

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ,
МОЛОДІ ТА СПОРТУ УКРАЇНИ**

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ

**Робоча програма
навчальної дисципліни
"ОСНОВИ ОБ'ЄКТНО-ОРІЄНТОВАНОГО
ПРОГРАМУВАННЯ"**

**для студентів напряму підготовки 6.051501
"Видавничо-поліграфічна справа"
всіх форм навчання**

Харків. Вид. ХНЕУ, 2012

Затверджено на засіданні кафедри комп'ютерних систем і технологій.
Протокол № 2 від 29.08.2011 р.

Укладач Браткевич В. В.

P58 Робоча програма навчальної дисципліни "Основи об'єктно-орієнтованого програмування" для студентів напряму підготовки 6.051501 "Видавничо-поліграфічна справа" всіх форм навчання / укл. Браткевич В. В. – Х. : Вид. ХНЕУ, 2012. – 44 с. (Укр. мов.)

Подано тематичний план навчальної дисципліни та її зміст за модулями й темами, вміщено плани лекцій, лабораторних занять і самостійної роботи студента, систему оцінювання знань студентів. Визначено обсяги знань, які повинен опанувати студент з даної навчальної дисципліни.

Рекомендовано для студентів напряму підготовки 6.051501 "Видавничо-поліграфічна справа" всіх форм навчання.

ВСТУП

Навчальну дисципліну "Основи об'єктно-орієнтованого програмування" віднесено до групи вибіркових дисциплін підготовки бакалаврів для студентів напряму підготовки 6.051501 "Видавничо-поліграфічна справа" всіх форм навчання.

Враховано рекомендації щодо кредитно-модульної системи організації навчального процесу.

Мета навчальної дисципліни – надання студентам теоретичних основ, практичних і методичних рекомендацій щодо застосування принципів об'єктно-орієнтованого програмування в різноманітних галузях видавничо-поліграфічної діяльності де потрібна розробка сучасних програмних засобів.

Завдання навчальної дисципліни – оволодіння навичками об'єктно-орієнтованого програмування для інструментальної підтримки технологічного процесу виробництва видавничо-поліграфічних та мультимедійних продуктів.

Об'єкт навчальної дисципліни – процес об'єктно-орієнтованого програмування.

Предмет навчальної дисципліни – основні принципи об'єктно-орієнтованого програмування; зміст і структура типових бібліотек класів і шаблонів; технологія створення об'єктно-орієнтованих додатків.

Навчальна дисципліна дозволяє формувати у студентів такі компетенції:

компетенції соціально-особистісні, які пов'язані зі здатністю учитися, системно мислити, наполегливістю у досягненні мети та турботою про якість виконуваної роботи;

загальнонаукові компетенції, що забезпечують здатність використовувати математичних методів аналізу і синтезу; здатність мислити в термінах об'єктно-орієнтованої моделі; базові знання сучасних інформаційних технологій та уміння використовувати Інтернет-ресурси;

інструментальні компетенції, які дозволяють оволодіти навичками використання програмних засобів створення об'єктно-орієнтованих програмних продуктів, а також навичками виконання дослідницьких робіт;

загальнопрофесійні компетенції, які пов'язані з навичками роботи з комп'ютером; дають базові уявлення про об'єктно-орієнтовану модель програмування і її властивості, уявлення про розвиток інформаційних технологій на сучасному етапі, а також надають здатність координувати роботу декількох груп над єдиним об'єктно-орієнтованим проектом;

професійні компетенції, що забезпечують: здатність здійснювати декомпозицію проекту на сукупність окремих блоків і складати технічні вимоги до інформаційного забезпечення кожного з них; здатність обґрунтувати структуру класу та доцільність застосування певних типів даних для його опису; здатність планувати й реалізовувати заходи щодо підвищення ефективності існуючих об'єктно-орієнтованих програмних засобів;

спеціалізовано-професійні компетенції, які дозволяють: налаштовувати сучасні системи об'єктно-орієнтованого програмування (наприкладі Visual Studio. NET) з урахуванням заданих вимог; користуватися раніше складеними об'єктно-орієнтованими програмами і вносити зміни в програму, виконувати налагоджування програм за допомогою вбудованих інструментальних засобів; використовувати стандартні бібліотечні класи; створювати і використовувати власні класи; використовувати математичний апарат для розробки об'єктно-орієнтованих алгоритмів; здійснювати супровід зовнішніх об'єктно-орієнтованих програм.

Сфера реалізацій набутих знань, вмінь, компетенцій у майбутній професії

Основною сферою реалізацій набутих при вивченні знань, вмінь, компетенцій є області, де потрібне застосування сучасних об'єктно-орієнтованих мов програмування для інструментальної підтримки технологічного процесу виробництва видавничо-поліграфічних та мультимедійних продуктів (підрозділи електронних і мультимедійних видань, сучасні середовища розробки мультимедіа і Web-проектів).

Структура навчальної дисципліни "Основи об'єктно-орієнтованого програмування" наведена в табл. 1.

Структура навчальної дисципліни

Характеристика дисципліни: підготовка бакалаврів	Напрямок, галузь знань підготовки, освітньо-кваліфікаційний рівень	Характеристика навчальної дисципліни
Кількість кредитів відповідних ECTS: 3, у тому числі: змістовних модулів - 2; самостійна робота	Шифр та назва галузі знань: 0515 "Видавничо-поліграфічна справа"	Вибіркова Рік підготовки: 2. Семестр: 3
Кількість годин за змістовними модулями: модуль 1 – 58 год.; модуль 2 – 50 год. Усього –108 год.	Шифр та назва напрямку підготовки: 6.051501 "Видавничо-поліграфічна справа"	Кількість годин, відведених на лекції (теоретична підготовка) – 34 год.; лабораторні заняття – 34 год.; самостійну роботу – 40 год.
Кількість тижнів викладення навчальної дисципліни: 17. Кількість годин на тиждень – 4	Освітньо-кваліфікаційний рівень: бакалавр	Вид контролю: підсумковий модульний контроль (залік)

1. Кваліфікаційні вимоги до студентів

Необхідна навчальна база перед початком вивчення дисципліни: для успішного вивчення дисципліни необхідні базові знання, отримані студентами в об'ємі шкільної програми, а також знання при освоєнні ними попередніх навчальних дисциплін "Основи програмування" та "Інформаційні технології".

Практична спрямованість

Після вивчення навчальної дисципліни студенти повинні

ЗНАТИ:

основні парадигми об'єктно-орієнтованого програмування:

технологію розробки об'єктно-орієнтованих додатків;

склад та структуру типових бібліотек класів та шаблонів;

можливості застосування сучасних базових інструментальних програмних засобів, призначених для створення та налагодження об'єктно-орієнтованих додатків;

УМІТИ:

використовувати принципи об'єктно-орієнтованого програмування та сучасне інструментальне програмне забезпечення (Visual Studio .NET) для створювання об'єктно-орієнтованих додатків мовою C#;

користуватися раніше складеними програмами і здійснювати супровід програм, вносити зміни в програму, виконувати налагоджування програм за допомогою вбудованих інструментальних засобів.

2. Тематичний план навчальної дисципліни

При вивченні навчальної дисципліни студент має ознайомитися з програмою дисципліни, з її структурою, формами та методами навчання, видами та методами контролю знань.

Тематичний план навчальної дисципліни складається з двох модулів, кожний з них об'єднує у собі відносно окремий самостійний блок дисципліни, який логічно пов'язує кілька навчальних елементів дисципліни за змістом та взаємозв'язками. Навчальний процес здійснюється у таких формах: лекції, лабораторні заняття, індивідуальна робота, самостійна робота, контрольні заходи.

Розподіл навчального часу за формами навчання та видами занять відповідно до робочого навчального плану наведено в табл. 2.

Таблиця 2

Структура залікового кредиту навчальної дисципліни

Тема	Кількість годин, відведених на:		
	лекції	лабораторні заняття	самостійну роботу
1	2	3	4
Змістовний модуль 1. Вступ в об'єктно-орієнтоване програмування			
Тема 1. Базові концепції об'єктно-орієнтованого програмування (основні поняття)	2	4	4
Тема 2. Вступ у класи	4	4	4

1	2	3	4
Тема 3. Структури і переліки в класах	2		2
Тема 4. Відносини між класами	6	6	6
Тема 5. Інтерфейси	2		2
Тема 6. Делегати і події	2	4	4
Разом (модуль 1) – 58 год.	18	18	22
Змістовний модуль 2 Проектування графічних об'єктно-орієнтованих додатків			
Тема 7. Вступ у Windows-форми	4	4	4
Тема 8. Організація інтерфейсу і робота з Designer Forms	4	6	6
Тема 9. Вступ у графік.	4	4	4
Тема 10. Геометричні перетворення	4	2	4
Разом (модуль 2) – 50 год.	16	16	18
Усього годин 108	34	34	40

3. Зміст навчальної дисципліни за модулями та темами

Змістовний модуль 1. Вступ в об'єктно-орієнтоване програмування

Тема 1. Базові концепції об'єктно-орієнтованого програмування (основні поняття)

1.1. Поняття об'єкта і об'єктної моделі.

Визначення об'єкта. Принципи ООП: абстрагування, інкапсуляція, ієрархія, модульна.

1.2. Опис елементів об'єктної моделі і її властивостей.

Застосування уніфікованої мови моделювання (UML) для опису класів і об'єктів. Властивості і поля. Організація доступу до стану об'єкта. Методи. Життєвий цикл об'єкта. Конструктори. Деструктори.

1.3. Методи ООП.

Інтерфейси. Спадкоємство. Режими доступу до елементів даного класу. Віртуальні елементи базового класу. Абстрактні базові класи. Поліморфізм. Взаємостосунки між об'єктами. Перевантаження операторів. Події. Демонстраційний приклад взаємодії об'єктів у Windows-додатку.

1.4. Особливості вивчення ООП.

Порівняння ефективності методик "заучування" і "вивчення" ООП.

Тема 2. Вступ у класи

2.1. Визначення класу.

Синтаксис класу. Поля класу. Доступ до полів.

2.2. Методи класу.

Доступ до методів. Методи-властивості. Індексатори. Статичні поля і методи класу. Константи. Конструктори. Деструктори. Приклади проектування класу.

2.3. Майстри Visual Studio для роботи з елементами класу.

Додавання методів. Додавання властивостей. Додавання полей.

Тема 3. Структури і переліки в класах

3.1. Класи і структури.

Розгорнені і посилальні типи. Синтаксис структур. Вбудовані структури.

3.2. Переліки.

Синтаксис переліків. Приклади оголошень класів-переліків.

Тема 4. Відносини між класами

4.1. Типи відносин між класами.

Відношення "клієнт-постачальник. Відношення "батько-спадкоємець.

4.2. Спадкоємство.

Проста ієрархія класів. Доступ до елементів класу і спадкоємство. Використовування захищеного доступу. Конструктори і спадкоємство. Віртуальні методи і їх перевизначення. Абстрактні класи.

Заховання методів базового класу. Виклик перевизначених або прихованих методів базового класу. Ключове слово this. Вкладені визначення типів

4.3. Поліморфізм.

Механізми, що забезпечують поліморфізм.

Тема 5. Інтерфейси

5.1. Визначення інтерфейсів.

Синтаксис визначення інтерфейсів. Порівняння інтерфейсів і абстрактних класів.

5.1. Реалізація інтерфейсів.

Реалізація інтерфейсів у класах. Явна реалізація елементів інтерфейсу.

Тема 6. Делегати і події

6.1. Делегати.

Призначення делегатів. Синтаксис оголошення. Приклад використання. Застосування делегатів для створення списку викликів. Приклад.

6.2. Події.

Взаємодія об'єктів. Посилка і отримання повідомлення про подію. Делегати і події. Синтаксис оголошення події.

6.3. Обробка подій в середовищі .NET Framework.

Класи receiver. Клас Sender. Використовування вбудованого делегата EventHandler. Приклади програм з обробкою подій.

Змістовний модуль 2. Проектування графічних об'єктно-орієнтованих додатків

Тема 7. Вступ у Windows-форми.

7.1. Типова структура Windows-додатка.

Клас Form. Структура простого Windows-додатка, що заснована на застосуванні вікон. Аналіз коду стандартної Windows-форми.

7.2. Створення управляючих елементів.

Розробка коду управляючого елемента (на прикладі кнопки).

7.3. Обробка повідомлень у Windows-додатках.

Обробка повідомлень у Windows-додатках (на прикладі обробки повідомлень від кнопки).

7.4. Вікна повідомлень.

Клас MessageBox. Метод Show(). Приклади і особливості застосування.

7.5. Створення меню.

Розробка коду меню головного вікна. Класи MainMenu і MenuItem.
Алгоритм створення меню. Демонстраційний приклад.

Тема 8. Організація інтерфейсу і робота з Designer Forms

8.1. Огляд стандартних елементів.

Група командних об'єктів. Група текстових об'єктів. Група перемикачів. Група контейнерів. Група графічних елементів. Діалогові вікна.

8.2. Форма і стандартні елементи управління.

Ієрархія класів елементів управління. Взаємодія форм. Модальні і немодальні форми. Передача інформації між формами.

8.3. Типові приклади Windows Forms додатків.

Додавання і видалення елементів управління в режимі роботи додатку. Організація перевірки значень, що вводяться користувачем. Демонстраційний приклад багато віконного додатка-презентації.

Тема 9. Вступ у графіку

9.1. Огляд виведення графічної інформації.

Клас Graphics. Система координат. Кольори. Видалення об'єктів.

9.2. Малювання фігур і ліній, виведення тексту.

Виведення ліній за допомогою класу Pen. Малювання фігур за допомогою класу Brush. Графічне виведення тексту за допомогою класу Font.

9.3. Малювання з використанням образів.

Малювання за допомогою класу TextureBrush. Малювання образів за допомогою класу Pen. Графічне виведення тексту за допомогою класу Image. Подвійна буферизація.

Тема 10. Геометричні перетворення

10.1. Матриці перетворення.

Анімація. Віртуальний осцилограф. Виведення об'єкта, що рухається, в задане місце на фоновому малюнку.

10.3. Синтез цифрових зображень.

Фрактали. Узори. Геометричні фрактали. Сніжинка Коха. Драконова ламана. Трикутник Серпінського. Множина Юліа.

4. Плани лекцій

Змістовний модуль 1. Вступ в об'єктно-орієнтоване програмування

Тема 1. Базові концепції об'єктно-орієнтованого програмування (основні поняття)

- 1.1 Поняття об'єкта і об'єктної моделі.
- 1.2. Опис елементів об'єктної моделі і її властивостей.
- 1.3. Методи ООП.
- 1.4. Особливості вивчення ООП.

Література: основна [1; 3]; додаткова [5]; ресурси мережі Інтернет [7].

Тема 2. Вступ у класи

- 2.1. Визначення класу.
- 2.2. Методи класу.
- 2.3. Майстри Visual Studio для роботи з елементами класу.

Література: основна [1; 3]; додаткова [5]; ресурси мережі Інтернет [7].

Тема 3. Структури і переліки в класах

- 3.1. Класи і структури.
- 3.2. Переліки.

Література: основна [1; 2]; додаткова [5]; ресурси мережі Інтернет [6; 7].

Тема 4. Відносини між класами

- 4.1. Типи відносин між класами.
- 4.2. Спадкоємство.
- 4.3. Поліморфізм.

Література: основна [1; 2]; додаткова [4; 5]; ресурси мережі Інтернет [6; 7].

Тема 5. Інтерфейси

- 5.1. Визначення інтерфейсів.
- 5.2. Реалізація інтерфейсів.

Література: основна [2; 3]; додаткова [5]; ресурси мережі Інтернет: [6;7].

Тема 6. Делегати і події

- 6.1. Делегати.
- 6.2. Події.
- 6.3. Обробка подій у середовищі .NET Framework.

Література: основна [1; 2]; додаткова [5]; ресурси мережі Інтернет [6;7].

Змістовний модуль 2. Проектування графічних об'єктно-орієнтованих додатків

Тема 7. Вступ у Windows-форми

- 7.1. Типова структура Windows-дodatка.
- 7.2. Створення управляючих елементів.
- 7.3. Обробка повідомлень у Windows-дodatках.
- 7.4. Вікна повідомлень.
- 7.5. Створення меню.

Література: основна [1; 2]; додаткова [5]; ресурси мережі Інтернет [6;7].

Тема 8. Організація інтерфейсу і робота з Designer Forms

- 8.1. Огляд стандартних елементів.
- 8.2. Форма і стандартні елементи управління.
- 8.3. Типові приклади Windows Forms додатків.

Література: основна [1; 2]; додаткова [5]; ресурси мережі Інтернет [6].

Тема 9. Вступ у графіку

- 9.1. Огляд виведення графічної інформації.
- 9.2. Малювання фігур і ліній, виведення тексту.
- 9.3. Малювання з використанням образів.

Література: основна [1–3]; додаткова [5]; ресурси мережі Інтернет [6].

Тема 10. Геометричні перетворення

- 10.1. Матриці перетворення.
- 10.3. Синтез цифрових зображень.

Література: основна [2]; додаткова [5]; ресурси мережі Інтернет [6].

5. Плани лабораторних занять

Лабораторне заняття – це форма навчального процесу, спрямована на закріплення студентом теоретичних знань, отриманих на лекційних заняттях, а також форма вироблення практичних вмінь та навичок щодо вирішування типових задач програмування в галузі підтримки різноманітних технологій видавничо-поліграфічної справи.

Лабораторні заняття проводяться в обчислювальному центрі з половиною академічної групи.

Передбачається, що основну роботу щодо виконання лабораторних завдань студент здійснює дома, а у процесі проведення лабораторного заняття викладач організує тільки такі види методичної роботи з студентами:

перевірку, або формулювання індивідуальних завдань щодо розробки поточних програм та алгоритмів;

консультацію та допомогу у вирішенні найбільш складних частин реалізації лабораторних завдань;

демонстраційний захист індивідуальних робіт окремих студентів, або творчих груп із 2–3 студентів;

дискусії щодо можливих варіантів виконання окремих проектів або їх частин.

На кожному лабораторному занятті викладач оцінює підготовлені студентами поточні звіти, активність у дискусії, уміння працювати на робочому місці, а також можливість формулювати і відстоювати свою позицію тощо.

Підсумкові оцінки за кожне лабораторне заняття вносяться у відповідний журнал.

Отримані студентом оцінки враховуються при виставленні поточної модульної (практичний модульний контроль) оцінки з даної навчальної дисципліни.

Перелік тем лабораторних занять наведено у табл. 3.

Таблиця 3

Перелік тем лабораторних занять

Назва теми	Перелік лабораторних занять	Кількість годин	Література
1	2	3	4
Змістовний модуль 1. Вступ в об'єктно-орієнтоване програмування			
Тема 1. Базові концепції об'єктно-орієнтованого програмування (основні поняття)	Лабораторне заняття № 1. Визначення і дослідження найпростіших класів.	2	Основна [1; 3]; додаткова [5]; ресурси мережі Інтернет [7]
	Лабораторне заняття № 2. Розробка і дослідження простих методів.	2	
Тема 2. Вступ у класи	Лабораторне заняття № 3. Вивчення особливостей застосування конструкторів	4	

1	2	3	4
Тема 4. Відносини між класами	Лабораторне заняття № 4. Розробка програм з ієрархією класів	6	Основна [1; 2]; додаткова [4; 5]; ресурси мережі Інтернет [6; 7]
Тема 6. Делегати і події	Лабораторне заняття № 5. Застосування делегатів для обробки подій	4	Основна [1; 2]; додаткова [5]; ресурси мережі Інтернет: [6; 7]
Разом (модуль 1) годин		18	
Змістовний модуль 2. Проектування графічних об'єктно-орієнтованих додатків			
Тема 7. Вступ у Windows-форми	Лабораторне заняття № 6. Розробка і дослідження типового каркаса графічного Windows-дodatка	4	Основна [1; 2]; додаткова [5]; ресурси мережі Інтернет [6; 7]
Тема 8. Організація інтерфейсу і робота з Designer Forms	Лабораторне заняття № 7. Організація інтерфейсу графічного Windows-дodatка і робота з Designer Forms.	6	Основна [1; 2]; додаткова [5]; ресурси мережі Інтернет [6]
Тема 9. Вступ у графіку	Лабораторне заняття № 8. Розробка і дослідження Windows-дodatків з елементами анімаційної графіки	6	Основна [1; 2]; додаткова [5]; ресурси мережі Інтернет [6; 7]
Тема 10. Геометричні перетворення			
Разом (модуль 2) годин		16	
Разом годин за модулями		34	

Змістовний модуль 1. Вступ в об'єктно-орієнтоване програмування

Лабораторне заняття № 1 Визначення і дослідження найпростіших класів

Мета заняття: отримати практичні навички розробки програм з найпростішими класами.

Зміст: після виконання лабораторної роботи студент повинен

знати:

основні концепції об'єктно-орієнтованого програмування;

загальну форму визначення класу;

правила створення об'єктів;

особливості застосування змінних посилального типу;

типову структуру C#-програми з декількома класами;

уміти:

написати програму, що містить клас / класи, який інкапсулює інформацію про заданий об'єкт (наочна область задається індивідуально);

написати програму, яка обробляє інформацію (елементарні обчислення і висновки на екран вмісту елементів-даних) про декілька об'єктів одного класу.

Лабораторне заняття № 2

Розробка і дослідження простих методів

Мета заняття: отримати практичні навички розробки програм з класами, що містять найпростіші методи.

Зміст: після виконання лабораторної роботи студент повинен

знати:

формат запису методів;

способи повернення з методів і повернення значень;

правила передачі параметрів-значень;

уміти:

для заданої наочної області визначити клас, який інкапсулює елементи-методи, оброблювальні елементи-дані;

написати програму, яка обробляє інформацію (обчислення і висновки на екран результатів обчислень і вмісту елементів-даних) про декілька об'єктів одного класу заданої наочної області.

Лабораторне заняття № 3

Вивчення особливостей застосування конструкторів

Мета заняття: отримати практичні навички роботи з конструкторами.

Зміст: після виконання лабораторної роботи студент повинен

знати:

призначення конструктора і формати його запису;
особливості застосування оператора new до змінних типу значень;
призначення системи "збору сміття" в С# і формат запису деструкції;
призначення ключового слова this;

уміти:

для заданої наочної області визначити клас, який інкапсулює елементи-дані, елементи-методи і конструктори, що параметризуються.

написати програму, оброблювальну інформацію (обчислення і висновки на екран результатів обчислень) про декілька об'єктів одного класу заданої наочної області.

Лабораторне заняття № 4 **Розробка програм з ієрархією класів**

Мета заняття: отримати практичні навички розробки об'єктно-орієнтованих програм з ієрархією простих класів.

Зміст: після виконання лабораторної роботи студент повинен

знати:

призначення спадкоємства як фундаментального принципу ООП;
організацію доступу до елементів класу при спадкоємстві;
механізм використання захищеного доступу до елементів класу за допомогою модифікатора protected;
особливості використання конструкторів при спадкоємстві;

уміти:

для заданої наочної області написати програму, яка:
складається з розроблених простого базового класу і виведеного з нього двох – трьох похідних класів;
здійснює елементарну обробку і висновки інформації про властивості і результати обробки для окремих об'єктів успадкованого класу.

Лабораторне заняття № 5 **Застосування делегатів для обробки подій**

Мета заняття: отримати практичні навички розробки програм, оброблювальних найпростіші подій.

Зміст: після виконання лабораторної роботи студент повинен

знати:

призначення і синтаксис оголошення делегатів;

призначення і синтаксис оголошення подій;

типову процедуру обробки подій;

уміти:

написати програму, яка здійснює обробку найпростіших подій (натиснення заданих клавіш на клавіатурі).

Лабораторне заняття № 6

Розробка і дослідження типового каркаса графічного Windows-дodatка

Мета заняття: отримати практичні навички розробки (шляхом написання відповідного коду) типового каркаса Windows-дodatка, що містять основне меню в головному вікні.

Зміст: після виконання лабораторної роботи студент повинен

знати:

особливості взаємодії користувача з Windows;

типову структуру Windows-дodatка, засновану на застосуванні вікон;

технологію створення типових управляючих елементів;

уміти:

використовувати стандартні вікна повідомлень;

розробити Windows-дodatок, що складається з декількох вікон і здійснює виклик з головного меню основного вікна додатку, дочірні вікна із заданими зображеннями

Лабораторне заняття № 7

Організація інтерфейсу графічного Windows-дodatка і робота з Designer Forms

Мета заняття: отримати практичні навички розробки графічних Windows-дodatки, використовуючи стандартні засоби проектування.

Зміст: після виконання лабораторної роботи студент повинен

знати:

призначення і особливості застосування групи командних і текстових об'єктів; групи перемикачів; групи контейнерів; групи графічних елементів; діалогових вікон.

ієрархію класів елементів управління. Взаємодія форм. Призначення модальних і немодальних форми. Принципи організації обміну інформацією між формами;

уміти:

розробити багатовіконний графічний Windows-додаток, використовуючи відповідні майстри і стандартні елементи Designer Forms;

додавати і видаляти елементи управління в режимі роботи додатку. Здійснювати перевірку що вводяться користувачем значень.

Лабораторне заняття № 8

Розробка і дослідження Windows-додатків з елементами анімаційної графіки

Мета заняття: отримати практичні навички розробки Windows-додатків з елементами анімаційної графіки.

Зміст: після виконання лабораторної роботи студент повинен

знати:

принципи застосування GDI++;

принципи анімації;

принципи синтезу цифрових зображень;

методи форми для обробки сигналів від миші та клавіатури;

уміти:

малювати фігури і ліній, виконувати виведення тексту;

виконувати виведення ліній за допомогою класу Pen. Виконувати малювання фігур за допомогою класу Brush;

виконувати малювання з використанням образів;

виконувати малювання з використанням фракталів.

6. Самостійна робота студентів

Метою самостійної роботи є поглиблення знань, які було отримано на лекційних заняттях, та підтвердження і реалізація умінь, що були сформовані на лабораторних заняттях. Така робота потребує від студентів інтенсивної самостійної роботи над спеціальною літературою, та програмним забезпеченням. Студентам потрібно ознайомитись конкретними реалізаціями відповідних алгоритмів обробки інформації.

Також рекомендується опрацювати аналітичні статті у журналах, що присвячені проблемам функціонування та розвитку комп'ютерних технологій і програмування.

Основні види самостійної роботи, які запропоновані студентам:

1. Вивчення лекційного матеріалу.
2. Робота з опрацювання та вивчення рекомендованої літератури.
3. Виконання завдань на ПЕОМ з використанням програмного забезпечення.
4. Вивчення основних термінів та понять за темами дисципліни.
5. Підготовка до лабораторних занять, дискусій, роботи в малих групах, тестування.
6. Контрольна перевірка кожним студентом особистих знань за запитаннями для самоконтролю.

Перелік питань для самостійного опрацювання подано в табл. 4.

Таблиця 4

**Перелік питань для самостійного опрацювання та вивчення
рекомендованої літератури**

Назва теми	Питання для самостійного опрацювання (за модулями та темами)	Рекомендована література
1	2	3
Змістовний модуль 1. Вступ в об'єктно-орієнтоване програмування		
Тема 1. Базові концепції об'єктно-орієнтованого програмування (основні поняття)	Застосування уніфікованої мови моделювання (UML) для опису класів і об'єктів	Основна [1; 3]; додаткова [5]; ресурси мережі Інтернет [7]
Тема 2. Вступ у класи	Приклади проектування класу	Основна [1; 4]; додаткова [5]; ресурси мережі Інтернет [1]
Тема 3. Структури і переліки в класах	Приклади оголошень класів-переліків	Основна [1; 2]; додаткова [5]; ресурси мережі Інтернет [6;7]
Тема 4. Відносини між класами	Віртуальні методи і їх перевизначення. Абстрактні класи	Основна [1; 2]; додаткова [4; 5]; ресурси мережі Інтернет [6; 7]

1	2	3
Тема 5. Інтерфейси	Реалізація інтерфейсів у класах	Основна [2; 3]; додаткова [5]; ресурси мережі Інтернет [6; 7]
Тема 6. Делегати і події	Використовування вбудованого делегата EventHandler. Приклади програм з обробкою подій	Основна [1; 2]; додаткова [5]; ресурси мережі Інтернет [6; 7]
Змістовний модуль 2 Проектування графічних об'єктно-орієнтованих додатків		
Тема 7. Вступ у Windows-форми	Обробка повідомлень в Windows-додатках. Розробка коду меню головного вікна	Основна [1; 2]; додаткова [5]; ресурси мережі Інтернет [6; 7]
Тема 8. Організація інтерфейсу і робота з Designer Forms	Приклад багатовіконного додатка-презентації	Основна [1; 2]; додаткова [5]; ресурси мережі Інтернет [6].
Тема 9. Вступ у графіку	Малювання образів за допомогою класу Pen. Графічне виведення тексту за допомогою класу Image	Основна [1; 2]; додаткова [5]; ресурси мережі Інтернет [6]
Тема 10. Геометричні перетворення	Приклад розробки віртуального осцилографу або часів	Основна [2]; додаткова [5]; ресурси мережі Інтернет [6; 7]

Перелік завдань для самостійного опрацювання

Нижче наведено перелік пропонованих завдань до самостійної роботи студентів.

Завдання 1

Розробити презентацію (10 – 15 кадрів) на тему – "Ілюстрація базових концепцій об'єктно-орієнтованого програмування на прикладі заданої наочної області (тут указується індивідуально вибрана наочна область)".

Мета. Закріпити основні теоретичні положення, що лежать в основі об'єктно-орієнтованого програмування.

Як основа для виконання завдання 1 слід використовувати роздавальний матеріал до лекційної теми 1 "Базові концепції об'єктно-орієнтованого програмування (основні поняття)".

Кількість і склад презентаційних слайдів необхідно погоджувати з викладачем. Термін задачі – 8 навчальний тиждень.

Індивідуальні варіанти завдання.

Варіант № 1

Оголосити клас "Садиба" (Manor) і визначити його методи. Клас повинен містити наступні поля і методи:

рік споруди (1600 – 2006);

кількість поверхів (1 – 20);

вартість реставраційних робіт (1000 – 100000 грн);

вартість садиби (10000 – 500000 \$);

конструктор із значеннями полів за замовчуванням;

метод для установки значень полів;

метод для друку значень полів.

(Перелік полів і методів може бути доповнений користувачем за узгодженням з викладачем.)

Написати програму-драйвер, яка створює три екземпляри шаблону класу "Садиба". Програма повинна роздруковувати значення полів створених об'єктів, потім встановлювати нові значення полів кожного з об'єктів і знов роздруковувати результат.

Варіант № 2

Оголосити клас "Мультиплікаційний фільм" (Animated film) і визначити його методи. Клас повинен містити наступні поля і методи:

назва мультиплікаційного фільму (не більше 25 символів);

рік виходу на екран (1950 – 2006);

час, що відводиться на трансляцію (у хвиликах);

вартість озвучування (100 – 10000 грн);

конструктор із значеннями полів за замовчуванням;

метод для установки значень полів;

метод для друку значень полів.

(Перелік полів і методів може бути доповнений користувачем за узгодженням з викладачем.)

Написати програму-драйвер, яка створює три екземпляри шаблону класу "Мультиплікаційний фільм". Програма повинна роздруковувати значення полів створених об'єктів, потім встановлювати нові значення полів кожного з об'єктів і знов роздруковувати результат.

Варіант № 3

Оголосити клас "Бокс" (Boxing) і визначити його методи. Клас повинен містити наступні поля і методи:

прізвище боксера (не більше 25 символів);

кількість проведених боїв (0 – 50);
кількість перемог нокаутом (0 – 30);
час найкоротшого бою, що закінчився перемогою (в хвиликах);
гонорар за останній проведений бій (1000 – 1000000 грн);
конструктор із значеннями полів за замовчуванням;
метод для установки значень полів;
метод для друку значень полів.

(Перелік полів і методів може бути доповнений користувачем за узгодженням з викладачем.)

Написати програму-драйвер, яка створює три екземпляри шаблону класу "Бокс". Програма повинна роздруковувати значення полів створених об'єктів, потім встановлювати нові значення полів кожного з об'єктів і знов роздруковувати результат.

Варіант № 4

Оголосити клас "Зоопарк" (Zoo) і визначити його методи. Клас повинен містити наступні поля і методи:

назва зоопарку (не більше 30 символів);
прізвище директора (не більше 25 символів);
середня заробітна платня працівника зоопарку (300 – 1500 грн);
вартість квитка (1 – 10 грн);
конструктор із значеннями полів за замовчуванням;
метод для установки значень полів;
метод для друку значень полів.

(Перелік полів і методів може бути доповнений користувачем за узгодженням з викладачем.)

Написати програму-драйвер, яка створює три екземпляри шаблону класу "Зоопарк". Програма повинна роздруковувати значення полів створених об'єктів, потім встановлювати нові значення полів кожного з об'єктів і знов роздруковувати результат.

Варіант № 5

Оголосити клас "Банк " (Bank) і визначити його методи. Клас повинен містити наступні поля і методи:

назва банку (не більше 15 символів);
прізвище директора банку (не більше 25 символів);
статутної капітал (в млн грн);
процентна ставка за кредитом (10 – 30 %);
конструктор із значеннями полів за замовчуванням;

метод для установки значень полів;

метод для друку значень полів.

(Перелік полів і методів може бути доповнений користувачем за узгодженням з викладачем.)

Написати програму-драйвер, яка створює три екземпляри шаблону класу "Банк". Програма повинна роздруковувати значення полів створених об'єктів, потім встановлювати нові значення полів кожного з об'єктів і знов роздруковувати результат.

Варіант № 6

Оголосити клас "Студент" (Student) і визначити його методи. Клас повинен містити наступні поля і методи:

прізвище студента (не більше 20 символів);

рік вступу до університету (2001 – 2006);

назва факультету (не більше 15 символів);

вік (17 – 25 років);

розмір стипендії (100 – 250 грн);

кількість годин у тиждень, що проводиться в бібліотеці університету (0 – 25);

конструктор із значеннями полів за замовчуванням;

метод для установки значень полів;

метод для друку значень полів.

(Перелік полів і методів може бути доповнений користувачем за узгодженням з викладачем.)

Написати програму-драйвер, яка створює три екземпляри шаблону класу "Студент". Програма повинна роздруковувати значення полів створених об'єктів, потім встановлювати нові значення полів кожного з об'єктів і знов роздруковувати результат.

Варіант № 7

Оголосити клас "Потяг" (Train) і визначити його методи. Клас повинен містити наступні поля і методи:

назва потягу (не більше 25 символів);

платформа, з якою відбувається відправлення потягу (1 – 15);

вартість квитка (20 – 300 грн);

кількість годин в дорозі (1 – 200);

конструктор із значеннями полів за замовчуванням;

метод для установки значень полів;

метод для друку значень полів.

(Перелік полів і методів може бути доповнений користувачем за узгодженням з викладачем).

Написати програму-драйвер, яка створює три екземпляри шаблону класу "Потяг". Програма повинна роздруковувати значення полів створених об'єктів, потім встановлювати нові значення полів кожного з об'єктів і знов роздруковувати результат.

Варіант № 8

Оголосити клас "Конкурс краси" (Competition beauty) і визначити його методи. Клас повинен містити наступні поля і методи:

місце проведення конкурсу (не більше 20 символів);

рік проведення конкурсу (1995 – 2006);

сума головного призу \$ (1000 – 10000);

загальна кількість учасниць (10 – 120);

вік наймолодшої учасниці (14 – 18 років);

вік самої старшої учасниці (22 – 27);

конструктор із значеннями полів за замовчуванням;

метод для установки значень полів;

метод для друку значень полів.

(Перелік полів і методів може бути доповнений користувачем за узгодженням з викладачем.)

Написати програму-драйвер, яка створює три екземпляри шаблону класу "Конкурс краси". Програма повинна роздруковувати значення полів створених об'єктів, потім встановлювати нові значення полів кожного з об'єктів і знов роздруковувати результат.

Варіант № 9

Оголосити клас "Наукова стаття" (Scientific article) і визначити його методи. Клас повинен містити наступні поля і методи:

назва (не більше 40 символів);

прізвища авторів (не більше 35 символів);

кількість сторінок (4 – 25);

вартість публікації однієї сторінки в науковому журналі або збірці наукової статі (10 – 50 грн);

назва наукового журналу або збірки наукової статі (не більше 25 символів);

банківський відсоток за оплату наукової статті (1 – 3 %);

конструктор із значеннями полів за замовчуванням;

метод для установки значень полів;

метод для друку значень полів.

(Перелік полів і методів може бути доповнений користувачем за узгодженням з викладачем.)

Написати програму-драйвер, яка створює три екземпляри шаблону класу "Наукова стаття". Програма повинна роздруковувати значення полів створених об'єктів, потім встановлювати нові значення полів кожного з об'єктів і знов роздруковувати результат.

Варіант № 10

Оголосити клас "Пісня" (Song) і визначити його методи. Клас повинен містити наступні поля і методи:

назва (не більше 20 символів);

сценічний псевдонім або прізвище виконавця (не більше 25 символів);

рейтинг популярності пісні (по 10 вимірній шкалі);

вартість аранжировки (задається користувачем самостійно);

вартість однієї години оренди студії звукозапису (10 – 200 грн);

загальний час оренди студії звукозапису (1 – 200 годин);

конструктор із значеннями полів за замовчуванням;

метод для установки значень полів;

метод для друку значень полів.

(Перелік полів і методів може бути доповнений користувачем за узгодженням з викладачем.)

Написати програму-драйвер, яка створює три екземпляри шаблону класу "Пісня". Програма повинна роздруковувати значення полів створених об'єктів, потім встановлювати нові значення полів кожного з об'єктів і знов роздруковувати результат.

Варіант № 11 – Варіант № 25

Побудувати ієрархію класів відповідно до варіанту завдання:

Варіант № 11

Студент, викладач, персона, завідувач кафедри

Варіант № 12

Службовець, персона, робітник, інженер

Варіант № 13

Робітник, кадри, інженер, адміністрація

Варіант № 14

Деталь, механізм, виріб, вузол

Варіант № 15

Організація, компанія страховки, нафтогазова компанія

Варіант № 16

Журнал, книга, друкарське видання, підручник

Варіант № 17

Тест, іспит, випускний іспит, випробування

Варіант № 18

Місце, область, місто, мегаполіс

Варіант № 19

Іграшка, продукт, товар, молочний продукт

Варіант № 20

Квитанція, невігідна, документ, рахунок

Варіант № 21

Автомобіль, потяг, транспортний засіб, експрес

Варіант № 22

Двигун, двигун внутрішнього згоряє, дизель-двигун

Варіант № 23

Республіка, монархія, королівство, держава

Варіант № 24

Ссавець, парнокопитний, птах, тварина

Варіант № 25

Корабель, пароплав, парусник, корвет

Завдання 2

Розробити програмне забезпечення оболонки для мультимедійної презентації умовного курсового проекту. Мультимедійні компоненти і сценарій презентації треба погоджувати з викладачем.

Деталізація завдання:

Необхідно розробити графічний Windows-додаток, що складається із заставки і декількох діалогових вікон та здійснює виклик з головного діалогового вікна інші заздалегідь задані додатки – мультимедійні компоненти.

Мета. Досліджувати програмні засоби підтримки процесноорієнтованої багатозадачності в багатомодульних C#-проектах. Закріпити практичні навички розробки багатовіконних Windows-додатків засобами Windows Form Design.

Після виконання завдання 2 студент повинен уміти використовувати основні управляючі елементи Windows Form Design середовища Visual Studio .NET для побудови найпростішої мультимедійної оболонки, а також застосовувати програмні засоби підтримки процесноорієнтованої багатозадачності в багатомодульних C#-проектах.

Для виконання завдання слід залучати роздавальний матеріал до лабораторних робіт № 6, № 7, № 8.

Як індивідуальні варіанти графічних мультимедійних компонентів оболонки рекомендується використовувати 5–7 екранних форм, розроблених у рамках виконання першого самостійного завдання.

Термін задачі – 16 навчальний тиждень.

Вимоги до звіту

Звіт з описом результату виконання самостійного завдання має включати наступні розділи.

Титульна сторінка. Повинна містити назву університету; назву кафедри; назву навчальної дисципліни; тему ІНДЗ; прізвище, ініціали студента, номер академічної групи; дату подання ІНДЗ викладачеві на перевірку (день, місяць, рік).

Зміст. Він відтворює назви розділів, параграфів тощо, які розкривають тему ІНДЗ, з зазначенням номерів сторінок, на яких вони розміщені.

Вступ. У "Вступі" студентом розкривається сутність і стан задачі дослідження та її значущість, підстави та вихідні дані для розробки теми ІНДЗ.

Основна частина. Складається з трьох розділів.

Перший розділ повинен містити словесне описання алгоритму рішення заданої задачі. Треба звернути увагу на особливі випадки, які можуть зустрітися при обробці інформації.

Другий розділ є аналітичною частиною ІНДЗ і повинен містити інформацію щодо обґрунтування вибору конкретного алгоритму серед можливих варіантів. У цьому розділу треба дати відповідні схеми алгоритмів та ув'язати їх з першим розділом звіту.

У третій розділ поміщуються усі машинні роздруківки з початковим текстом програми та результатом її роботи. Програма має включати відповідні коментарі.

Висновки. У висновках викладають перелік пропозицій і рекомендацій та практичні результати, одержані в ІНДЗ. Далі формулюють висновки щодо практичного використання здобуття результатів.

Список літератури. Джерела розміщувати в списку в алфавітному порядку прізвищ перших авторів або заголовків. Відомості про джерела, які включені до списку, необхідно давати згідно з вимогами державного стандарту з обов'язковим наведенням праць (Як приклад дивись список літератури до даної робочої програми).

7.Контрольні запитання для самодіагностики

Змістовний модуль 1. Вступ в об'єктно-орієнтоване програмування

Тема 1. Базові концепції об'єктно-орієнтованого програмування (основні поняття)

Що означає поняття об'єкта і об'єктної моделі?

Дайте визначення об'єкта.

Наведіть основні принципи ООП.

Опишіть приклад елементів об'єктної моделі і її властивостей.

Що таке абстрагування? Наведіть приклад.

Що таке інкапсуляція? Наведіть приклад.

Що таке ієрархія? Наведіть приклад.

Чому доцільно застосування уніфікованої мови моделювання (UML) для опису класів і об'єктів?

Чим відрізняються властивості і поля?

Як організується доступу до стану об'єкта?

Що розуміється під життєвим циклом об'єкта?

Яке призначення конструкторів та деструкторів?

Що означає термін "інтерфейс".

У чому суть спадкоємства?

Наведіть режими доступу до елементів даного класу.

Що таке віртуальні елементи базового класу?

Навіщо потрібні абстрактні базові класи?

Наведіть приклади поліморфізму.

Які взаємостосунки між об'єктами ви знаєте?

Що таке перевантаження операторів?

Що розуміється під подією? Наведіть приклади.

Дайте порівняння ефективності методик "заучування" і "вивчення" ООП.

Тема 2. Вступ у класи

Дайте визначення класу.

Синтаксис класу. Наведіть приклад у вигляді відповідного коду.

Поля класу. Доступ до полів. Наведіть приклади.

Опишіть методи класу. Як реалізується доступ до методів?

Що таке методи-властивості?

Навіщо потрібні індексатори?

Охарактеризуйте статичні поля і методи класу. Коли їх доцільно примінять?

Наведіть приклад визначення константи у класі.

Наведіть приклад визначення конструктора і деструктора.

Які майстри Visual Studio для роботи з елементами класу ви знаєте?

Тема 3. Структури і переліки в класах

У чому різниця між класами і структурами?

Опишіть розгорнені і посилальні типи.

Який синтаксис структур. Наведіть приклад.

Що таке вбудовані структури?

Коли доцільно застосовувати переліки?

Синтаксис переліків. Наведіть приклади оголошень класів-переліків.

Тема 4. Відносини між класами

Які типи відносин між класами ви знаєте?

Що означає відношення "клієнт-постачальник"?

Що означає відношення "батько-спадкоємець"?

Наведіть приклад простої ієрархії класів.

У чому суть використання захищеного доступу?

Як пов'язані конструктори і спадкоємство?

Наведіть приклад віртуальних методів і їх перевизначення.

Наведіть приклад опису абстрактного класу.

Що означає заховання методів базового класу?

Як реалізується виклик перевизначених або прихованих методів базового класу.

Коли доцільно застосовувати ключове слово `this`. Наведіть приклад.

Які механізми, що забезпечують поліморфізм, ви знаєте?

Тема 5. Інтерфейси

Синтаксис визначення інтерфейсів. Наведіть приклад.

Дайте порівняння інтерфейсів і абстрактних класів.

Як здійснюється реалізація інтерфейсів у класах?

Що таке явна реалізація елементів інтерфейсу?

Тема 6. Делегати і події

Дайте призначення делегатів.

Який синтаксис оголошення делегатів?

Наведіть приклад використання делегатів.

Як здійснюється застосування делегатів для створення списку ви-
кликів?

Що таке взаємодія об'єктів?

Опишіть алгоритм посилки і отримання повідомлення про подію.

Як пов'язані делегати і події?

Синтаксис оголошення події.

Опишіть особливості обробки подій у середовищі .NET Framework.

Як реалізується використання вбудованого делегата Event-
Handler?

Змістовний модуль 2. Проектування графічних об'єктно-орієнтованих додатків

Тема 7. Вступ у Windows-форми.

Клас Form. Коли його доцільно використовувати?

Опишіть структуру простого Windows-додатка, що заснована на за-
стосуванні вікон.

Дайте аналіз коду стандартної Windows-форми.

Наведіть алгоритм створення управляючих елементів.

Дайте приклад розробки коду управляючого елемента.

Як здійснюється обробка повідомлень у Windows-додатках?

Коли доцільно використовувати вікна повідомлень?

Клас MessageBox. Метод Show(). Наведіть приклади і особливості
їх застосування.

Як здійснюється розробка коду меню головного вікна?

Опишіть класи MainMenu і MenuItem.

Тема 8. Організація інтерфейсу і робота з Designer Forms

Наведіть перелік групи командних об'єктів для створення інтерфей-
су користувача.

Наведіть перелік групи текстових об'єктів.

Наведіть перелік групи перемикачів.

Наведіть перелік групи контейнерів.

Наведіть перелік групи графічних елементів.

Що таке діалогові вікна і коли їх доцільно використовувати.

Яку ієрархія класів елементів управління ви знаєте?

Як реалізується взаємодія форм?

Що таке модальні і немодальні форми?

Наведіть алгоритм передачі інформації між формами.

Як здійснюється додавання і видалення елементів управління в режимі роботи додатка?

У чому суть перевірки значень, що вводяться користувачем? Як ця перевірка організується? Наведіть приклад.

Тема 9. Вступ у графіку

Зробіть огляд висновку графічної інформації. Опишіть клас Graphics.

Навіщо потрібна система координат? Як вона використовується при програмуванні?

Наведіть методи роботи з кольорами.

Які принципи застосовані в механізмі малювання фігур і ліній, та виведення тексту?

Як здійснюється виведення ліній за допомогою класу Pen?

Наведіть приклад коду малювання фігур за допомогою класу Brush.

Як здійснюється графічне виведення тексту за допомогою класу Font?

У чому суть малювання з використанням образів?

Як здійснюється малювання за допомогою класу TextureBrush?

Малювання образів за допомогою класу Pen. Наведіть приклад.

Опишіть графічне виведення тексту за допомогою класу Image.

Навіщо потрібна подвійна буферизація?

Тема 10. Геометричні перетворення

Наведіть приклади матриць перетворення геометричних зображень. У яких випадках їх доцільно застосовувати? Укажіть особливості програмування наведених прикладів.

Дайте перелік можливих типів анімації геометричних об'єктів. За допомогою яких операторів відбувається їх реалізація?

На чому базується синтез цифрових зображень.

Що таке фрактал?

Наведіть приклади застосування фракталів.

8. Індивідуально-консультативна робота

Індивідуально-консультативна робота здійснюється за графіком індивідуально-консультативної роботи у формі: індивідуальних занять, консультацій, перевірки виконання індивідуальних завдань, перевірки та захисту завдань, що винесені на поточний контроль тощо.

Формами організації індивідуально-консультативної роботи є:

а) за засвоєнням теоретичного матеріалу:

консультації індивідуальні (запитання – відповідь);

групові (розгляд типових прикладів – ситуацій);

б) за засвоєнням практичного матеріалу:

консультації індивідуальні і групові;

в) для комплексної оцінки засвоєння програмного матеріалу: індивідуальне здавання виконаних робіт.

9. Методики активізації процесу навчання

При викладанні навчальної дисципліни для активізації навчально-пізнавальної діяльності студентів передбачено застосування таких навчальних технологій, як проблемні лекції, командна робота (в малих групах), мозкові атаки (табл. 5).

Таблиця 5

Використання навчальних технологій для активізації процесу навчання

Методики активізації процесу навчання	Практичне застосування навчальних технологій
1	2
Проблемні лекції направлено на розвиток логічного мислення студентів, коло питань теми обмежується двома-трьома ключовими моментами, використовується досвід закордонних навчальних закладів з роздачею студентам під час лекцій друкованого матеріалу та виділенням головних висновків з питань, що розглядаються. При читанні лекцій студентам даються питання для самостійного розмірковування, на які лектор відповідає сам, не чекаючи відповідей студентів	Проблемна лекція з питання "Роль дисципліни "Основи об'єктно-орієнтованого програмування" у формуванні професійних знань, умінь і навички у фахівця в області виробництва мультимедійних продуктів" (за темою 1)
	Проблемна лекція з питання "Використання класів у програмних середовищах розробки мультимедійних видань" (за темою 8)

1	2
Мозкові атаки – метод розв'язання невідкладних завдань, сутність якого полягає в тому, щоб висловити якомога більшу кількість ідей за дуже обмежений проміжок часу, обговорити і здійснити їх селекцію	Мозкова атака щодо реалізації інтерфейсу користувача мультимедійного додатку (лабораторне заняття № 7) Мозкова атака щодо реалізації графічного наповнення мультимедійного додатку згідно сценарію (лабораторне заняття № 8)
Робота в команді (в малих групах) дає змогу структурувати практичні заняття за формою і змістом, створює можливості для участі кожного студента в роботі за темою заняття, забезпечує формування особистісних якостей та досвіду професійного і соціального спілкування	Робота в малих групах при виконанні лабораторних робіт та самостійних завдань: студенти розбиваються на підгрупи по 2 особи, кожна підгрупа вибирає тему навчальної науково-дослідної роботи та протягом семестру виконує практичні завдання стосовно до обраної тематики

10. Система поточного та підсумкового контролю знань студентів

Система оцінювання знань, вмінь та навичок студентів враховує види занять, які згідно з програмою навчальної дисципліни "Основи об'єктно-орієнтованого програмування" передбачають лекційні та лабораторні заняття, а також самостійну роботу.

Контрольні заходи включають поточний і підсумковий контроль.

Перевірка та оцінювання знань студентів проводиться в наступних формах:

1. Оцінювання знань студента під час лабораторних занять (проводиться за результатами захисту звіту з лабораторної роботи).
2. Оцінювання результатів виконання самостійної роботи.
3. Проведення проміжного тестування (два рази у семестрі – після закінчення першого та другого змістовного модулей).
4. Проведення поточного модульного контролю (два рази у семестрі – після закінчення першого та другого змістовного модулей).

Оцінювання знань студента під час лабораторних занять

Оцінювання знань студента під час лабораторних занять має на меті перевірку рівня підготовленості студента до виконання конкретної роботи.

Оцінювання проводиться за 12-бальною шкалою за такими критеріями:

1) розуміння, ступінь засвоєння теорії та методології проблем, що розглядаються;

2) ступінь засвоєння фактичного матеріалу навчальної дисципліни;

3) ознайомлення з рекомендованою літературою з питань, що розглядаються;

4) вміння поєднувати теорію з практикою при розгляді практичних ситуацій, розв'язанні задач, проведенні розрахунків, при виконанні завдань, винесених для самостійного опрацювання, та завдань, винесених на розгляд в аудиторії;

5) логіка, структура, стиль викладу матеріалу в письмових звітах і при виступах в аудиторії, вміння обґрунтовувати свою позицію, здійснювати узагальнення інформації та робити висновки.

Відсутність тієї або іншої складової знижує оцінку на відповідну кількість балів.

При оцінюванні лабораторних робіт увага також приділяється якості, самостійності та своєчасності здачі виконаних завдань викладачу (згідно з графіком навчального процесу). Якщо якась із вимог не буде виконана, то на розсуд викладача оцінка буде знижена.

Оцінка 12 балів ставиться за глибоке засвоєння програмного матеріалу, застосування для відповіді не тільки рекомендованої, а й додаткової літератури та творчого підходу; чітке володіння понятійним апаратом, методами та методиками досліджень; вміння використовувати їх для виконання конкретних практичних завдань. Оформлення відповіді повинно бути акуратним, логічним та послідовним.

Оцінка 11 балів ставиться за глибоке засвоєння програмного матеріалу, засвоєння рекомендованої літератури; чітке володіння понятійним апаратом, методами та методиками досліджень; вміння використовувати їх для виконання конкретних практичних завдань. Оформлення відповіді повинно бути акуратним, логічним та послідовним.

Оцінка 10 балів ставиться за повне засвоєння програмного матеріалу та рекомендованої літератури; чітке володіння понятійним апаратом, методами та методиками досліджень; вміння використовувати їх для виконання конкретних практичних завдань. Допускаються незначні випадкові погрішності, які суттєво не впливають на повноту та змістовність відповіді.

Оцінка 9 балів ставиться за повне засвоєння програмного матеріалу та наявне вміння орієнтуватися в ньому, усвідомлене застосування знань для розв'язання практичних задач; за умови виконання всіх вимог, які передбачено для оцінки "відмінно", при наявності незначних арифметичних або синтаксичних помилок (тобто методичний підхід до вирішення задачі є правильним, але допущені незначні неточності у розрахунках певних показників) або не зовсім повних висновків за одержаними результатами розв'язання задачі. Оформлення виконаного завдання має бути охайним.

Оцінка 8 балів ставиться за повне засвоєння програмного матеріалу та наявне вміння орієнтуватися в ньому, усвідомлене застосування знань для розв'язання практичних задач. Практичні завдання виконуються в цілому правильно з використанням типового алгоритму, але при їх виконанні студент припускається окремих помилок. Оформлення виконаного завдання має бути охайним.

Оцінка 7 балів ставиться, якщо студент при виконанні практичних завдань ефективно застосовує основні знання навчального матеріалу, що передбачені навчальною програмою. Практичні завдання виконуються в цілому правильно з використанням типового алгоритму, але при їх виконанні студент припускається значних помилок.

Оцінка 6 балів ставиться за недостатнє вміння застосовувати теоретичні знання для розв'язання практичних задач; за умови, якщо завдання в основному виконане та мету завдання досягнуто, а студент при відповіді продемонстрував розуміння основних положень матеріалу навчальної дисципліни.

Оцінка 5 балів ставиться за часткове вміння застосовувати теоретичні знання для розв'язання практичних задач; за умови, якщо завдання частково виконане, а студент при відповіді продемонстрував розуміння основних положень матеріалу навчальної дисципліни.

Оцінка 4 бали ставиться у випадках, якщо студент при виконанні практичних завдань без достатнього розуміння застосовує навчальний

матеріал, припускається суттєвих помилок, стикається з труднощами при аналізі та порівнянні вивчених явищ та процесів.

Оцінка 3 бали ставиться студенту, який не опанував значної частини програмного матеріалу, не може правильно виконати практичні завдання, стикається зі значними труднощами при аналізі вивчених явищ та процесів.

Оцінка 2 бали ставиться студенту, який неопанував програмний матеріал, не може правильно виконати практичні завдання, стикається зі значними труднощами при аналізі вивчених явищ та процесів.

Оцінка 1 бал ставиться за невиконання завдання загалом.

Оцінювання результатів виконання самостійної роботи

Самостійна робота оцінюється за критеріями: самостійності виконання; використання теоретичного матеріалу, вивченого на лекціях; використання практичного досвіду, отриманого на лабораторних заняттях; цілісності, логічності, повноти й якості оформлення звіту.

Проміжний тестовий контроль

Проміжний тестовий контроль проводиться двічі за семестр. При проведенні проміжного тестування визначається рівень знань студентів з теоретичних питань навчальної дисципліни.

Тестові завдання охоплюють провідні теми, які вивчаються в межах навчальної дисципліни "Основи об'єктно-орієнтованого програмування" та згруповані за двома модулями, кожен з яких складається з тестових завдань різного рівня складності.

Формат тестових завдань поділяється на:

– завдання закритої форми із запропонованими відповідями, із яких вибирають одну правильну;

– завдання відкритої форми з вільно конструйованими відповідями.

Тестові завдання розрізняються за принципом побудови відповіді.

Альтернативні тестові завдання передбачають наявність двох варіантів відповіді типу "так – ні"; "правильно – неправильно" тощо. Їх використовують для перевірки правильності вибору або прийняття рішення в згорнутій формі.

Тестові завдання *на відновлення відповідності частин* становлять модифікацію тестових завдань з множинним вибором і подаються у вигляді двох чи більше колонок слів, фраз, графічних зображень, цифрових або літерних позначень тощо.

Тестові завдання *на порівняння і протиставлення (на аналіз взаємозв'язку)* рекомендуються для перевірки вмінь виявляти розпізнавальні ознаки різних явищ, ситуацій тощо.

Тестові завдання з *множинними відповідями* "правильно – неправильно" передбачають, що відповіді або рішення можуть бути тільки правильними або неправильними. За ними тестується глибина знань, розуміння різних аспектів явищ, процесів тощо.

Тестові завдання *на визначення причинної залежності* використовуються при необхідності перевірки розуміння певної причинної залежності між двома явищами. Спочатку необхідно визначити, правильне чи неправильне кожне з двох тверджень окремо, а лише потім, якщо обидва правильні, визначити, правильна чи неправильна причинна залежність між ними.

Тестові завдання *на відтворення правильної послідовності*, які потребують переструктурування даних або елементів будь-якої комбінації, використовуються в разі тестування вмінь та знань правильної послідовності дій (нормативної діяльності), алгоритмів діяльності, технологічних прийомів тощо, а також знань загальноприйнятих формулювань визначень, правил, законів, фрагментів нормативних документів і т. ін.

Тестові завдання *відкритого типу* передбачають вільні відповіді. Вони є завданнями без запропонованих варіантів відповідей і використовуються для виявлення знань термінів, визначень, понять і т. д.

Тестові завдання для проміжного тестового контролю обираються за відповідними модулями з загального переліку.

Тестове завдання містить запитання одиничного і множинного вибору щодо перевірки знань основних категорій навчальної дисципліни.

Для оцінювання рівня відповідей студентів на тестові завдання використовуються такі критерії; оцінка "відмінно" – 90 – 100 % правильних відповідей; оцінка "добре" – 75 – 89 % правильних відповідей; оцінка "задовільно" – 55 – 74 % правильних відповідей; оцінка "незадовільно" – менше ніж 55 % правильних відповідей.

Поточно-модульний контроль

Поточно-модульний контроль здійснюється та оцінюється за двома складовими: практичний модульний контроль і лекційний модульний контроль. Оцінка за практичну складову модульного контролю виставляється за результатами оцінювання знань студента під час лабораторних занять, виконання завдань самостійної роботи, індивідуального завдання та проміжного тестового контролю згідно з графіком навчального процесу.

Лекційний модульний контроль здійснюється в письмовій формі за відповідними білетами, зміст яких містить питання всіх тем модуля.

Теоретична частина завдання модульного контролю містить: або теоретичні питання, або тестові завдання закритої та відкритої форм.

Практична частина завдання модульного контролю складається з практичних завдань (аналізу або написання невеличких програм) різного ступеня складності.

Зразок модульного завдання

Зразок завдання поточно-модульного контролю "Вступ в об'єктно-орієнтоване програмування"

1. Які типи відносин між класами ви знаєте? Наведіть приклад простої ієрархії класів.

2. Проаналізуйте наведену нижче програму.

```
using System;
class TwoDShape
{
    public double width;
    public double height;
    public void showDim()
    {
        Console.WriteLine("Ширина і висота рівні " +
            width + " і " + height);
    }
}
class Triangle : TwoDShape
{
    public string style
    public double area()
```

```

    {
        return width * height / 2;
    }
    public void showStyle()
    {
        Console.WriteLine("Трикутник " + style);
    }
}
class Shapes
{
    public static void Main()
    {
        Triangle t1 = new Triangle();
        Triangle t2 = new Triangle();
        t1.width = 4.0;
        t1.height = 4.0;
        t1.style = "рівнобедрений";
        t2.width = 8.0;
        t2.height = 12.0;
        t2.style = "прямокутний";
        Console.WriteLine("Інформація про t1: ");
        t1.showStyle();
        t1.showDim();
        Console.WriteLine("Площа рівна " + t1.area());
        Console.WriteLine();
        Console.WriteLine("Інформація про t2: ");
        t2.showStyle();
        t2.showDim();
        Console.WriteLine("Площа рівна " + t2.area());
    }
}

```

2.1. Напишіть на окремому папері результат її роботи.

2.2. Наберіть проаналізовану програму на клавіатурі комп'ютера і запустіть її на виконання.

2.3. Порівняйте отриманий на екрані результат з написаною раніше вашою відповіддю. Проаналізуйте помилки, якщо вони є, і зробіть висновки у вигляді переліку питань, які потребують з вашого боку додаткової уваги.

Нижче (для контролю) наведено отриманий на екрані результат:

Інформація про t1:

Трикутник рівнобедрений

Ширина і висота рівна 4 і 4

Площа рівна 8

Інформація про t2:

Трикутник прямокутний

Ширина і висота рівна 8 і 12

Площа рівна 48

Press any key to continue

Критерії оцінювання модульного завдання

Модульне завдання оцінюється за 12-бальною шкалою і структурно складається з двох питань: 1) теоретичного (максимальна оцінка 4 бали); 2) трьох розділів практичного (максимальна оцінка 8 балів).

Теоретичне питання. Максимальна оцінка (4 бали) виставляється за повну і правильну відповідь. При цьому студент повинен продемонструвати упевнене володіння теоретичним матеріалом, здійснити узагальнення інформації і сформулювати правильні висновки.

Оцінка знижується:

на 1 бал, якщо у визначеннях, що приводяться, є незначні неточності;

на 2 бали – при неправильних визначеннях або відсутності прикладу, який ілюструє теоретичні положення;

на 3 бали, якщо є помилки у визначеннях і у прикладі.

Практичне завдання. Максимальна оцінка (8 балів) виставляється за повну і правильну відповідь на усі три підрозділи практичного завдання.

Оцінка знижується:

на 1 бал, якщо у відповіді, що приводиться, на завдання 2.1. є незначні помилки;

на 2 бали – при грубих помилках у відповіді на завдання 2.1;

на 3 бали – при відсутності аналізу отриманого на ПК результату (завдання 2.3);

на 5 балів, якщо результати виконання завдань 2.1 і 2.3 не співпадають;

на 6 балів – за відсутності коду початкової програми, що відкомпілювався (завдання 2.2).

Підсумкова оцінка з навчальної дисципліни складається як середня з кількох складових, що враховує оцінки кожного виду контролю (дві оцінки за результатами поточного модульного контролю за роботу протягом семестру).

Підсумкова оцінка з навчальної дисципліни згідно з Методикою переведення показників успішності знань студентів університету в систему оцінювання за шкалою ECTS конвертується в підсумкову оцінку за шкалою ECTS (табл. 6).

Таблиця 6

Переведення показників успішності знань студентів у систему оцінювання за шкалою ECTS

Відсоток студентів, які зазвичай досягають відповідної оцінки	Оцінка за шкалою ECTS		Оцінка за бальною шкалою, що використовується в ХНЕУ	Оцінка за національною шкалою
1	2	3	4	5
10	відмінне виконання	A	12 – 11	відмінно
25	вище середнього рівня	B	10	
30	взагалі робота правильна, але з певною кількістю помилок	C	9 – 7	добре
25	непогано, але зі значною кількістю недоліків	D	6	задовільно
10	виконання задовольняє мінімальні критерії	E	5 – 4	
–	потрібне повторне перескладання	FX	3	незадовільно

11. Рекомендована література

11.1. Основна

1. Петцолд Ч. Программирование для Microsoft Windows на С#. В 2-х т. / Ч. Петцолд; Пер. с англ. – М. : Издательско-торговый дом "Русская Редакция", 2002. – 576 с. : ил.
2. Программист – программисту. С# / Карли Ватсон и др.; пер. с англ. СПб. : Издательство "Лори", 2005. – 862 с.
3. С# для профессионалов. В 2-х т. / Симон Робинсон, Олли Корнес, Джей Глинн и др.; пер. с англ. – СПб. : Издательство "Лори", 2003. – 734 с.

11.2. Додаткова

4. Робинсон У. С# без лишних слов / У. Робинсон ; пер. с англ. – М. : ДМК Пресс, 2002. – 352 с.
5. Просиз Д. Программирование для Microsoft NET. / Д. Просиз ; пер. с англ. – Издательство "Русская Редакция", 2003. 704 с.

11.3 Ресурси мережі Інтернет

6. Программирование на языке С# [Электронный ресурс]. – Режим доступа : <http://www.natahaus.ru/>.
7. Особенности реализации графики в Visual Studio 2010 [Электронный ресурс] – Режим доступа : <http://www.intuit.ru/department/pl/visualcsharp/1/2.html>.

Зміст

Вступ.....	3
1. Кваліфікаційні вимоги до студентів	5
2. Тематичний план навчальної дисципліни.....	6
3. Зміст навчальної дисципліни за модулями та темами.....	7
4. План лекцій	11
5. Плани лабораторних занять.....	12
6. Самостійна робота студентів	18
7. Контрольні запитання для самодіагностики	28
8. Індивідуально-консультативна робота.....	32
9. Методики активізації процесу навчання	32
10. Система поточного та підсумкового контролю знань студентів.....	33
11. Рекомендована література.....	42

НАВЧАЛЬНЕ ВИДАННЯ

**Робоча програма
навчальної дисципліни
"ОСНОВИ ОБ'ЄКТНО-ОРІЄНТОВАНОГО
ПРОГРАМУВАННЯ"**

**для студентів напряму підготовки 6.051501
"Видавничо-поліграфічна справа"
всіх форм навчання**

Укладач **Браткевич В'ячеслав В'ячеславович**

Відповідальний за випуск **Пушкар О. І.**

Редактор **Замазій О. Є.**

Коректор **Бриль В. О.**

План 2012 р. Поз. № 372.

Підп. до друку

Формат 60×90 1/16. Папір MultiCopy. Друк Riso.

Ум.-друк. арк. 2,75. Обл.-вид. арк. 3,44. Тираж

прим. Зам. №

Видавець і виготівник – видавництво ХНЕУ, 61166, м. Харків, пр. Леніна, 9а

*Свідоцтво про внесення до Державного реєстру суб'єктів видавничої справи
Дк № 481 від 13.06.2001 р.*