. *Знати:* основні поняття програмування та принципи розробки програм; базові алгоритми та структури даних.
2. *Вміти:* проектувати, розробляти та тестувати програми на базовому рівні.
3. *Володіти елементарними навичками:* програмування мовами C, C++.

3. Анотація навчальної дисципліни:

Навчальна дисципліна “Основи об'єктно-орієнтованого програмування” є складовою освітньо-професійної програми підготовки фахівців за першим (*бакалаврським*) рівнем вищої освіти *галузі знань* 12 „Інформаційні технології” зі *спеціальності* 121 „Інженерія програмного забезпечення”, *освітньо-професійної програми* „Програмна інженерія”.

Дана дисципліна є обов'язковою навчальною дисципліною за *програмою* “**Програмна інженерія**”.

Викладається у **3 та 4 семестрах 2 курсу в обсязі – 240 год.**

(8 кредитів ECTS) зокрема: *лекції – 60 год., лабораторні – 56 год., консультації – 4 год., самостійна робота – 120 год.* У курсі передбачено **4 змістових модулі** та **4 модульні контрольні роботи**. Завершується дисципліна – **екзаменом в 4 семестрі**.

В результаті вивчення навчальної дисципліни студент повинен:

знати принципи організації та функціонування об'єктно-орієнтованих програмних систем; основні етапи життєвого циклу програмного забезпечення; основні методи та підходи до проектування об'єктно-орієнтованих програмних систем, зокрема використання патернів проектування; основні принципи об'єктно-орієнтованого програмування, зокрема інкапсуляцію, успадкування та поліморфізм.

вміти створювати програми з використанням мови програмування C/C++, використовувати засоби стандартних бібліотек мови C/C++; використовувати сучасні інструментальні засоби розробки, зокрема компілятори, інтегровані середовища розробки (IDE), сучасні засоби модульного тестування (unit testing), сучасні засоби командної розробки, зокрема системи контролю версій (revision control); проектувати об'єктно-орієнтовані програмні системи з використанням графічної нотації UML; знаходити та усувати дефекти в програмному коді.

Для допуску до дисципліни „Основи об'єктно-орієнтованого програмування” освітньо-професійної програми «Програмна інженерія» студент повинен опанувати компетентності та результати навчання, які надають дисципліни „Основи програмування” та „Програмування” програми «Програмна інженерія». Дисципліна „Основи об'єктно-орієнтованого програмування” є базовою для засвоєння дисциплін «Об'єктно-орієнтоване програмування», «Java-технології», «Міжплатформне програмування» програми «Програмна інженерія».

4. Завдання (навчальні цілі):

набуття знань, умінь та навичок (компетентностей) на рівні новітніх досягнень у програмуванні, відповідно до кваліфікації фахівця з інформаційних технологій. Зокрема, розвивати:

- здатність до абстрактного мислення, аналізу та синтезу;
- здатність застосовувати знання у практичних ситуаціях;
- здатність працювати в команді;
- визначеність і наполегливість щодо поставлених завдань і взятих обов'язків;
- володіння знаннями про інформаційні моделі даних та системи, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних;
- здатність до алгоритмічного та логічного мислення;
- здатність реалізувати сучасні методи побудови та аналізу ефективних алгоритмів в конкретних застосуваннях.

5. Результати навчання за дисципліною:

Результат навчання (1. знати; 2. вміти; 3. комунікація; 4. автономність та відповідальність)		Форми (та/або методи і технології) викладання і навчання	Методи оцінювання та пороговий критерій оцінювання (за необхідності)	Відсоток у підсумковій оцінці з дисципліни
Код	Результат навчання			
PH1.1	<i>Знати принципи організації та функціонування об'єктно-орієнтованих програмних систем</i>	<i>Лекція, лабораторне заняття</i>	<i>Тест, 60% правильних відповідей, екзамен</i>	8%
PH1.2	<i>Знати основні етапи життєвого циклу програмного забезпечення</i>	<i>Лекція, лабораторне заняття</i>	<i>Тест, 60% правильних відповідей, екзамен</i>	8%
PH1.3	<i>Знати основні методи та підходи до проектування об'єктно-орієнтованих програмних систем, зокрема використання патернів проектування</i>	<i>Лекція, лабораторне заняття</i>	<i>Тест, 60% правильних відповідей, екзамен</i>	16%
PH1.4	<i>Знати основні принципи об'єктно-орієнтованого програмування, зокрема інкапсуляцію, успадкування та поліморфізм</i>	<i>Лекція, лабораторне заняття</i>	<i>Тест, 60% правильних відповідей, екзамен</i>	8%
PH2.1	<i>Вміти створювати програми з використанням мови програмування C/C++, використовувати засоби стандартних бібліотек мови C/C++.</i>	<i>Лабораторне заняття, самостійна робота</i>	<i>Захист лабораторної роботи, екзамен</i>	10%
PH2.2	<i>Вміти використовувати сучасні інструментальні засоби розробки, зокрема компілятори, інтегровані середовища розробки (IDE), сучасні засоби модульного тестування (unit testing), сучасні засоби командної розробки, зокрема системи контролю версій (revision control)</i>	<i>Лабораторне заняття, самостійна робота</i>	<i>Захист лабораторної роботи, екзамен</i>	10%
PH2.3	<i>Вміти проектувати об'єктно-орієнтовані програмні системи з використанням графічної нотації UML</i>	<i>Лабораторне заняття, самостійна робота</i>	<i>Захист лабораторної роботи, екзамен</i>	10%
PH2.4	<i>Вміти знаходити та усувати дефекти в програмному коді</i>	<i>Лабораторне заняття, самостійна робота</i>	<i>Захист лабораторної роботи, екзамен</i>	10%

PH3.1	Обґрунтовувати власний погляд на задачу, спілкуватися з колегами з питань проектування та розробки програм, писати технічну документацію програм	Лабораторне заняття	Поточне оцінювання, захист ЛР	10%
PH4.1	Організувати самостійну роботу для вивчення нових технологій	Самостійна робота	Поточне оцінювання, Захист лабораторної роботи	5%
PH4.2	Відповідально ставитися до виконуваних робіт, нести відповідальність за їх якість	Лабораторна робота	Захист лабораторної роботи	5%

6. Співвідношення результатів навчання дисципліни із програмними результатами навчання

Результати навчання дисципліни Програмні результати навчання	PH1.1	PH1.2	PH1.3	PH1.4	PH2.1	PH2.2	PH2.3	PH2.4	PH3.1	PH4.1	PH4.2
	(з опису освітньої програми)										
ПР-3. Знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення.	+	+									
ПР-5. Знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізів та математичного моделювання для розробки програмного забезпечення.	+		+								+
ПР-6. Уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення.						+					+
ПР-12. Знати ефективні підходи щодо проектування програмного забезпечення.			+				+				
ПР-13. Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.				+	+						
ПР-16. Мати навички командної розробки, погодження, оформлення і випуску всіх видів програмної документації.		+							+		
ПР-20. Знати підходи щодо оцінки та забезпечення якості програмного забезпечення.								+			
ВПР-2. Аналізувати, оцінювати і вибирати інструментальні та обчислювальні засоби, технології, алгоритмічні і програмні рішення для розв'язання завдань інженерії програмного забезпечення.										+	
ВПР-3. Знати та мати навички реалізації основних алгоритмів та структур даних програмування.					+	+					

7. Схема формування оцінки.

7.1 Форми оцінювання студентів:

- семестрове оцінювання:

1. Контрольна робота (тест) 1: РН 1.1, РН 1.2 — 10 балів/6 балів.
2. Контрольна робота (тест) 2: РН1.3, РН 1.4 - 10 балів/6 балів.
3. Лабораторна робота 1 (проект): РН1.1, РН2.1, РН3.1 – 10 балів/6 балів.
4. Лабораторна робота 2 (проект): РН1.2, РН2.1, РН2.2, РН3.1 – 10 балів/6 балів.
5. Лабораторна робота 3 (проект): РН1.3, РН2.3, РН3.1 – 10 балів/6 балів.
6. Лабораторна робота 4 (проект): РН1.4, РН2.4, РН3.1 – 10 балів/6 балів.

- підсумкове оцінювання (у формі екзамену) вказується:

- максимальна кількість балів які можуть бути отримані студентом: 40 балів;
- результати навчання які будуть оцінюватись: РН1.1, РН1.2, РН1.3, РН2.3;
- форма проведення і види завдань: письмова.

Види завдань: 4 письмових завдання.

Критерії оцінювання на екзамені

Завдання	Тема завдання	Максимальний відсоток від 40 балів	Всього відсотків
Завдання 1	Письмове запитання з принципів ООП	25%	25%
Завдання 2	Письмове запитання з патернів	25%	25%
Завдання 3	Письмове запитання з проектування програм	25%	25%
Завдання 4	Письмове запитання з графічної нотації UML	25%	25%
			100%

Запитання для підготовки до екзамену

1. Визначення класу. Дані-члени класу.
2. Функції члени класу. Доступ до членів класу.
3. Функції члени класу з кваліфікаторами const та volatile. Визначення mutable.
4. Закриті та відкриті функції-члени.
5. Неявний вказівник this. Використання вказівника this.
6. Статичні члени класу. Статичні функції члени класу.
7. Вказівник на члени класу. Робота з вказівниками на члени класу.
8. Об'єднання. Бітове поле.
9. Поняття області видимості класу. Розв'язання імен в області видимості класу.
10. Вкладені класи. Розв'язання імен в області видимості вкладеного класу.
11. Ініціалізація класу. Конструктор класу.
12. Конструктор за замовченням. Обмеження прав на створення об'єкта.
13. Копіювальний конструктор. Деструктор класу. Явний виклик деструктора.
14. Масиви та вектори об'єктів Ініціалізація масиву в кучі. Почлена ініціалізація.
15. Перевантаження операторів.
16. Operator=, operator[], operator(), operator++, operator new, operator delete.
17. Перетворення визначені користувачем.

18. Конвертори.
19. Конвертер як конструктор.
20. Визначення шаблону класу.
21. Конкретизація шаблону класу. Аргументи шаблону для параметрів констант.
22. Функції – члени шаблонів класів. Друзі в шаблонах класів.
23. Статичні члени в шаблонах класів. Вкладені типи шаблонів класів. Шаблони – члени.
24. Наслідування. Визначення ієрархії класів.
25. Ідентифікація членів ієрархії. Визначення базового класу. Визначення похідних класів.
26. Доступ до членів базового класу. Конструювання базового та похідного класу. Конструктор базового класу. Конструктор похідного класу.
27. Деструктор. Віртуальні функції в базовому та похідному класі.
28. Чисто віртуальні функції. Статичний виклик віртуальної функції. Віртуальні деструктори.
29. Множинне успадкування. Відкрите, закрите та захищене успадкування.
30. Успадкування та композиція. Композиція об'єктів.
31. Віртуальне успадкування. Визначення віртуального базового класу.
32. Семантика ініціалізації. Порядок викликів конструкторів та деструкторів.
33. Видимість членів віртуального базового класу.
34. Ідентифікація типів під час виконання. Оператор `dynamic_cast`.
35. Оператор `typeid`. Клас `type_info`. Виключення.
36. Оператор `try {} catch {}`. Об'єкти виключення. Виключення та успадкування.
37. Розподіл вимог по суб'єктам та прецедентам. Діаграма прецедентів.
38. Діаграма класів. Асоціація, агрегація, композиція використання та наслідування.
39. Діаграма послідовностей. Діаграма взаємодії.
40. Діаграма компонентів. Діаграма розгортання.

Студент не допускається до екзамену, якщо під час семестру набрав менше ніж 20 балів. Студент допускається до екзамену за умови виконання 70% передбачених планом лабораторних робіт.

7.2 Організація оцінювання:

Терміни проведення форм оцінювання:

1. Контрольна робота (тест): до 7 тижня семестру.
2. Контрольна робота (тест): до 15 тижня семестру.
3. Лабораторна робота 1 (проект): до 4 тижня семестру.
4. Лабораторна робота 2 (проект): до 8 тижня семестру.
5. Лабораторна робота 3 (проект): до 12 тижня семестру.
6. Лабораторна робота 4 (проект): до 15 тижня семестру.

Студент має право на одне перескладання кожної контрольної роботи із можливістю отримання максимально 80% початково визначених за цю контрольну роботу балів. Термін перескладання визначається викладачем.

У випадку відсутності студента з поважних причин відпрацювання та перездачі контрольних робіт здійснюються у відповідності до „Положення про порядок оцінювання знань студентів при кредитно-модульній системі організації навчального процесу” від 1 жовтня 2010 року.

У разі неякісного виконання лабораторної роботи, викладач має право не зарахувати лабораторну роботу, або знизити за неї бали.

Студент має право здавати лабораторні роботи після закінчення визначеного для них терміну, але з втратою одного балу за кожен тиждень, який пройшов з моменту закінчення терміну її здачі.

7.3 Шкала відповідності оцінок

Відмінно / Excellent	90-100
Добре / Good	75-89
Задовільно / Satisfactory	60-74
Незадовільно / Fail	0-59
Зараховано / Passed	60-100
Не зараховано / Fail	0-59

8. Структура навчальної дисципліни. Тематичний план лекцій і лабораторних занять

III семестр

№ лекції	Назва лекції	Кількість годин		
		лекції	лаборат.	С/Р
<i>Частина 1. Основи програмування з використанням мови C/C++</i>				
1	Тема 1. Процедурне та об'єктно-орієнтоване програмування.	2	2	4
2	Тема 2. Програмне середовище C/C++	2	2	4
3	Тема 3. Базові поняття програмування	2	2	4
4	Тема 4. Використання вказівників (pointers)	2	2	4
5	Тема 5. Використання псевдонімів (reference)	2	2	4
6	Тема 6. Вирази та інструкції	2	2	4
7	Тема 7. Типи: масиви, ланцюжки символів, структури	1	2	4
	Модульна контрольна робота 1	1		
<i>Частина 2 Процедурне програмування</i>				
8	Тема 8. Функції в мові C/C++			
9	Тема 9. Сигнатура функції	2	2	4
10	Тема 10. Передача параметрів	2	2	4
11	Тема 11. Обчислення значення функції	2	2	4
12	Тема 12. Довизначення (overloading) функцій	2	2	4
13	Тема 13. Непряме використання функцій	2	2	4
14	Тема 14. Області видимості	2	1	4
15	Тема 15. Тривалість життя об'єктів	1	1	4
	Модульна контрольна робота 2	1		
	ВСЬОГО	30	28	60

Загальний обсяг 120 год, в тому числі:

Лекцій – 30 год.

Лабораторні роботи – 28 год.

Самостійна робота - 60 год.

Консультації - 2 год.

IV семестр

№ лекції	Назва лекції	Кількість годин		
		лекції	лаборат.	С/Р
Частина 3. Об'єктно-орієнтоване програмування				
1	Тема 16 Класи. Об'єкти класів.	2	2	4
2	Тема 17. Функції-члени класів	2	2	4
3	Тема 18. Статичні члени класів	2	2	4
4	Тема 19. Об'єднання. Бітове поле. Область видимості класа	2	2	4
5	Тема 20. Вкладені класи. Класи як простір імен	2	2	4
6	Тема 21. Ініціалізація, конструктори та деструктори класів	2	2	4
7	Тема 22. Перевантаження операторів та перетворення типів	1	2	4
	Модульна контрольна робота 3	1		
Частина 4. Проектування об'єктно-орієнтованих систем				
8	Тема 23. Шаблони класів (<i>templates</i>)			
9	Тема 24. Наслідування, віртуальні функції	2	2	4
10	Тема 25. Множинне наслідування	2	2	4
11	Тема 26. Віртуальне наслідування	2	2	4
12	Тема 27. Механізм RTTI. Виключення	2	2	4
13	Тема 28. Мова UML. Діаграма прецедентів, класів та об'єктів	2	2	4
14	Тема 29. Діаграми послідовностей, станів та діяльності	2	1	4
15	Тема 30. Патерни проектування (<i>design patterns</i>)	1	1	4
	Модульна контрольна робота 4	1		
	ВСЬОГО	30	28	60

Загальний обсяг 120 год, в тому числі:

Лекцій – 30 год.

Лабораторні роботи – 28 год.

Самостійна робота - 60 год.

Консультації - 2 год.

Теми, винесені на самостійне вивчення:

Використання систем контролю версій.

Виконання лабораторних робіт 1-4.

Умови лабораторних робіт:

Лабораторна робота 1: Розробка програми для 3D моделювання.

Лабораторна робота 2: Розробка мережевого застосунку.

Лабораторна робота 3: Проектування програмної системи з використанням UML.

Лабораторна робота 4: Розробка застосунку для Android.

Деталізовані умови лабораторних робіт розміщено за посиланнями:

https://drive.google.com/drive/folders/1285vxEpjq5GEREq7IVXFjVHDg9G_OXtm

9. Рекомендовані джерела:

Основна

1. Вирт Н. Систематическое программирование. Введение. М. Мир, 1977
2. Липшман С. Лажойе Ж. Язык программирования С++. Вводный курс. М. ДМК. 2001
3. Страуступ Б. Язык программирования С++. Специальное издание , БИНОМ, М. 2001.
4. Пильщиков В. Сборник упражнений по языку Паскаль. М. Наука 1989.
5. Завьялов Ю., Квасов Б., Мирошниченко В. Методы сплайн-функций М. Наука 1980.
6. Саттер Г. Решение сложных задач на С++. М. Вильямс 2002.
7. Самарский А. Введение в теорию разностных схем М. Наука 1971.

Додаткова:

8. Евстигнеев В. Применение теории графов в программировании. М. Наука 1985.
9. Мацяшек Л. Анализ требований и проектирование систем. М. Вильямс 2002.
10. Черников С. Линейные неравенства. М.Наука 1968.
11. Акулич И. Математическое программирование в примерах и задачах. М.Высш.шк.,1985.
12. <https://msdn.microsoft.com/>
13. <http://www.cplusplus.com/>
14. <http://en.cppreference.com/>
15. <https://git-scm.com/book/>.

10. Додаткові ресурси:

https://drive.google.com/drive/folders/1285vxEpjq5GEREq7IVXFjVHDg9G_OXtm

https://drive.google.com/drive/folders/1X_NdykxXuAX0c5oHoBlHpqHqt5NqNzgx