

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БУДІВНИЦТВА І
АРХІТЕКТУРИ

Кафедра інформаційних технологій

“ЗАТВЕРДЖУЮ”
Проректор з
навчально-методичної роботи
_____ / Г.М. Тонкачєв /
“ 09 ” червня 2017 року



РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Об’єктно-орієнтоване програмування
(назва навчальної дисципліни)

галузь знань 12 Інформаційні технології
(шифр і назва напрямку підготовки)

спеціальність 122 Комп’ютерні науки
126 Інформаційні системи і технології
(шифр і назва спеціальності)

спеціалізацій «Інформаційні управляючі системи та технології»,
«Інформаційні технології проектування»
(назва спеціалізації)

факультет автоматизації і інформаційних технологій
(назва факультету)

Київ – 2017 рік

Робоча програма з дисципліни:

«Об'єктно-орієнтоване програмування»

(назва навчальної дисципліни)

для студентів за напрямом підготовки:

6.050101 «Комп'ютерні науки»

(номер і назва)

Розробники:

Доманецька І.М., к.т.н., доцент /

(прізвище та ініціали, науковий ступінь, звання)

/ (підпис)

_____ /

(прізвище та ініціали, науковий ступінь, звання)

/ (підпис)

_____ /

(прізвище та ініціали, науковий ступінь, звання)

/ (підпис)

Робоча програма затверджена на засіданні кафедри

інформаційних технологій

протокол № 15 від "18" травня 2015 року

завідувач кафедри

_____ (підпис)

(Білощицький А.О.)

(прізвище та ініціали)

Схвалено методичною комісією ФАІТ

Протокол № 13 від "17" червня 2015 року

Голова МК

_____ (підпис)

(Смірнов В.М.)

(прізвище та ініціали)

1. Опис навчальної дисципліни

Галузь знань:

12 Інформаційні технології
(шифр і назва напрямку підготовки)

Спеціальність: 125 «Кібербезпека» (БІКС)
(шифр і назва спеціальності)

Ступінь:

бакалавр

Кількість кредитів – 8,5

Модулів – 2

Змістових модулів – 6

Загальна кількість годин – 255

Кількість годин для денної форми навчання:

аудиторних – 90

самостійної роботи студента – 165

Індивідуальне завдання : 1 РГР, Курсова робота
(курсний проект чи робота)

Вид навчальної роботи	Характеристика навчальної дисципліни			
	Денна форма навчання		Заочна форма навчання	
	Рік підготовки		Рік підготовки	
	2			
	семестр		семестр	
	3	4		
Лекції (год.)	12	20		не передбачено
Практичні заняття (год.)	6	22		
Лабораторні заняття (год.)	12	18		
Самостійна робота (год.)	45	120		
Індивідуальна робота (год.)				
Індивідуальне завдання (к-ть)	1РГР	1КР		
Вид контролю (зал. чи екз.)	зал.	екз.		
Усього (годин)	75	180		

Примітка.

Співвідношення кількості годин аудиторних занять до самостійної і індивідуальної роботи становить:

для денної форми навчання – $90/165=0,545$;

для заочної форми навчання – не передбачена.

2. Мета та завдання навчальної дисципліни

Метою викладання дисципліни є:

- ◆ набуття студентами теоретичних знань та практичних навичок по

- ♦ об'єктно-орієнтованому програмуванню;
- ♦ ознайомлення з сучасними мовами об'єктно-орієнтованого програмування та сучасними засобами розробки програм;
- ♦ набуття первинних знань з об'єктно-орієнтованого аналізу та проектування інформаційних систем та використання UML-нотацій для опису програмного застосування на різних етапах розробки.

Основні завдання:

- вивчення базових понять об'єктно-орієнтованого підходу до розробки програм, синтаксису та семантики сучасних об'єктно-орієнтованих мов програмування;
- знайомство з об'єктно-орієнтованою технологією розробки програмного забезпечення (об'єктно-орієнтований аналіз предметної області, об'єктно-орієнтоване проектування та програмування ПЗ);
- оволодіння сучасними інструментальними програмними засобами розробки об'єктно-орієнтованого програмного забезпечення.

В наслідок вивчення дисципліни “Об'єктно-орієнтоване програмування” студенти повинні *знати*:

- ♦ основні поняття, терміни та принципи об'єктно-орієнтованої технології розробки інформаційно-управляючих систем;
- ♦ засоби реалізації основних принципів об'єктно-орієнтованого підходу в сучасних мовах програмування високого рівня;
- ♦ засоби моделювання об'єктно-орієнтованого програмного забезпечення.

В наслідок вивчення дисципліни “Об'єктно-орієнтоване програмування” студенти повинні *вміти*:

- ♦ описувати абстрактні типи даних – класи та утворювати ієрархію класів;
- ♦ використовувати принци поліморфізму під час розробки ієрархії класів;
- ♦ використовувати принцип інкапсуляції при розробці класів;
- ♦ використовувати стандартні класи під час розробки програмного забезпечення;
- ♦ будувати UML –діаграми для опису елементів програмного забезпечення;
- ♦ за побудованими UML –діаграмами розробляти елементи програмного забезпечення;
- ♦ розробляти програмне забезпечення на базі сучасних інструментальних засобів розробки програм.

Міждисциплінарні зв'язки. Навчальна дисципліна “Об'єктно-орієнтоване програмування” є базовою для вивчення основних фахових дисциплін з напрямку підготовки “Комп'ютерні науки”.

Для вивчення дисципліни необхідні знання та навички з дисципліни “Алгоритмізація та програмування”.

3. Програма навчальної дисципліни

Модуль 1. Об'єктно-орієнтований підхід до створення програмного забезпечення

Змістовий модуль 1. Технологія об'єктно-орієнтованого програмування

Тема 1. Основи об'єктно-орієнтованого аналізу та проектування програмного забезпечення.

Еволюція технології розробки програмного забезпечення. Структурний та об'єктно-орієнтований підходи. Визначення об'єктно-орієнтованого аналізу, проектування та програмування.

Поняття об'єктно-орієнтованого аналізу предметного середовища. Загальні принципи побудови об'єктної моделі предметного середовища.

Поняття об'єктно-орієнтованого проектування програмного забезпечення. Поняття класу та об'єкту (екземпляру класу). Поля та методи класу.

Взаємовідносини між класами та об'єктами. Поняття успадкування. Композиція та колекція об'єктів.

Основи об'єктно-орієнтованого проектування програмного забезпечення мовою UML. Основні діаграми UML. Правила побудови UML-діаграм класів.

Тема 2. Об'єктно-орієнтоване програмування. Сучасні інструментальні засоби візуального програмування.

Сучасні середовища візуального програмування. Основні компоненти графічного інтерфейсу користувача. Використання бібліотеки візуальних компонентів (меню, панелі інструментів, шаблони діалогових вікон тощо). Побудова та опис діаграм прецедентів.

Тема 3. Основи програмування керованого подіями.

Архітектура програм, орієнтованих на події. Обробники подій від миші, клавіатури, команд меню, елементів управління тощо. Побудова та опис діаграм взаємодії та компонентів.

Змістовий модуль 2. Основи об'єктно-орієнтованої мови програмування C#

Тема 1. Розробка класів та об'єктів мовою C#.

Опис класу. Конструктори, деструктори класів. Створення екземпляру класу. Масиви об'єктів.

Статичні, константні члени класів.

Тема 2. Інкапсуляція.

Поняття інкапсуляції. Директиви видимості класів C# за замовчанням. Організація доступу до захищених полів класу. Дружні функції та класи.

Тема 3. Спадковість, просте та множинне успадкування.

Поняття спадковості. Опис класів-нащадків в C# при одиничній спадковості. Директиви видимості при спадкуванні в C#. Явний виклик конструкторів-предків. Послідовність викликів конструкторів/ деструкторів для ієрархії класів.

Множинна спадковість в C#. Проблеми ромбовидної спадковості, віртуальні предки.

Тема 4. Показчики на об'єкти. Передача об'єктів як параметрів функцій.

Операції динамічного виділення пам'яті new та delete мови C#.

Особливості передачі параметрів-об'єктів за значенням мови C#. Використання показчиків та посилань на об'єкти. Використання конструктору копії об'єкту. Повернення об'єкту в якості результату роботи функції.

Тема 5. Перевантаження функцій.

Неоднозначність при перевантаженні (сумісність типів параметрів, параметри-посилання, параметри за замовчанням). Перевантаження методів класу та конструкторів.

Тема 6. Перевантаження операцій.

Перевантаження бінарних та унарних операцій. Перевантаження операції індексування. Перевантаження операції присвоєння. Особливості присвоєння об'єктів мови C#. Перевантаження операцій потокового уведення-виведення.

Тема 7. Поліморфізм.

Поняття раннього та пізнього зв'язування. Поняття перекриття та заміщення методів. Механізм виклику заміщуваних методів, таблиці віртуальних методів. Використання показчиків на класи, як спосіб застосування поліморфізму. Виклик заміщуваних методів нащадків з класів предків. Абстрактні методи.

Тема 8. Шаблони функцій і класів.

Поняття шаблону функції. Опис шаблону. Створення екземпляру шаблону функції. Відмінності перевантаження функції та шаблонів функції.

Опис шаблону-класу. Створення екземпляру шаблону класу.

Змістовий модуль 3. Оброблення виключень та бібліотеки класів

Тема 1. Бібліотека стандартних шаблонів STL: основні поняття.

Складові бібліотеки STL: контейнери, ітератори, алгоритми. Базові класи-контейнери, їх загальна характеристика. Особливості роботи з елементами контейнера небазових типів. Ітератори як показчики на елементи контейнерів, типи ітераторів. Загальне поняття алгоритму.

Тема 2. Використання класів-контейнерів бібліотеки STL: vector, list, map.

Клас-контейнер vector. Способи створення. Методи класу вектор. Ітератори на елементи вектора.

Клас-контейнер list. Способи створення. Методи класу списку. Ітератори на елементи списку.

Клас-контейнер map. Способи створення. Методи класу асоціативний список. Ітератори на елементи.

Тема 3. Застосування алгоритмів бібліотеки STL.

Базові алгоритми бібліотеки STL. Поняття предикату, унарний та бінарний предикат. Приклади використання алгоритмів.

Тема 4. Класи потоків введення та виведення.

Перевантаження операцій потокового введення/виведення. Файлове введення/виведення. Управління потоковим введенням/виведенням.

Тема 5. Оброблення виняткових ситуацій в C#.

Використання оператора try-catch. Повторне збудження виключної ситуації throw.

Тема 6. Розроблення DLL-бібліотек.

Модуль 2. Курсова робота. Розробка об'єктно-орієнтованої програми

Змістовий модуль 1. Об'єктно-орієнтований аналіз та проектування програми

Тема 1. Дослідження та опис предметної області, розробка функцій програми, розробка прецедентів.

Тема 2. Розробка концептуальної моделі предметної області та діаграми класів.

Тема 3. Розробка діаграми компонентів.

Змістовий модуль 2. Розробка об'єктно-орієнтованої програми

Тема 1. Розробка математичних постановок задач.

Тема 2. Розробка ієрархії класів програмних об'єктів.

Тема 3. Розробка програмного інтерфейсу користувача.

Тема 4. Реалізація методів класів програмних об'єктів.

Тема 5. Перевірка працездатності програмного забезпечення: тестування та налагодження програми.

Змістовий модуль 3. Оформлення пояснювальної записки до курсової роботи

Тема 1. Побудова UML-діаграм: прецедентів, класів, послідовностей, компонентів.

Тема 2. Опис UML-діаграм, програмного інтерфейсу користувача, тестових прикладів роботи програми.

4. Структура навчальної дисципліни

Назви змістових модулів і тем	Кількість годин											
	денна форма						Заочна форма					
	усього	у тому числі					усьо го	у тому числі				
		л	п	ла б	ІР К	с.р.		л	п	лаб	ІРК	с.р.
1	2	3	4	5	6	7	8	9	10	11	12	13
Модуль 1. Об'єктно-орієнтований підхід до створення програмного забезпечення												
Змістовий модуль 1. Технологія об'єктно-орієнтованого програмування												
Тема 1. Основи об'єктно-орієнтованого аналізу та проектування програмного забезпечення.	6	2				4						
Тема 2. Об'єктно-орієнтоване програмування. Сучасні інструментальні засоби візуального програмування.	8	2		2		4						
Тема 3. Основи програмування керованого подіями.	10	2	2	2		4						
Разом за змістовим модулем 1	24	6	2	4	0	12						
Змістовий модуль 2. Основи об'єктно-орієнтованої мови програмування C#												
Тема 1. Розробка класів та об'єктів мовою C#.	10	2	2	2		4						
Тема 2. Інкапсуляція.	10	2	2	2		4						
Тема 3. Спадковість, просте та множинне успадкування.	10	2	2	2		4						

Назви змістових модулів і тем	Кількість годин											
	усього	денна форма					Заочна форма					
		у тому числі					усього	у тому числі				
1	2	л	п	лаб	ІРК	с.р.		8	л	п	лаб	ІРК
Тема 4. Показчики на об'єкти. Передача об'єктів як параметрів функцій.	10	2	2	2			4					
Тема 5. Перевантаження функцій.	9	1	2	2			4					
Тема 6. Перевантаження операцій.	9	1	2	2			4					
Тема 7. Поліморфізм.	10	2	2	2			4					
Тема 8. Шаблони функцій і класів.	6	2					4					
Разом за змістовим модулем 2	74	14	14	14	0		32					
Змістовий модуль 3. Оброблення виключень та бібліотеки класів												
Тема 1. Бібліотека стандартних шаблонів STL: основні поняття.	10	2	2	2			4					
Тема 2. Використання класів-контейнерів бібліотеки STL: vector, list, map.	11	2	2	2			5					
Тема 3. Застосування алгоритмів бібліотеки STL.	14	2	2	2			8					

Назви змістових модулів і тем	Кількість годин											
	усього	денна форма					Заочна форма					
		у тому числі					усього	у тому числі				
1	2	л	п	ла б	ІР К	с.р.		8	л	п	лаб	ІРК
Тема 4. Класи потоків уведення та виведення.	14	2	2	2		8						
Тема 5. Оброблення виняткових ситуацій в С#.	14	2	2	2		8						
Тема 6. Розроблення DLL-бібліотек.	14	2	2	2		8						
Разом за змістовим модулем 3	77	12	12	12	0	41						
Індивідуальне завдання. РГР. Реалізація спадковості.	24					24						
Усього годин	199	32	28	30	0	109						

Назви змістових модулів і тем	Кількість годин												
	денна форма						Заочна форма						
	усього	у тому числі					усього	у тому числі					
		л	п	лаб	ІРК	с.р.		л	п	лаб	ІРК	с.р.	
1	2	3	4	5	6	7	8	9	10	11	12	13	
Модуль 2. Курсова робота. Розробка об'єктно-орієнтованої програми													
Змістовий модуль 1. Об'єктно-орієнтований аналіз та проектування програми													
1	усього	Л	П	л	І	С	8	9	10	11	12	13	
Тема 1. Дослідження та опис предметної області, розробка функцій програми, розробка прецедентів.	4					4							
Тема 2. Розробка концептуальної моделі предметної області та діаграми класів.	4					4							
Тема 3. Розробка діаграми компонентів.	4					4							
Разом за змістовим модулем 1	12	0	0	0	0	12							
Змістовий модуль 2. Розробка об'єктно-орієнтованої програми													
Тема 1. Розробка математичних постановок задач.	4					4							
Тема 2. Розробка ієрархії класів програмних об'єктів.	8					8							

Назви змістових модулів і тем	Кількість годин											
	денна форма						Заочна форма					
	усього	у тому числі					усього	у тому числі				
		л	п	лаб	ІРК	с.р.		л	п	лаб	ІРК	с.р.
1	2	3	4	5	6	7	8	9	10	11	12	13
Тема 3. Розробка програмного інтерфейсу користувача.	8					8						
Тема 4. Реалізація методів класів програмних об'єктів.	8					8						
Тема 5. Перевірка працездатності програмного забезпечення: тестування та налагодження програми.	6					6						
Разом за змістовим модулем 2	34	0	0	0		34						
Змістовий модуль 3. Оформлення пояснювальної записки до курсової роботи												
Тема 1. Побудова UML-діаграм: прецедентів, класів, послідовностей, компонентів.	6					6						
Тема 2. Опис UML-діаграм, програмного інтерфейсу користувача, тестових прикладів роботи програми.	4					4						
Разом за змістовим модулем 3	10	0	0	0		10						

Назви змістових модулів і тем	Кількість годин											
	денна форма						Заочна форма					
	усього	у тому числі					усього	у тому числі				
л		п	лаб	ІРК	с.р.	л		п	лаб	ІРК	с.р.	
1	2	3	4	5	6	7	8	9	10	11	12	13
Усього годин	56	0	0	0		56						
Разом	255	32	28	30		165						

5. Темі практичних занять

№	Назва теми	Кількість годин	
		денна	заочна
1.	Розробка класів та об'єктів мовою С#. Показчики на об'єкти.	2	
2.	Абстракція.	2	
3.	Інкапсуляція.	2	
4.	Спадковість, просте та множинне успадкування.	2	
5.	Перевантаження функцій. Передача об'єктів як параметрів функцій.	2	
6.	Перевантаження операцій.	2	
7.	Поліморфізм.	2	
8.	Наслідування.	2	
9.	Шаблони функцій і класів.	2	
10.	Використання класів-контейнерів бібліотеки STL: vector.	2	
11.	Використання класів-контейнерів бібліотеки STL: list.	2	
12.	Використання класів-контейнерів бібліотеки STL: map.	2	
13.	Строки та делегати.	2	
14.	Застосування алгоритмів бібліотеки STL.	2	
	Разом	28	

6. Темі лабораторних занять

№	Назва теми	Кількість годин	
		денна	заочна
1.	Сучасні інструментальні засоби візуального програмування. Основи програмування керованого подіями.	2	
2.	Основи програмування керованого подіями.	2	
3.	Створення багатовіконних застосувань	2	
4.	Розробка класів та об'єктів мовою С#.	2	
5.	Інкапсуляція.	2	
6.	Спадковість, просте успадкування.	2	
7.	Спадковість, множинне успадкування.	2	
8.	Перевантаження функцій.	2	

9.	Перевантаження бінарних та унарних операцій.	2	
10.	Поліморфізм.	2	
11.	Використання класів-контейнерів бібліотеки STL: vector, list, map.	2	
12.	Застосування алгоритмів бібліотеки STL.	2	
13.	Класи потоків введення та виведення.	2	
14.	Оброблення виняткових ситуацій в C#.	2	
15.	Розроблення DLL-бібліотек.	2	
	Разом	30	

7. Самостійна робота

№	Назва теми	Кількість годин	
		денна	заочна
1.	Основи об'єктно-орієнтованого аналізу та проектування програмного забезпечення.	4	
2.	Об'єктно-орієнтоване програмування. Сучасні інструментальні засоби візуального програмування.	4	
3.	Основи програмування керованого подіями.	4	
4.	Розробка класів та об'єктів мовою C#. Інкапсуляція.	4	
5.	Розробка класів та об'єктів мовою C#. Спадковість, просте та множинне успадкування.	4	
6.	Перевантаження функцій. Показчики на об'єкти. Передача об'єктів як параметрів функцій.	4	
7.	Розробка класів та об'єктів мовою C#. Перевантаження операцій.	4	
8.	Розробка класів та об'єктів мовою C#. Поліморфізм.	4	
9.	Шаблони функцій і класів.	4	
10.	Бібліотека стандартних шаблонів STL: основні поняття.	4	
11.	Використання класів-контейнерів бібліотеки STL: vector, list, map та застосування алгоритмів бібліотеки STL.	5	
12.	Розроблення DLL-бібліотек.	5	
	Всього	50	
13.	Індивідуальне завдання. РГР. Реалізація спадковості.	24	
14.	Індивідуальне завдання. Модуль 2. Курсова робота.	56	
	Підготовка до іспиту		
15.	Повторення теоретичного матеріалу змістовного модуля 1	4	
16.	Розв'язок практичних завдань до ЗМ1. Побудова UML - діаграм прецедентів, класів, послідовностей та компонентів	4	

17.	Повторення теоретичного матеріалу змістовного модуля 2	4	
18.	Розв'язок практичних завдань до ЗМ2. Опис ієрархії класів, інкапсуляція, поліморфізм	4	
19.	Розв'язок практичних завдань до ЗМ2. Перевантаження функцій та операцій.	4	
20.	Розв'язок практичних завдань до ЗМ2.	4	
21.	Повторення теоретичного матеріалу змістовного модуля 3	4	
22.	Розв'язок практичних завдань до ЗМ3. Використання класів-шаблонів бібліотеки STL.	4	
23.	Розв'язок практичних завдань до ЗМ3. Управління потоковим введенням-виведенням	3	
	Всього	35	
	Разом	165	

8. Індивідуальні завдання

№	Назва теми	Кількість годин	
		денна	заочна
	Індивідуальне завдання. РГР. Реалізація спадковості.		
1.	Розробка батьківського класу та класу-нащадку	4	
2.	Розробка програмного інтерфейсу з користувачем	4	
3.	Тестування програмного забезпечення	4	
4.	Побудова діаграми прецедентів, опис прецеденту	4	
5.	Побудова та опис діаграми класів	4	
6.	Побудова та опис діаграми компонентів	4	
	Всього	24	
	Індивідуальне завдання. Модуль 2. Курсова робота.		
1.	Дослідження та опис предметної області, розробка функцій програми, розробка прецедентів.	4	
2.	Розробка концептуальної моделі предметної області та діаграми класів.	4	
3.	Розробка діаграми компонентів.	4	
4.	Розробка математичних постановок задач.	4	
5.	Розробка ієрархії класів програмних об'єктів.	8	
6.	Розробка програмного інтерфейсу користувача.	8	
7.	Реалізація методів класів програмних об'єктів.	8	
8.	Перевірка працездатності програмного забезпечення: тестування та налагодження програми.	6	
9.	Побудова UML-діаграм: прецедентів, класів, послідовностей, компонентів.	6	
10.	Опис UML-діаграм, програмного інтерфейсу користувача, тестових прикладів роботи програми.	4	

	Всього	56	
	Разом	80	

Тема курсових робіт:

1. Розробка програми для обліку посібників з навчальних дисциплінах.
2. Розробка програми калькуляції вартості страв на основі даних про кулінарні рецепти.
3. Розробка програми продажу квитків на поточні авіарейси.
4. Розробка програми роботи з відомостями про маршрути громадського транспорту.
5. Розробка програма – утилізатора файлів.
6. Розробка програми для підтримки процесу курсового проектування.
7. Розробка програми ведення ділового щоденника.
8. Розробка програми ведення обліку земельних ділянок населених пунктів.
9. Розробка програми ведення обліку учбового навантаження викладачів кафедри (“Вчений секретар”).
10. Розробка програми ведення розрахунків за комунальні платежі ЖЕКа.
11. Розробка програми графічного редактора зі збереженням зображення, списку об’єктів та їх параметрів у файлі.
12. Розробка програми для ведення довідника про випускників університету.
13. Розробка програми для контролю виконання студентами програми певної дисципліни.
14. Розробка програми для обліку пацієнтів ветеринарної клініки.
15. Розробка програми каталогізації носіїв інформації.
16. Розробка програми обліку відвідування занять студентами.
17. Розробка програми обліку зайнятості авто в агенції по прокату автомобілів.
18. Розробка програми обліку замовлень в агенції таксі.
19. Розробка програми обліку ліків в аптеці.
20. Розробка програми обліку міжміських та міжнародних телефонних переговорів.
21. Розробка програми обліку постояльців в готелі.
22. Розробка програми обліку спортивних змагань.
23. Розробка програми обліку товарів, що зберігаються на складах.
24. Розробка програми продажу путівок туристичної агенції.

- 25. Розробка програми роботи з об'єктами вулично-дорожньої мережі міста.
- 26. Розробка програми формування меню ресторану.
- 27. Розробка програми формування програми теленерעד на каналах.
- 28. Розробка програми формування сітки мовлення телевізійного каналу.

9. МЕТОДИ НАВЧАННЯ

При викладанні навчальної дисципліни використовуються словесний, інформаційно-ілюстративний, наочний та практичний, проблемний та пошуковий методи навчання із застосуванням лекцій, задач, ситуаційних завдань, моделювання конкретних ситуацій, комплексних розрахункових завдань, реферативних оглядів, провокаційних вправ і запитань, ділових ігор, мозкових атак.

10. МЕТОДИ КОНТРОЛЮ

Контрольні заходи передбачають проведення вхідного (за необхідності), поточного, модульного та семестрового контролю.

Вхідний, поточний, модульний контроль здійснюється під час проведення практичних та індивідуальних занять з викладачем.

Семестровий контроль виконується за окремим графіком, складеним деканатом факультету.

Засоби контролю засвоєння матеріалу курсових робіт – представлення та захист роботи чи проекту.

11. Розподіл балів, які отримують студенти

Поточне оцінювання			Підсумковий тест (екзамен)	Сума
Змістовий модуль № 1	Змістовий модуль № 2	Змістовий модуль №3		
~10	~30	~30	~30	100

Курсова робота

ЗМ1.ООАналіз та ООПроектування	ЗМ2. Розробка ООПрограми	ЗМ3. Оформлення та захист роботи	Сума
~30	~30	~40	100

Шкала оцінювання: національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	
		для екзамену, курсового проекту (роботи), практики	для заліку
90 – 100	A	відмінно	зараховано
82-89	B	добре	
74-81	C		
64-73	D	задовільно	
60-63	E		
35-59	FX	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
0-34	F	незадовільно з обов'язковим повторним вивченням дисципліни	не зараховано з обов'язковим повторним вивченням дисципліни

12. Методичне забезпечення

1. Конспект лекцій з дисципліни «Об'єктно-орієнтоване програмування». К.: КНУБА, 2008 - 51 с. (електронний варіант)
2. Методичні вказівки до виконання лабораторних робіт з дисципліни «Об'єктно-орієнтоване програмування» К.: КНУБА, 2009 - 32 с.
3. Методичні вказівки до виконання індивідуального завдання з дисципліни «Об'єктно-орієнтоване програмування» К.: КНУБА, 2011. – 32 с.
4. Методичні вказівки до виконання курсової роботи з дисципліни «Об'єктно-орієнтоване програмування» К.: КНУБА, 2011. – 28 с.

13. Рекомендована література

Базова

1. Глинський Я.М., Анохін В.Є., Ряжська В.А. С# і С# Builder: Навч.посібник. - 3-е видання. Львів: Деол, СПД Глинський, 2006. - 198 с.
2. Ларман К. Применение UML 2.0 и шаблонов проектирования. Введение в объектно-ориентированный анализ, проектирование и итеративную разработку: Учебное пособие. - 3-е издание М.:Вильямс, 2004. - 736 с.
3. Страуструп Б. Язык программирования С#. – М.:Бином-Пресс, 2008. - 1098

Допоміжна

1. Буч Г., Объектно-ориентированный анализ и проектирование с примерами приложений на С#, 3-е изд. \ Пер. с англ.- М.:ООО «И.Д. Вильямс», 2008 г. – 721 с. ил.
2. Шилдт Г. С#: базовый курс .-М.:Вильямс, 2008 .-620с.
3. Шилдт Г. Самоучитель С#. - СПб.: БХВ-Петербург, 2003. – 687 с.

14. Інформаційні ресурси

1. <http://library.knuba.edu.ua/>
2. <http://org.knuba.edu.ua/>