

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

**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ**

**Робоча програма  
навчальної дисципліни  
"СУЧАСНІ ЗАСОБИ ПРОГРАМУВАННЯ"  
для студентів спеціальності 7.05010101  
"Інформаційні управляючі системи і технології"  
всіх форм навчання**

**Харків. Вид. ХНЕУ, 2012**

Затверджено на засіданні кафедри інформаційних систем.  
Протокол № 7 від 14.12.2011 р.

**Укладачі:** Парфьонов Ю. Е.  
Гриньов Д. В.

**P58** Робоча програма навчальної дисципліни "Сучасні засоби програмування" для студентів спеціальності 7.05010101 "Інформаційні управляючі системи і технології" всіх форм навчання / укл. Парфьонов Ю. Е., Гриньов Д. В. – Х. : Вид. ХНЕУ, 2012. – 26 с. (Укр. мов.)

Подано тематичний план навчальної дисципліни та її зміст за модулями й темами, вміщено плани лекцій і лабораторних занять, матеріал щодо закріплення знань, а саме: самостійну роботу, контрольні запитання, а також методичні рекомендації та оцінювання знань студентів.

Рекомендовано для студентів спеціальності 7.05010101 "Інформаційні управляючі системи і технології".

## Вступ

Широкі можливості інформаційних систем у питаннях збору, обробки та видачі необхідної інформації здатні значно підвищити якість економічних розрахунків, зробити більш ефективним процес обґрунтування економічних рішень. Але використання потужних комп'ютеризованих засобів неможливо без програмного забезпечення.

Таким чином, важливість галузі розроблення програмного забезпечення збільшується, оскільки тенденції розвитку комп'ютерної техніки свідчать про те, що з одного боку складність та функціональні можливості комп'ютерної техніки постійно і швидко зростають, а з другого боку, це потребує більш досконалих програмних засобів для задоволення потреб користувачів. Це вимагає від відповідних фахівців глибоких знань щодо використання новітніх інформаційних технологій для створення програмного забезпечення інформаційних систем.

У теперішній час більшість інформаційних систем є розподіленими. Вони будуються на основі гетерогенних або різнорідних мереж, в які можуть входити комп'ютери, що мають різну архітектуру і операційні системи. Одними з найпоширеніших сучасних технологій створення розподілених програмних систем є технології, що використовують програмну платформу Microsoft .NET: Windows Communication Foundation, ASP .NET Web Services та інші.

**Метою навчальної дисципліни** є засвоєння необхідних знань щодо основних сучасних технологій створення розподілених додатків, які базуються на платформі Microsoft.NET, а також удосконалення практичних навичок з використання .NET Framework під час розробки розподілених додатків.

**Предметом вивчення дисципліни** є методи використання технологій .NET Framework під час розробки розподілених додатків.

Необхідним елементом успішного засвоєння навчального матеріалу дисципліни є самостійна робота студентів з технічною літературою, та сучасними програмними засобами розроблення програм.

Структура навчальної дисципліни "Сучасні засоби програмування" наведена в табл.1.

У процесі навчання студенти отримують необхідні знання під час проведення аудиторних занять: лекційних та лабораторних. Велике значення в процесі вивчення та закріплення знань має самостійна робота студентів.

## Структура навчальної дисципліни

Характеристика дисципліни: підготовка спеціалістів	Напрямок, галузь знань, освітньо-кваліфікаційний рівень	Характеристика навчальної дисципліни
Кількість кредитів, відповідних до ECTS – 3; у тому числі: змістовних модулів – 2; самостійна робота	Шифр та назва галузі знань: 0501 "Інформатика та обчислювальна техніка"	Обов'язкова Рік підготовки: 5. Семестр: 9
Кількість годин: усього – 108; за змістовними модулями: модуль 1 – 55 год.; модуль 2 – 53 год.	Шифр та назва спеціальності: 7.05010101 "Інформаційні управляючі системи і технології"	Лекції (теоретична підготовка) – 17 годин. Лабораторні заняття – 34 години. Самостійна робота – 57 годин у тому числі: Поточні консультації – 6
Кількість тижнів викладання дисципліни: 17. Кількість годин за тиждень: 2; 4	Освітньо-кваліфікаційний рівень: спеціаліст	Вид контролю: іспит

Зміст усіх видів занять розроблено відповідно до вимог кредитно-модульної системи організації навчального процесу.

### 1. Кваліфікаційні вимоги до студентів

Дисципліна "Сучасні засоби програмування" є обов'язковою для підготовки спеціалістів за спеціальністю "Інформаційні управляючі системи і технології".

**Необхідна навчальна база перед початком вивчення дисципліни:** з метою найкращого засвоєння матеріалу студенти повинні до початку вивчення дисципліни засвоїти теоретичні знання та опанувати практичні вміння з дисциплін "Організація баз даних та знань", "Web-програмування".

У результаті вивчення даної дисципліни студенти повинні:

**знати:**

основні принципи, концепції та сучасні проблеми щодо розподілених програмних систем;

архітектуру та стандарти розподілених компонентних моделей на платформі Microsoft .NET;

особливості використання збірок на платформі Microsoft .NET;

принципи використання метаданих та доменів додатків на платформі MS .NET;

основні елементи механізму "віддзеркалення" на платформі MS .NET;

основи використання технологій .NET Remoting, ASP .NET Web Service, Windows Communication Foundation під час розробки розподілених додатків;

принципи використання технології Silverlight під час розробки графічного інтерфейсу користувача для клієнтської частини розподілених додатків.

інструменти для розроблення розподілених програмних систем на платформі Microsoft .NET;

**отримати такі компетентності:**

здатність застосовувати технології, методи проектування та інструменти для розроблення розподілених додатків на платформі Microsoft .NET;

здатність критично вивчати, аналізувати і оцінювати з різних точок зору технології, процеси, методи та інструменти для проектних робіт, що пов'язані з розробленням розподілених програмних систем, особисті знання і потребу в інших знаннях, перспективи та загальні тенденції щодо розподілених програмних систем;

здатність вивчати нові технології, методи та прийоми щодо розроблення розподілених програмних систем, а також критично аналізувати їх для професійної роботи.

## **2. Тематичний план навчальної дисципліни**

Під час вивчення дисципліни "Сучасні засоби програмування" студент має ознайомитися з програмою дисципліни, її структурою, формами та методами навчання, видами та методами контролю знань.

Тематичний план дисципліни "Сучасні засоби програмування" складається з двох модулів, кожний з яких об'єднує у собі відносно окремий самостійний блок дисципліни, який логічно пов'язує кілька навчальних елементів дисципліни за змістом та взаємозв'язками (табл. 2).

### Структура тематичного плану навчальної дисципліни

Тема	Кількість годин		
	Лекції	Лабораторні заняття	Самостійна робота
<b>Модуль 1. Особливості використання мови С# для розроблення розподілених додатків</b>			
Тема 1. Огляд нових елементів мови С# та платформи Microsoft .NET	2	6	9
Тема 2. Особливості використання модулів компіляції на платформі Microsoft .NET	2	6	11
Тема 3. Створення розподілених додатків на платформі Microsoft .NET	4	6	9
Разом годин за модулем	8	18	29
<b>Модуль 2. Основні технології розроблення розподілених додатків на платформі Microsoft .NET</b>			
Тема 4. Web-служби	2	4	7
Тема 5. Використання технології Windows Communication Foundation	2	4	7
Тема 6. Використання технології Microsoft Silverlight	5	8	14
Разом годин за модулем	9	16	28
Всього годин	17	34	57

### 3. Зміст навчальної дисципліни за модулями та темами

#### Модуль 1. Особливості використання мови С# для розроблення розподілених додатків

##### Тема 1. Огляд нових елементів мови С# та платформи Microsoft .NET

Неявна типізація локальних змінних. "Автоматичні" властивості. Методи "розширення". Ініціалізатори об'єктів. Типи даних System.Numerics.BigInteger та System.Numerics.Complex. Вбудовані запити LINQ: призначення, основні операції, варіанти використання.

## **Тема 2. Особливості використання модулів компіляції на платформі Microsoft .NET**

Модулі компіляції в середовищі CLR. Структура модулю компіляції. Маніфест модуля компіляції. Приватні та спільні модулі компіляції. Глобальний кеш модулів компіляції. Модулі компіляції зі "строгими" іменами. Створення спільних модулів компіляції. Безпека модулів компіляції. Управління версіями модулів компіляції.

Метадані типів. Отримання інформації щодо типів під час виконання. Віддзеркалення. Віддзеркалення методів, полів, властивостей структур, класів та інтерфейсів.

Динамічні компонувальні блоки, динамічне зв'язування. Динамічне завантаження та використання типів.

Процеси та потоки виконання. Взаємодія з процесами платформи Microsoft .NET: читання даних конкретного процесу, інформація про набір модулів процесу, початок і зупинка процесу програмними засобами.

Домени додатків на платформі Microsoft .NET. Переваги ізоляції додатків. Програмне створення нових доменів додатків. Програмне вивантаження доменів додатків. Завантаження збірок у домен додатку.

Основні властивості й методи класів System.AppDomain, System.Type, System.Reflection.Assembly.

## **Тема 3. Створення розподілених додатків на платформі Microsoft .NET**

Серіалізація та десеріалізація у .NET. Створення класів, об'єктів, які можливо серіалізувати. Процеси серіалізації та десеріалізації. Формати серіалізації. Вибір формату серіалізації.

Віддалена взаємодія на платформі .NET. Архітектура віддаленої взаємодії у .NET. Простір імен стосовно віддаленої взаємодії .NET. Агенти та повідомлення. Канали. Терміни віддаленої взаємодії .NET. Варіанти маршалінга для віддаленого об'єкта. Режими активації віддаленого об'єкту. Запроси та відповіді в .NET Framework. Використання елементів простору імен System.Net для виконання запитів до серверу.

Сокети. Типи сокетів. Протоколи, які можуть використовувати сокети. Створення клієнтського сокету. Створення серверного сокету. Використання синхронних та асинхронних серверних сокетів.

## **Модуль 2. Основні технології розроблення розподілених додатків на платформі Microsoft .NET**

### **Тема 4. Web-служби**

Web-служби та платформа Microsoft .NET. Основні відомості про веб-служби XML в ASP.NET. Роль та переваги Web-служб XML. Компоненти Web-служб XML. Простір імен System.Web.Services. Особливості створення Web-методів. Створення програмних клієнтів для взаємодії з Web-службами. Робота з Web-посиланнями в Microsoft Visual Studio. Доступ до клієнтських типів Web-методів. Доступ до масивів. Доступ до структур.

### **Тема 5. Використання технології Windows Communication Foundation**

Програмні інтерфейси розподілених обчислень: роль DCOM, роль служб COM+/Enterprise Services, роль MSMQ, роль .NET Remoting, роль Web-служб XML, іменовані канали, сокети і P2P.

Основи WCF. Принципи побудови сервіс-орієнтовної архітектури.

Основні зборки WCF. Базова архітектура додатку WCF. Побудова служби WCF. Хостинг служби WCF.

Поняття кінцевої точки. Структура кінцевої точки. Поняття контрактів, прив'язок та адресів WCF. Типи прив'язок. Створення кінцевої точки в коді.

Побудова клієнтського додатку WCF. Хостинг служби WCF у вигляді служби Windows.

Налаштування служб WCF з використанням файлів конфігурації.

Асинхронний виклик служби.

Контракти даних WCF та їх використання. Створення базового контракту даних для класу або структури.

### **Тема 6. Використання технології Microsoft Silverlight**

Основи технології Silverlight. Архітектура Silverlight. Використання мови XAML у додатках Silverlight. Елементи управління. Прив'язка до даних. Події. Використання стилів, графіки, трансформацій та анімації в додатках Silverlight.

Побудова служби WCF для клієнтів Silverlight. Взаємодія клієнта Silverlight із службою WCF.



## **4. Плани лекцій**

### **Модуль 1. Особливості використання мови С# щодо розроблення розподілених додатків**

#### **Тема 1. Огляд нових елементів мови С# та платформи Microsoft .NET**

Вступ до дисципліни.

Нові елементи мови С# і платформи Microsoft. NET.

**Література:** основна [1; 2].

#### **Тема 2. Особливості використання модулів компіляції на платформі Microsoft .NET**

Модулі компіляції в NET.

Домени додатків в .NET.

Використання віддзеркалення.

**Література:** основна [1; 2], додаткова [3].

#### **Тема 3. Створення розподілених додатків на платформі Microsoft .NET**

Серіалізація та десеріалізація у .NET.

Проміжні середовища.

Віддалена взаємодія .NET.

Маршалінг та активація віддалених об'єктів.

Запроси та відповіді в .NET Framework.

Використання сокетів.

**Література:** основна [1; 2], додаткова [4].

### **Модуль 2. Основні технології розроблення розподілених додатків на платформі Microsoft .NET**

#### **Тема 4. Web-служби**

Загальні відомості про Web-служби.

Архітектура Web-служб XML.

Розроблення та використання Web-служб ASP .NET.

**Література:** основна [1].

## **Тема 5. Використання технології Windows Communication Foundation**

Основні елементи технології WCF.

Кінцеві точки: контракти, прив'язки, адреси.

Розроблення та використання служби WCF.

**Література:** основна [1; 2].

## **Тема 6. Використання технології Microsoft Silverlight**

Основні елементи технології Silverlight.

Використання мови XAML.

Стили, графіка, трансформації та анімація.

Елементи управління.

Взаємодія служби WCF та клієнта Silverlight.

**Література:** основна [2].

## **5. Плани лабораторних занять**

Лабораторне заняття – це організаційна форма навчального заняття, на якому студенти під керівництвом викладача формують уміння й навички з практичного застосування основних теоретичних положень навчальної дисципліни шляхом виконання завдань до лабораторних робіт.

Лабораторні заняття з дисципліни "Сучасні засоби програмування" проводяться в спеціально обладнаному навчальному класі з використанням комп'ютерного устаткування, що пристосоване до умов навчального процесу.

З метою підвищення якості навчального процесу, під час проведення лабораторного заняття призначається ще один викладач і навчальна група ділиться на дві підгрупи. Кожний студент працює самостійно, виконуючи індивідуальне завдання для лабораторного дослідження.

Лабораторне заняття включає проведення поточного контролю підготовленості студентів до виконання конкретної лабораторної роботи, виконання завдань теми заняття, оформлення індивідуального звіту з виконаної роботи та його захист перед викладачем.

Виконання лабораторної роботи оцінюється викладачем. Підсумкові оцінки, що отримані студентом за виконання лабораторних робіт, враховуються під час виставлення семестрової підсумкової оцінки з дисципліни.

Тематика проведення лабораторних занять наведена у табл. 3.

## План проведення лабораторних занять

Назва теми	Назва лабораторного заняття та питання, що опрацьовуються	Кількість годин	Література
<b>Модуль 1. Особливості використання мови С# щодо розроблення розподілених додатків</b>			
Тема 1. Огляд нових елементів мови С# та платформи Microsoft .NET	1. Використання нових елементів мови С# та платформи Microsoft .NET у розробленні додатків	6	
Тема 2. Особливості використання модулів компіляції на платформі Microsoft .NET	2. Використання віддзеркалення та доменів додатків	6	
Тема 3. Розподілені додатки на платформі Microsoft .NET	3. Розроблення додатків з використанням мережних бібліотек платформи Microsoft .NET	6	
<b>Модуль 2. Основні технології розроблення розподілених додатків на платформі Microsoft .NET</b>			
Тема 4. Web-служби	4. Розроблення та використання Web-служб	4	
Тема 5. Використання технології Windows Communication Foundation	5. Використання технології Windows Communication Foundation у розподілених додатках	4	
Тема 6. Використання технології Microsoft Silverlight	6. Використання технології Microsoft Silverlight у розподілених додатках	8	
Разом годин		34	

## 6. Самостійна робота студентів

### 6.1. Основні форми самостійної роботи студентів

Для опанування матеріалу дисципліни "Сучасні засоби програмування" окрім лекційних та лабораторних занять, тобто аудиторної роботи, значну увагу необхідно приділяти самостійній роботі.

Основні форми самостійної роботи студента:

1. Вивчення лекційного матеріалу.
2. Вивчення окремих тем або питань, що передбачені для самостійного опрацювання.
3. Вивчення основних термінів та понять за темами дисципліни.
4. Підготовка до лабораторних занять.
5. Контрольна перевірка кожним студентом знань за питаннями для самодіагностики.
6. Підготовка до проміжного та підсумкового модульного контролю.
7. Систематизація вивченого матеріалу перед іспитом.
8. Виконання індивідуального навчально-дослідного завдання.
9. Оформлення звітів з лабораторних робіт.
10. Робота з опрацювання та вивчення рекомендованої літератури.
11. Робота над курсовим проектом.

### 6.2. Питання для самостійного опрацювання

#### **Модуль 1. Особливості використання мови C# щодо розроблення розподілених додатків**

##### **Тема 1. Огляд нових елементів мови C# та платформи Microsoft .NET**

1. Елементи класу Regex та інших класів, що пов'язані з регулярними виразами.
2. Використання регулярних виразів у розробленні програм розбору текстів і пошуку за зразком.
3. Особливості створення та використання вбудованих запитів LINQ для роботи з основними джерелами даних.
4. Практичне застосування методів "розширення".

**Література:** основна [1; 2], додаткова [3].

## **Тема 2. Особливості використання модулів компіляції на платформі Microsoft .NET**

1. Властивості збірок.
2. Багатофайлові збірки.
3. Засоби створення цифрового підпису збірок.
4. Пізні зв'язування.
5. Способи застосування віддзеркалення.
6. Динамічне завантаження збірок.
7. Настроювання доменів додатків.
8. Диспетчері доменів додатків.

**Література:** основна [1; 2], додаткова [3].

## **Тема 3. Створення розподілених додатків на платформі Microsoft .NET**

1. Використання форматів серіалізації.
2. Особливості віддаленої взаємодії .NET.
3. Варіанти маршалінга для об'єктів.
4. Використання TCP-каналів.
5. Адресація за протоколом IPv6.
6. Конфігурування Internet-додатків.

**Література:** основна [1; 2], додаткова [4].

## **Модуль 2. Основні технології розроблення розподілених додатків на платформі Microsoft .NET**

### **Тема 4. Web-служби**

1. Особливості Web-служб XML.
2. Використання мови WDSL.
3. Доступ до клієнтських типів Web-методів. Доступ до масивів.

Доступ до структур.

**Література:** основна [1].

### **Тема 5. Використання технології Windows Communication Foundation**

1. Використання контрактів даних.
2. Хостинг служби WCF.
3. Налаштування служб WCF з використанням файлів конфігурації.

4. Інтеграція Windows Communication Foundation з іншими технологіями Microsoft.

5. Використання класу повідомлень.

6. Фільтрація повідомлень.

**Література:** основна [1, 2].

### **Тема 6. Використання технології Microsoft Silverlight**

1. Модель додатку Silverlight.

2. Використання стилів Silverlight.

3. Використання графіки в додатках Silverlight.

4. Побудова служби для клієнтів Silverlight.

5. Сумісність з технологією WPF.

6. Підтримка позабраузерного використання технології Silverlight.

**Література:** основна [2].

#### *6.3. Контрольні запитання для самодіагностики*

## **Модуль 1. Особливості використання мови С# щодо розроблення розподілених додатків**

### **Тема 1. Огляд нових елементів мови С# та платформи Microsoft .NET**

1. Властивості й методи класу Regex та інших класів, що пов'язні з регулярними виразами.

2. Область використання регулярних виразів.

3. Що таке "автоматичні властивості"?

4. Використання ініціалізаторів об'єктів.

5. Що таке запити LINQ?

6. Для чого використовується неявна типізація локальних змінних?

7. Призначення типу даних System.Numerics.BigInteger

8. Призначення типу даних System.Numerics.Complex

**Література:** основна [1; 2], додаткова [3; 5].

## **Тема 2. Особливості використання модулів компіляції на платформі Microsoft .NET**

1. Які елементи входять до складу збірки?
2. Призначення маніфесту модуля компіляції.
3. У чому полягає різниця між приватними та спільними збірками?
4. Призначення глобального кешу модулів компіляції.
5. Де знаходиться глобальний кеш модулів компіляції в файловій системі операційної системи Windows?
6. Як створити Модуль компіляції зі "строгим" іменем?
7. У чому полягає управління версіями модулів компіляції?
8. Рефлексія типів.
9. Пізні зв'язування.
10. Як завантажити збірку під час виконання?
11. Співвідношення між процесами, потоками виконання та доменами додатків.
12. Призначення доменів додатків.
13. Як створити домен додатку?

**Література:** основна [1; 2], додаткова [3].

## **Тема 3. Розподілені додатки на платформі Microsoft .NET**

1. Які формати серіалізації існують на платформі Microsoft .NET?
2. Як створити клас, об'єкти якого можливо серіалізувати?
3. Які є варіанти маршалінга віддалених об'єктів?
4. Варіанти активації віддалених об'єктів.
5. Варіанти конфігурації WCO-типу.
6. Характеристики MBR-об'єктів.
7. Які існують способи виконання запитів до серверу з використанням елементів простору імен System.Net?
8. Що таке сокет?
9. Клієнтські та серверні сокети.
10. Які протоколи можуть використовувати сокети?
11. Адресація за протоколом IPv6.

**Література:** основна [1; 2], додаткова [4].

## **Модуль 2. Основні технології розроблення розподілених додатків на платформі Microsoft .NET**

### **Тема 4. Web-служби**

1. Для чого призначені Web-служби?
2. Компоненти Web-служби XML.
3. Переваги та недоліки Web-служб XML.
4. Що необхідно зробити щоб метод Web-служби став доступним для використання в клієнтському додатку?
5. Правила створення клієнтських додатків.
6. Для чого використовується мова WDSL?
7. Для чого використовуються клієнтські проху-класи?
8. Коли створюється об'єкт клієнтського проху-класу?

**Література:** основна [1].

### **Тема 5. Використання технології Windows Communication Foundation**

1. Для чого необхідна технологія WCF?
2. Як можна виконати налаштування служби WCF.
3. Структура кінцевої точки.
4. Які типи прив'язок існують?
5. Призначення контрактів даних.
6. Взаємодія з віддаленим типом.
7. Основні збірки WCF.
8. Склад додатку WCF.
9. Хостинг служби WCF.
10. Які протоколи можуть використовувати служби WCF.
11. Як виконати фільтрацію повідомлень WCF?

Література основна [1, 2].

### **Тема 6. Використання технології Microsoft Silverlight**

1. Призначення технології Silverlight.
2. Для чого використовується мова XAML у додатках Silverlight?
3. Які елементи управління-контейнери існують?
4. Як виконати прив'язку елемента управління до даних?
5. Які види трансформацій існують?
6. Правила побудови служби WCF для клієнтів Silverlight.



7. Як використовуються стилі Silverlight?
8. Як виконати найпростішу анімацію в додатку Silverlight?
9. Як співвідносяться технології WPF та Silverlight?

**Література:** основна [2].

## **7. 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.

З теоретичної частини дисципліни індивідуально-консультативна робота проводиться у вигляді:

а) індивідуальних консультацій, на яких студент отримує відповідь від викладача на конкретні запитання або пояснення певних теоретичних положень чи аспектів їх практичного застосування;

б) групових консультацій, на яких викладач розглядає типові приклади з використання концепцій об'єктно-орієнтованого програмування.

З практичної частини дисципліни індивідуально-консультативна робота проводиться у вигляді:

а) індивідуальних консультацій, на яких викладач розглядає лабораторні завдання, стосовно яких виникли запитання у студента;

б) групових консультацій, на яких викладач розглядає практичні ситуації, які потребують колективного обговорення.

Індивідуально-консультативна робота для комплексної оцінки засвоєння програмного матеріалу проводиться у вигляді:

а) індивідуального захисту самостійних та індивідуальних завдань;

б) підготовки рефератів для виступу на науковому семінарі;

в) підготовки рефератів для виступу на науковій конференції.

## **8. Методики активізації процесу навчання**

Під час викладання навчальної дисципліни "Сучасні засоби програмування" для активізації навчально-пізнавальної діяльності студентів передбачено застосування таких навчальних технологій, як: проблемні лекції, робота в малих групах, семінари-дискусії, презентації (табл. 4).

**Використання навчальних технологій  
для активізації процесу навчання**

Методики активізації процесу навчання	Практичне застосування навчальних технологій
<p>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. Під час читання лекцій студентам даються питання для самостійного розміркування, на які лектор відповідає сам, не чекаючи на відповіді студентів</p>	<p>Проблемна лекція з питання: "Розроблення розподілених додатків на платформі Microsoft .NET" (за темою 3) Проблемна лекція з питання: "Використання технології Windows Communication Foundation" (за темою 5)</p>
<p>Робота в малих групах дає змогу структурувати лабораторні заняття за формою і змістом, створює можливості для участі кожного студента в роботі за темою заняття, забезпечує формування особистісних якостей та досвіду соціального спілкування</p>	<p>Робота в малих групах під час розроблення додатку з використанням віддзеркалення та доменів додатків (Лабораторна робота за модулем 1)</p>
<p>Семінари-дискусії передбачають обмін думками і поглядами учасників з приводу даної теми, а також розвивають мислення, допомагають формувати погляди і переконання, виробляють вміння формулювати думки і висловлювати їх, вчать оцінювати пропозиції інших людей, критично підходити до власних поглядів</p>	<p>Проблемне повідомлення та дискусія під час роботи студентів з питання: "Можливості технології Silverlight" (Лабораторна робота за модулем 2)</p>
<p>Презентації – виступи перед аудиторією, що використовуються для представлення певних досягнень, результатів роботи групи</p>	<p>Презентація розробленої постановки задачі для розв'язання на ПК, архітектури додатку, демонстрація роботи розробленого додатку за даними контрольного прикладу (Лабораторна робота за модулем 1)</p>

## 9. Система поточного та підсумкового контролю знань студентів

Система оцінювання знань, вмінь та навичок студентів передбачає виставлення оцінок за всіма формами проведення занять.

Перевірка та оцінювання знань студентів може проводитись у наступних формах:

1. Оцінювання роботи студентів у процесі лабораторних занять.
2. Проведення поточно-модульного контролю.
3. Проведення підсумкового іспиту.

Загальна модульна оцінка складається з поточної оцінки, яку студент отримує під час лабораторних занять, та оцінки за виконання модульної контрольної роботи.

**Порядок оцінювання знань студентів під час лабораторних занять** проводиться за 12-бальною шкалою за такими критеріями:

а) розуміння, ступінь засвоєння матеріалу навчальної дисципліни;  
б) ознайомлення з рекомендованою літературою, а також із сучасною літературою з питань, що розглядаються;

в) уміння поєднувати теорію з практикою у розробці програм, розв'язанні задач, проведенні розрахунків під час виконання завдань, що винесені для самостійного опрацювання, та завдань, що винесені на розгляд в аудиторії;

г) логіка, структура, якість оформлення, стиль викладу матеріалу в письмових роботах та під час виступів в аудиторії, вміння обґрунтовувати свою позицію, здійснювати узагальнення інформації та робити висновки.

Оцінка "відмінно" (10 – 12 балів) ставиться за умови відповідності звіту з виконаного лабораторного завдання студента та його усної відповіді під час захисту завдання всім чотирьом зазначеним критеріям. Відсутність тієї або іншої складової знижує оцінку на відповідну кількість балів.

Під час оцінювання увага також приділяється якості, самостійності та своєчасності здачі виконаних завдань викладачу (згідно графіку навчального процесу).

### **Проведення поточно-модульного контролю**

Поточно-модульний контроль здійснюється та оцінюється за двома складовими: практичний модульний контроль і теоретичний модульний контроль. Оцінка за практичну складову модульного контролю виставляється за результатами оцінювання знань студента під час лабораторних занять. Теоретичний модульний контроль здійснюється у письмовій формі.

Для підведення підсумків роботи студентів зі змістовного модуля виставляється підсумкова оцінка з поточно-модульного контролю, яка враховує оцінки за практичний та теоретичний модульний контроль.

### Проведення підсумкового іспиту

Іспит здійснюється на комп'ютерах за екзаменаційними білетами. Екзаменаційний білет складається з двох завдань. Результати іспиту оцінюються за 12-ти бальною шкалою. Підсумкова оцінка за іспит є сумою оцінок за кожне завдання. Кожне завдання оцінюється від 0-я до 6-ти балів відповідно наступної шкали (табл. 5):

Таблиця 5

Кількість балів	Результат виконання завдання
6 балів	Завдання виконано в повному обсязі. Програма працює правильно. Інтерфейс та вихідний код програми відповідають встановленим вимогам
5 балів	Завдання виконано. Програма працює правильно, але одна з її функціональних можливостей реалізована з порушенням вимог, які вказані в завданні
4 бала	Завдання в основному виконано. Програма працює, але дві з її функціональних можливостей реалізовані з порушенням вимог, що вказані в завданні
3 бала	Завдання виконано, але не в повному обсязі. Програма працює, але не реалізована одна з функціональних вимог, які вказані в завданні, або три з функціональних можливостей програми реалізовані з порушенням вимог, що вказані в завданні
2 бала	Завдання не виконано. Програма запускається, але не реалізовані дві з функціональних вимог, які вказані в завданні, або чотири з функціональних можливостей програми реалізовані з порушенням вимог, що вказані в завданні
1 бал	Програма запускається та частково відповідає постановці завдання, але не реалізовані більше двох з функціональних вимог, які вказані в завданні, або більше чотирьох з функціональних можливостей програми реалізовані з порушенням вимог, що вказані в завданні. Програма не запускається або завершується аварійно, але є програмний код, який розроблений студентом, який відповідає постановці завдання
0 балів	Програма відсутня. Програма не містить програмного коду, що був розроблений студентом. Програма не відповідає постановці завдання. Програма має явні ознаки несаможитності її розробки

Невиконання або суттєве порушення будь-якої з загальних вимог до програми, наведених далі, зніжує кількість балів по завданню на 1.

### **Вимоги до інтерфейсу користувача**

1. Інтерфейс користувача має відповідати постановці завдання.
2. Назви елементів інтерфейсу повинні бути виконані українською або російською мовою.
3. Інтерфейс консольної програми повинен складатися з текстових повідомлень українською або російською мовою, які відносяться до введення та виведення основних та допоміжних даних.

### **Вимоги до вихідного тексту**

1. Додержання принципу інкапсуляції щодо рівнів доступу до полів, властивостей та методів класів.
2. Вихідний код кожного з класів програми повинен міститися в окремому файлі.
3. Наявність коментарів (для класів – призначення класу; для методів – призначення методу, опис параметрів та значення, що повертається) з обов'язковим використанням відповідних документаційних XML-тегів.

Для підведення підсумків роботи студента з навчальної дисципліни виставляється загальна оцінка, яка враховує оцінки з кожного виду контролю (дві оцінки поточно-модульного контролю за роботу протягом семестру та оцінку за результатами іспиту):

$$П = 0,6 \cdot E + 0,4 \cdot (M1 + M2), \quad (1)$$

де П – підсумкова оцінка з навчальної дисципліни;

Е – екзаменаційна оцінка;

М1, М2 – оцінки за модулями.

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

**Переведення показників успішності знань студентів  
у систему оцінювання за шкалою ECTS**

Відсоток студентів які зазвичай успішно досягають вдалої оцінки	Оцінка за шкалою ECTS		Оцінка за бальною шкалою, що використовується в ХНЕУ	Оцінка за національною шкалою
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 із значною кількістю помилок	D	6	задовільно
10	виконання задовольняє мінімальні критерії	E	5 – 4	
-	потрібне повторне перескладання	FX	3	незадовільно
-	повторне вивчення дисципліни	F	2 – 1	

**Приклад екзаменаційного білета з дисципліни**

Міністерство освіти і науки, молоді та спорту України  
Харківський національний економічний університет

Спеціальність 7.05010101

Дисципліна: "Сучасні засоби програмування" (5 курс)

**ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 14**

Завдання 1.

Розробити програму з графічним інтерфейсом користувача (Windows Forms), яка обчислює площу круга заданого діаметру. Програма не

повинна містити вихідний код методу обчислення. Клас, у якому визначений даний метод, має знаходитися в збірці (dll-бібліотеці), що розміщується в каталозі програми.

Програма повинна містити додатковий вихідний код для завантаження зазначеної збірки під час виконання, використання методу перетворення в програмі. При цьому повинен використовуватися механізм "відзеркалення" платформи MS .Net (простір імен System.Reflection).

Послідовність виконання програми:

1. Завантажити збірку (dll-бібліотеку) з файлу, що був заздалегідь розміщений в каталозі програми.
2. Після введення користувачем вихідних даних створити об'єкт класу, що знаходиться в завантаженій збірці.
3. Викликати метод цього об'єкту, який виконує обчислення.
4. Відобразити результат обчислення площі круга в інтерфейсі програми.

Під час виконання завдання необхідно в MS Visual Studio створити два "рішення": одне для проекту програм (відповідно до її типу), а друге для проекту типу "Class Library".

## Завдання 2.

Задано матрицю (двовимірний масив) цілих чисел, яка складається з 3-х рядків та 4-х стовпчиків.

З використанням WCF-служби розробити розподілений додаток для знаходження середнього арифметичного елементів матриці, які більше за число, яке введене користувачем.

Клієнт WCF-служби – додаток з графічним інтерфейсом користувача (Windows Forms), який дозволяє:

1. З використанням необхідної кількості компонентів TextBox вводити тільки цілочисельні значення елементів матриці в інтервали [- 5, 30], або повідомляти користувача про відповідну помилку (компонент Error Provider).
2. Увести ціле число, що необхідне для виконання операції з матрицею.

3. Викликати метод WCF-служби, що виконує зазначену операцію з матрицею.

4. Уводити вихідні дані (елементи матриці) та результати роботи програми в багаторядковий компонент TextBox.

5. Для організації взаємодії з користувачем використовувати панель меню (компонент MenuStrip) з меню "Файл" (пункти: "Вийти з програми") та меню "Матриця" (пункти: "Обчислити", "Очистити матрицю").

Розроблені до кожного завдання проекти (кожний в своїй папці з іменами "Task 1", "Task 2") зберегти разом в окремій папці на диску.

Затверджено на засіданні кафедри "Інформаційних систем"  
протокол № 7 від 14.12.2011 р.

Зав. кафедрою \_\_\_\_\_ проф. В. С. Пономаренко

Екзаменатор \_\_\_\_\_ доц. Ю. Е. Парфьонов



## **10. Рекомендована література**

### **10.1. Основна**

1. Троелсен Э. Язык программирования C# 2010 и платформа .NET 4.0; / пер. с англ. – М. : ООО "И.Д. Вильямс", 2011 – 1392 с.
2. Маки А. Введение в .NET 4.0 и Visual Studio 2010 для профессионалов / Алекс Маки; пер. с англ. – М. : ООО "И. Д. Вильямс", 2010. – 416 с.

### **10.2. Додаткова**

3. Нэш Т. C# 2010: ускоренный курс для профессионалов / Трей Нэш; пер. с англ. – М. : ООО "И.Д. Вильямс", 2010 – 592 с.
4. Фленов М. Е. Библия C#. / М. Е. Фленов. – Спб. : БХВ-Петербург, 2011 – 560 с.
5. Фридл Дж. Регулярные выражения ; пер. с англ. / Дж. Фридл. – Спб. : Питер, 2003 – 464 с.

### **10.3. Електронні ресурси**

6. База знань Microsoft Developer Network (MSDN). – Режим доступу : <http://msdn.microsoft.com/ru-ru>.
7. Сайт компанії Microsoft щодо технології Silverlight. – Режим доступу : [www.silverlight.net](http://www.silverlight.net).
8. Internet-інститут інформаційних технологій. – Режим доступу : [www.intuit.ru](http://www.intuit.ru).
9. База знань Russian Software Developer Network (RSDN). – Режим доступу : [www.rsdn.ru](http://www.rsdn.ru).

## Зміст

Вступ.....	3
1. Кваліфікаційні вимоги до студентів .....	4
2. Тематичний план навчальної дисципліни.....	5
3. Зміст навчальної дисципліни за модулями та темами .....	6
4. Плани лекцій .....	9
5. Плани лабораторних занять .....	10
6. Самостійна робота студентів.....	12
7. Індивідуально-консультативна робота.....	17
8. Методики активізації процесу навчання.....	17
9. Система поточного та підсумкового контролю знань студентів ...	19
10. Рекомендована література .....	25
10.1. Основна .....	25
10.2. Додаткова.....	25
10.3. Електронні ресурси .....	25

НАВЧАЛЬНЕ ВИДАННЯ

**Робоча програма  
навчальної дисципліни  
"СУЧАСНІ ЗАСОБИ ПРОГРАМУВАННЯ"  
для студентів спеціальності 7.05010101  
"Інформаційні управляючі системи і технології"  
всіх форм навчання**

Укладачі: **Парфьонов Юрій Едуардович**  
**Гриньов Денис Валерійович**

Відповідальний за випуск **Пономаренко В. С.**

Редактор **Шаповал Г. В**

Коректор **Мартовицька-Максимова В. А.**

План 2012 р. Поз. № 290.

Підп. до друку Формат 60 x 90 1/16. Папір MultiCopy. Друк Riso.

Ум.-друк. арк. 1,75. Обл.-вид. арк. 2,19. 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.*