

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ СЕМЕНА КУЗНЕЦЯ**

**Робоча програма
навчальної дисципліни
"ПРОГРАМУВАННЯ ЗАСОБІВ
МУЛЬТИМЕДІА"
для студентів напряму підготовки
6.051501 "Видавничо-поліграфічна справа"
денної форми навчання**

**Харків
ХНЕУ ім. С. Кузнеця
2016**

Затверджено на засіданні кафедри комп'ютерних систем і технологій.
Протокол № 7 від 12.12.2015 р.

Самостійне електронне текстове мережеве видання

Укладач В. В. Браткевич

Робоча програма навчальної дисципліни "Програмування Р 58 засобів мультимедіа" для студентів напряму підготовки 6.051501 "Видавничо-поліграфічна справа" денної форми навчання : [Електронне видання] / уклад. В. В. Браткевич. – Харків : ХНЕУ ім. С. Кузнеця, 2016. – 59 с.

Подано тематичний план навчальної дисципліни та її зміст за модулями й темами. Вміщено плани лекцій та лабораторних занять, матеріали для закріплення знань (завдання для самостійної роботи, контрольні запитання), методичні рекомендації щодо оцінювання знань студентів, професійні компетентності, якими повинен володіти студент після вивчення дисципліни.

Рекомендовано для студентів напряму підготовки 6.051501 "Видавничо-поліграфічна справа" денної форми навчання.

Вступ

Розроблення конкурентоспроможних мультимедійних продуктів практично неможлива без застосування засобів програмування. Це обумовлено тим, що переважна більшість сучасних авторських інструментів проектування мультимедіа (а саме: Adobe Flash, Adobe Director, Adobe Captivate, Microsoft Expression Blend тощо) дозволяють, поряд з візуальним програмуванням, здійснювати ефективне управління мультимедійним контентом за допомогою відповідного коду. Як правило, для цієї мети використовуються С-образні мови програмування, такі, як: С++, С#, Java Script та ін. Саме тому в якості базової мови вивчення в рамках даної дисципліни обрано алгоритмічну мову С#. Такий підхід дає можливість майбутньому фахівцю легко адаптуватися відносно застосування багатьох сучасних середовищ розроблення мультимедійних продуктів.

Структурно програма складається з двох розділів, зміст яких вивчається відповідно в третьому і четвертому семестрах. У першому розділі "Основи програмування" розглядаються питання організації процедурно-орієнтованих програм та особливості оброблення різноманітних типів даних. Другий розділ "Мультимедійні об'єктно-орієнтовані додатки" присвячено вивченню базових концепцій об'єктно-орієнтованого програмування (ООП) та організації мультимедійних програм і даних.

Навчальна дисципліна "Програмування засобів мультимедіа" є базовою навчальною дисципліною вона вивчається згідно з навчальним планом підготовки фахівців освітнього ступеня "бакалавр" напряму підготовки 6.051501 "Видавничо-поліграфічна справа".

1. Опис навчальної дисципліни

Найменування показників	Галузь знань, напрям підготовки, освітній ступінь	Характеристика навчальної дисципліни		
		денна форма навчання		
Кількість кредитів – 10	Галузь знань 0515 "Видавничо-поліграфічна справа"	Базова		
Змістових модулів – 4	Напрямок підготовки 6.051501 "Видавничо-поліграфічна справа"	Рік підготовки		
Загальна кількість годин – 300, 3 семестр – 148 год; 4 семестр – 152 год		2-й		
		Семестр		
		3	4	
		Лекції		
		16 год	18 год	
		Лабораторні		
Тижневих годин для денної форми навчання: аудиторних – 3; самостійної роботи студента – 5,8		34 год		34 год
		Самостійна робота		
		98 год	100 год	
	Вид контролю			
	залік	Екзамен		
	Освітній ступінь: бакалавр			

Примітка. Співвідношення кількості годин аудиторних занять до самостійної роботи становить:

для денної форми навчання – 51,5 %.

2. Мета та завдання навчальної дисципліни

Метою викладання даної навчальної дисципліни є надання студентам теоретичних основ, практичних і методичних рекомендацій щодо застосування принципів об'єктно-орієнтованого програмування в різноманітних галузях видавничо-поліграфічної діяльності, де потрібне розроблення сучасних програмних засобів мультимедіа.

Для досягнення мети поставлені такі основні **завдання**:

засвоєння основних принципів процедурного та об'єктно-орієнтованого стилю програмування;

оволодіння навичками самостійного розроблення консольних та графічних додатків для інструментальної підтримки технологічного процесу виробництва видавничо-поліграфічних та мультимедійних продуктів.

Об'єктом навчальної дисципліни є процес програмування засобів мультимедіа.

Предметом навчальної дисципліни є основні принципи процедурного та об'єктно-орієнтованого програмування; зміст і структура типових бібліотек класів і шаблонів; технології створення консольних та графічних об'єктно-орієнтованих додатків.

Теоретико-методологічною базою вивчення цієї дисципліни є навчальні дисципліни "Інформатика і комп'ютерна техніка" та "Інформаційні технології". У свою чергу, знання з даної дисципліни забезпечують успішне засвоєння таких навчальних дисциплін, як: "Основи проектування Web-видань", "Технології розробки Web-ресурсів", "Мультимедійні технології", "Розробка Web-додатків", а також виконання міждисциплінарних комплексних курсових робіт, бакалаврських і магістерських дипломних робіт.

У процесі навчання студенти отримують необхідні знання під час лекційних занять та виконання лабораторних завдань. Також велике значення в процесі вивчення та закріплення знань має самостійна робота студентів.

У результаті вивчення навчальної дисципліни студент повинен:

знати:

основні парадигми щодо процедурного та об'єктно-орієнтованого стилю програмування;

предметну, забезпечувальну та функціональну технології розроблення консольних та графічних об'єктно-орієнтованих мультимедійних додатків;

типову структуру Windows додатків.

склад та структуру типових бібліотек класів і шаблонів;

можливості застосування сучасних базових інструментальних програмних засобів, призначених для створювання та налагодження мультимедійних додатків;

вміти:

застосувати інтерфейсні інструменти середовища Visual Studio .NET для розроблення програм;

обґрунтовано визначати оптимальні типи даних для реалізації певних мультимедійних сценаріїв;

розробляти лінійні програми з застосуванням арифметичних і логічних операцій;

розробляти програми з застосуванням структур вибору альтернатив;
розробляти програми, які здійснюють оброблення масивів. Реалізувати типові алгоритми пошуку, перетворення та сортування елементів масиву;

розробляти програми, які здійснюють опрацювання структур;

розробляти програми, які використовують функцій;

застосувати методи-властивості, індексатори, статичні поля, методи класу і майстри Visual Studio для роботи з елементами класу;

розробляти програми, які мають ієрархію класів та елементи управління взаємодією форм;

розробляти бізнес-додатки з елементами мультимедіа;

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

У процесі викладання навчальної дисципліни основна увага приділяється оволодінню студентами професійними компетентностями, що наведені в табл. 2.1.

Таблиця 2.1

Професійні компетентності, які отримують студенти після вивчення навчальної дисципліни

Код компетентності	Назва компетентності	Складові компетентності
1	2	3
ПТ*	Здійснювати обґрунтування прийняття рішень щодо вибору відповідних етапів предметної технології реалізації сценаріїв роботи мультимедійних бізнес-додатків	Обґрунтовувати перелік і структуру класів, екземпляри яких найбільш адекватно забезпечують реалізацію заданого мультимедійного сценарію
		Обґрунтовано визначати оптимальні типи даних для реалізації певних мультимедійних сценаріїв
		Обґрунтовано визначати оптимальну кількість методів, що дозволяють спростити структуру програми
		Здійснювати перетворення вихідного словесного опису відповідних методів мультимедійного сценарію до їх формального подання у вигляді відповідних графічних схем

Закінчення табл. 2.1

1	2	3
ЗТ**	Приймати оптимальні рішення у виборі програмних засобів реалізації (забезпечувальна технологія) сценаріїв роботи мультимедійних бізнес-додатків	Здійснювати обґрунтований вибір середовища програмного розроблення мультимедійних продуктів
		Визначати способи пошуку та виправлення синтаксичних помилок і помилок виконання поточних програм
		Здійснювати покрокове тестування програм
		Формувати результат програмування у вигляді ехе-файла і супутнього йому мультимедійного ресурсу
		Настроювати середовище програмування для підтримки розробки консольних і графічних додатків
ФТ***	Розробляти ефективні рішення завдань предметної технології з застосуванням інструментальних засобів розробки мультимедіа (функціональна технологія)	Розробляти лінійні програми з застосуванням арифметичних і логічних операції
		Розробляти програми з застосуванням структур вибору альтернатив
		Розробляти програми, які здійснюють обробку масивів. Реалізувати типові алгоритми пошуку, перетворення та сортування елементів масиву
		Розробляти програми, які здійснюють опрацювання структур
		Розробляти програми, які використовують функцій
		Застосувати методи-властивості, індексатори, статичні поля, методи класу і майстри Visual Studio для роботи з елементами класу
		Розробляти програми, які мають ієрархію класів та елементи управління взаємодією форм
		Розробляти об'єктно-орієнтовані бізнес-додатки з елементами мультимедіа Користуватися раніше складеними програмами і здійснювати супровід програм, внести зміни в програму

* Предметна технологія.

** Забезпечувальна технологія.

*** функціональна технологія.

Структуру складових професійних компетентностей та їх формування відповідно до Національної рамки кваліфікацій України наведено в додатку А.

3. Програма навчальної дисципліни

Розділ 1

Основи програмування

Змістовий модуль 1

Організація процедуро-орієнтованих програм

Тема 1. Теоретичні та методологічні засади організації програм і даних

1.1. Основні концепції і термінологія.

Місце дисципліни "Програмування засобів мультимедіа" у навчальному процесі на пряму підготовки 6.051501 "Видавничо-поліграфічна справа". Принцип програмного управління комп'ютером. Стили програмування: процедурний, логічний, функціональний, об'єктно-орієнтований. Місце С-образних мов програмування в галузі розроблення сучасних мультимедійних продуктів.

1.2. Етапи розроблення програмного забезпечення.

Огляд основних етапів розроблення програмного забезпечення: створення специфікації програми, визначення вимог до програми; проектування програми, розроблення концепції, що дозволяє втілити вимоги специфікації в працюючій програмі; розроблення і написання початкового коду програми; тестування і налагодження програми.

1.3. Огляд середовища розроблення Visual Studio .Net.

Технологія .NET. Особливості інфраструктури .NET-платформи. Мови програмування і компілятори. Традиційний процес компіляції. Компіляція у .NET початкового коду C#.

1.4. Базова структура C#-програми.

Лексичні елементи мови. Алфавіт, синтаксис, семантика. Коментарі, ідентифікатори, службові слова. Дані, змінні, вирази, операнди.

Аналіз початкового коду програми. Приклади роботи з проектами та рішеннями.

Тема 2. Поняття типу даних

2.1. Концепція типу даних.

Класифікація типів даних. Огляд основних типів C#. Прості типи. Типи – переліки. Типи – структури. Типи – класи. Тип string. Типи – масиви. Типи – інтерфейси. Типи – делегати. Єдина система типів .NET (Common Type System – CTS).

2.2. Характеристика й особливості застосування простих типів.

Синтаксис оголошення змінних. Цілочисельні типи. Биті. Знакові і беззнакові цілочисельні типи. Цілочисельні літерали. Оператори привласнення. Перетворення вбудованих типів. Типи з плаваючою крапкою. Інтерпретація експоненціальної нотації. Особливості операцій над числами з плаваючою крапкою. Літерали з плаваючою крапкою. Тип decimal.

2.3. Константні величини.

Класифікація константних величин. Числові, символічні та іменовані константи.

Тема 3. Програмування обчислювальних процесів

3.1. Програмування лінійних обчислювальних процесів

Алгоритмічні структури. Огляд основних операцій C#: арифметичні і логічні операції, операції відносин. Синтаксичні блоки. Пріоритети операцій. Простір імен. Область видимості змінних. Область видимості і час існування змінних. Форматування числових значень. Перетворення типів даних. Приклади лінійних програм.

3.2. Оператори управління програмою.

Поняття потоку управління програмою. Структури вибору альтернатив. Структура вибору if. Структура вибору if / else. Множинний вибір – структура switch. Умовний вираз. Структури повторення. Цикл з перед умовою (while). Цикл з після умовою (do while). Оператори циклу for і foreach.

3.3. Управляючі оператори в циклах

Оператор break. Оператор continue. Особливості застосування оператора continue в циклі for. Оператор goto. Вкладені цикли. Рекомендації з вибору циклів. Приклади виконання завдань до лабораторних занять.

Змістовий модуль 2

Організація й оброблення складених типів даних

Тема 4. Масиви

4.1. Загальні відомості про масиви.

Призначення, оголошення і визначення масиву. Доступ до окремих елементів масиву. Ініціалізація масивів. Перебір елементів масиву за допомогою оператора `foreach`. Розмір і ранг масиву.

4.2. Типові приклади оброблення одновимірних масивів даних.

Алгоритми пошуку, перетворення та сортування елементів масиву.

4.3. Багатовимірні масиви.

Доступ до елементів двовимірного масиву. Подання двовимірного масиву як масиву масивів. Приклади оброблення матриць. Дії з рядками.

Тема 5. Структури

5.1. Загальні відомості про структури.

Призначення, оголошення і визначення структур.

5.2. Приклади елементарного оброблення структур.

Призначення, оголошення і визначення структур. Масиви структур. Особливості оброблення елементів структур. Приклади оброблення масиву структур.

Тема 6. Функції

6.1. Загальні відомості про функції

Призначення, опис та виклик функцій. Значення, що повертаються. Параметри функцій. Відповідність параметрів. Масиви параметрів.

6.2. Обмін інформацією з функцією.

Передача параметрів за посиланням і за значенням. Вихідні зміни. Область дії змінних. Параметри і значення, що повертаються за порівнянням з глобальними даними.

6.3. Функції і структури. Функції і масиви.

Перевантаження функцій. Делегати (посилання на функції). Рекурсія.

6.4. Виняткові ситуації.

Винятки в C#. Генерування виняткових ситуацій. Оброблення виняткових ситуацій. Оформлення блоків `try`, `catch` та `finally`.

6.5. Робота з файлами.

Поняття потоку. Використання бінарних та текстових файлів.

Розділ 2

Мультимедійні об'єктно-орієнтовані додатки

Змістовий модуль 3

Базові концепції об'єктно-орієнтованого програмування (ООП)

Тема 7. Особливості об'єктно-орієнтованого стилю програмування

7.1. Поняття об'єкта й об'єктної моделі.

Визначення об'єкта. Принципи ООП: абстрагування, інкапсуляція, спадкування (ієрархія). Опис елементів об'єктної моделі та її властивостей. Застосування уніфікованої мови моделювання (UML) для опису класів і об'єктів. Об'єкти на C #.

7.2. Визначення класу.

Синтаксис класу. Поля класу. Доступ до полів. Властивості і поля. Організація доступу до стану об'єкта. Життєвий цикл об'єкта.

Тема 8. Методи класів

8.1. Порівняння методів і функцій.

Доступ до методів. Методи-властивості. Індексатори. Статичні поля і методи класу. Метод Main (). Конструктори і деструктори.

8.2. Використання методів.

Обмін інформацією між методами. Рекурсивний виклик методів.

Тема 9. Відносини між класами

9.1. Типи відносин між класами.

Відносини "клієнт – постачальник. Відношення "батько – спадкоємець.

9.2. Спадкоємство.

Проста ієрархія класів. Доступ до елементів класу і спадкоємство. Використовування захищеного доступу. Конструктори і спадкоємство.

9.3. Абстрактні класи.

Визначення абстрактних класів. Віртуальні методи і їх перевизначення. Ключове слово this. Інтерфейси. Синтаксис визначення інтерфейсів. Порівняння інтерфейсів і абстрактних класів. Реалізація інтерфейсів в класах.

Змістовий модуль 4

Організація мультимедійних програм і даних

Тема 10. Принципи створення візуальних інтерфейсів

10.1. Візуальний інтерфейс.

Додаток. Простори імен. Потоки. Клас Application. Спадкоємець Form для головної форми. Ресурси програми. Файл для логіки модуля. Іменування форми. Аналіз коду, згенерований дизайнером. Властивості форми. Методи форми. Діалогові вікна. Немодальні вікна. Багатодокументний інтерфейс. Компоненти .NET. Загальні компоненти. Шаблони. Колекції.

10.2. Події.

Події та їх виклик. Події на прикладі форми. Використання власних делегатів.

10.3. Застосування Майстра Visual Studio для роботи з елементами класу.

Огляд базових компонентів Visual Studio. Динамічне створення компонентів.

Тема 11. Програмування графіки

11.1. Огляд технології GDI+.

Поверхня малювання Graphics. Класи та методи GDI+: графічні структури, пера Pen, пензлі Brush (одноколірні, лінійні складні градієнтні, текстурні і штрихові), колір Color, матриці кольору і трансформацій. Координатна система GDI+. Колірні моделі та формати графічних файлів. Способи малювання прямих ліній, геометричних примітивів, кривих Безьє, шляхів і областей. Робота з картинками. Графічний дизайнер. Малювання елементів списку ListBox.

11.2. Особливості розробка мультимедійного контенту в Windows Forms графічних додатках.

Графічні трансформації Якубовича. Визначення зіткнення різних фігур. Розроблення нових графічних класів. Створювання вікон і елементів управління будь-якої форми.

Тема 12. Програмування додатків з елементами мультимедіа

12.1. Особливості роботи зі звуком, відео та анімацією.

Приклади розроблення графічних додатків зі звуком, відео та анімацією. Принципи створення мультимедійних ігор (стратегії, головоломки, "бродилки", "стрілялки" тощо).

12.3. Перелік графічних додатків середовища Visual Studio.

Порівняльний огляд можливостей та галузей застосування графічних додатків Windows Form Application, WPW Application, Silverlight Application.

12.4. Перспективи застосування С-образних мов програмування у галузі поліграфії.

Порівняльний огляд сучасних авторських середовищ розроблення мультимедіа.

4. Структура навчальної дисципліни

Із самого початку вивчення навчальної дисципліни кожен студент має бути ознайомлений як із робочою програмою навчальної дисципліни і формами організації навчання, так і зі структурою, змістом та обсягом кожного з її навчальних модулів, а також з усіма видами контролю та методикою оцінювання сформованих професійних компетентностей.

Вивчення студентом навчальної дисципліни відбувається шляхом послідовного і ґрунтовного опрацювання навчальних модулів. Навчальний модуль – це окремих, відносно самостійний блок дисципліни, який логічно об'єднує кілька навчальних елементів дисципліни за змістом та взаємозв'язками. Тематичний план дисципліни складається з двох змістових модулів (табл. 4.1).

Таблиця 4.1

Структура залікового кредиту навчальної дисципліни

Назви змістових модулів і тем	Кількість годин				
	денна форма				
	усього	у тому числі			
		лекційні	лабораторні	самостійна робота	
			проведення підсумково- го контролю	підготовка до занять	
1	2	3	4	5	6
Змістовий модуль 1 Організація процедуро-орієнтованих програм					
<i>Тема 1.</i> Теоретичні та методологічні засади організації програм і даних	12	2	4	—	6
<i>Тема 2.</i> Поняття типу даних	12	2	4	—	6

Продовження табл. 4.1

1	2	3	4	5	6
<i>Тема 3. Програмування обчислювальних процесів</i>	21	2	4	–	15
<i>Тема 4. Масиви</i>	16	2	4	–	10
Разом за змістовим модулем 1	61	8	16		37
Змістовий модуль 2 Організація і оброблення складених типів даних					
<i>Тема 4. Масиви (продовження)</i>	23	2	6	–	15
<i>Тема 5. Структури</i>	12	2	4	–	6
<i>Тема 6. Функції</i>	52	4	8	–	40
<i>Разом за змістовим модулем 2</i>	87	8	18	–	61
Разом за змістовим модулем 1 та змістовим модулем 2	148	16	34	–	98
Змістовий модуль 3 Базові концепції об'єктно-орієнтованого програмування (ООП)					
<i>Тема 7. Особливості об'єктно-орієнтованого стилю програмування</i>	11	2	4	–	5
<i>Тема 8. Методи класів</i>	35	4	6	–	25
<i>Тема 9. Відносини між класами</i>	18	2	6	–	15
Разом за змістовим модулем 3	64	8	16	–	40
Змістовий модуль 4 Організація мультимедійних програм і даних					
<i>Тема 10. Принципи створення візуальних інтерфейсів</i>	16	2	4	–	10
<i>Тема 11. Програмування графіки</i>	32	4	8	–	20
<i>Тема 12. Програмування додатків з елементами мультимедіа</i>	28	4	6	–	18

1	2	3	4	5	6
<i>Підготовка до екзамену та екзамен</i>	12	–	–	12	
Разом за змістовим модулем 4	88	10	18	12	48
Разом за змістовим модулем 3 та змістовим модулем 4	152	18	34	12	82
Усього годин за модулем	300	34	68	12	186
				198	

5. Теми лабораторних занять

Лабораторне заняття – форма навчального заняття, за якої студент під керівництвом викладача особисто проводить експерименти чи досліди з метою практичного підтвердження окремих теоретичних положень навчальної дисципліни. У ході лабораторних робіт студент набуває професійних компетентностей та практичних навичок роботи з комп'ютерним обладнанням, відповідними програмними продуктами.

Напередодні проведення кожного лабораторного заняття (звичайно після відповідної лекції) студентам видається завдання, що містить:

тему і мету заняття;

список питань для підготовки (це можуть бути контрольні питання з теми, що вивчається, заповнення розданих матеріалів індивідуальними даними, розробка програм, таблиць і т. д.);

послідовність підлягаючих виконанню на занятті дій;

вимоги до змісту звіту.

Студент повинен вивчити навчальний матеріал, завдання, підготувати необхідні для роботи на занятті матеріали і знати відповіді на контрольні питання. У ході підготовки може бути створена заготовка звіту, що дозволить заощадити час на занятті.

Усі лабораторні заняття з дисципліни проводяться фронтально, кожний студент працює за окремим комп'ютером.

За результатами виконання завдання на лабораторному занятті студенти оформляють індивідуальні звіти про його виконання та захищають ці звіти перед викладачем (табл. 5.1).

Перелік тем лабораторних занять

Назва теми	Програмні питання	Кількість годин	Література
1	2	3	4
Змістовий модуль 1. Організація процедуро-орієнтований програм			
<i>Тема 1.</i> Теоретичні та методологічні засади організації програм і даних	<i>Лабораторне заняття 1.</i> Інтегроване середовище системи програмування Visual Studio.NET	4	Основна: [1 – 3]. Додаткова: [4, 5]
<i>Тема 2.</i> Поняття типу даних	<i>Лабораторне заняття 2.</i> Програмування лінійних обчислювальних процесів	4	Основна: [1 – 3]. Додаткова: [4,6]
<i>Тема 3.</i> Програмування обчислювальних процесів	<i>Лабораторне заняття 3.</i> Програмування обчислювальних процесів, що розгалужуються	4	Основна: [1 – 3]. Додаткова: [5]
	<i>Лабораторне заняття 4.</i> Програмування циклічних обчислювальних процесів	4	Основна: [1 – 3]. Додаткова: [5]
Змістовий модуль 2 Організація і оброблення складених типів даних			
<i>Тема 4.</i> Масиви	<i>Лабораторне заняття 5.</i> Оброблення одномірних масивів і матриць	6	Основна: [1 – 3]. Додаткова: [6]
<i>Тема 5.</i> Структури	<i>Лабораторне заняття 6.</i> Оброблення структур	4	Основна: [1 – 3]. Додаткова: [5]
<i>Тема 6.</i> Функції	<i>Лабораторне заняття 7.</i> Використання функцій	8	Основна: [1 – 3]. Додаткова: [4]
Разом годин за змістовими модулями 1 і 2		34	

1	2	3	4
Змістовий модуль 3			
Базові концепції об'єктно-орієнтованого програмування (ООП)			
<i>Тема 7. Особливості об'єктно-орієнтованого стилю програмування</i>	<i>Лабораторна робота 8. Використання простих класів</i>	4	Основна: [1 – 3]. Додаткова: [4]
<i>Тема 8. Методи класів</i>	<i>Лабораторна робота 9. Використання простих методів</i>	2	Основна: [1 – 3]. Додаткова: [6]
	<i>Лабораторна робота 10. Визначення особливостей застосування конструкторів</i>	4	Основна: [1 – 3]. Додаткова: [6]
<i>Тема 9. Відносини між класами</i>	<i>Лабораторна робота 11. Розроблення програм з ієрархією класів</i>	6	Основна: [1 – 3]. Додаткова: [6]
Змістовий модуль 4			
Організація мультимедійних програм і даних			
<i>Тема 10. Принципи створення візуальних інтерфейсів</i>	<i>Лабораторна робота 12. Розробка типового каркаса графічного Windows-дodatка</i>	4	Основна: [1 – 3]. Додаткова: [4]
<i>Тема 11. Програмування графіки</i>	<i>Лабораторна робота 13. Розробка MDI- і SDI-дodatків за допомогою компонентів Designer Forms.</i>	4	Основна: [1 – 3]. Додаткова: [4]
	<i>Лабораторна робота 14. Розробка Windows-дodatків з елементами графіки</i>	4	Основна: [1 – 3]. Додаткова: [4; 5]
<i>Тема 12. Програмування додатків з елементами мультимедіа</i>	<i>Лабораторна робота 15. Розробка мультимедійних Windows-дodatків (звук, відео, анімація)</i>	6	Основна: [1 – 3]. Додаткова: [4; 5]
Разом годин за змістовими модулями 3 і 4		34	
Усього годин		68	

5.1. Приклади типових лабораторних завдань за темами

Змістовий модуль 1

Організація процедуро-орієнтованих програм

Тема 3. Програмування обчислювальних процесів

Завдання.

1. Формули для обчислення у вигляді $F = f(a, b, c, x)$ і опис змінних взяти з відповідного варіанта індивідуального завдання (див. додатковий файл з індивідуальними завданнями).

Результат виконання завдання повинен відповідати такому сценарію роботи:

Введіть чотири дійсні числа:

a =? <число_1> <Введення>

b =? <число_1> <Введення>

c =? <число_1> <Введення>

x =? <число_1> <Введення>

Обрана гілка № <виводиться номер гілки 1,2 або 3>

F = <виводиться результат обчислення>

Результат повинен містити три варіанти відповіді (по одному для кожної з гілок обчислювального процесу) і відповідні значення функції $F = f(a, b, c, x)$, обчислені (для контролю) на калькуляторі.

2. Проаналізувати отримані вирази: визначити допустимі діапазони зміни вхідних величин, їх розмірність і тип.

3. Підготувати контрольні приклади (використовуючи, наприклад калькулятор), які повною мірою характеризують аналізовані вирази

4. Розробити алгоритм обчислення і намалювати його графічну схему (блок-схему).

5. Відповідно до алгоритму набрати і відкомпілювати текст програми, усуваючи у разі необхідності помилки.

6. Дослідити роботу програми, аналізуючи виконання контрольних прикладів.

Змістовий модуль 3

Базові концепції об'єктно-орієнтованого стилю програмування

Тема 7. Особливості об'єктно-орієнтованого стилю програмування

Завдання.

1. З таблиці (вона приведена в методичних вказівках до лабораторної роботи) вибрати індивідуальний варіант предметної області. Наприклад, "Будівлі".

2. Розробити відповідно предметної області клас, який інкапсулює елементи-дані й елементи-методи (без параметрів). Написати і налагодити програму, що демонструє роботу з об'єктом (або об'єктами), визначеного вище класу.

3. Визначити додатковий метод (з параметрами) і включити його до раніш розробленого класу. Написати і налагодити програму, що демонструє роботу з об'єктом (або об'єктами), визначеного вище класу.

6. Самостійна робота

Самостійна робота студента – це форма організації навчального процесу, за якої заплановані завдання виконуються студентом самостійно під методичним керівництвом викладача.

Мета самостійної роботи – засвоєння в повному обсязі навчальної програми та формування у студентів загальних і професійних компетентностей, які відіграють суттєву роль у становленні майбутнього фахівця вищого рівня кваліфікації.

Для поглибленого самостійного вивчення пропонуються наступні теми, а також індивідуальне завдання (табл. 6.1).

Таблиця 6.1

Теми для самостійної роботи студентів

Назва теми	Зміст самостійної роботи студентів	Кількість годин	Форми контролю СРС	Література
1	2	3	4	5
Змістовий модуль 1				
Організація процедур орієнтований програм				
<i>Тема 1.</i> Теоретичні та методологічні засади організації програм і даних	Вивчення лекційного матеріалу, підготовка до лабораторного заняття. Використання середовища Visual Studio для створення консольних додатків	6	Презентація результатів	Основна: [1; 2; 4]. Додаткова: [5]
<i>Тема 2.</i> Поняття типу даних	Вивчення лекційного матеріалу, підготовка до лабораторного заняття. Особливості застосування вбудованих типів даних	6	Презентація результатів	Основна: [2; 3]. Додаткова: [6] Інформаційні ресурси: [8]

1	2	3	4	5
<i>Тема 3.</i> Програмування обчислюваль- них процесів	Вивчення лекційного мате- ріалу, підготовка до лабо- раторного заняття. Розроблення консольної програми, яка реалізує об- числення лінійного мате- матичного виразу згідно з індивідуальним варіантом	5	Презентація результатів	Основна: [1]. Додаткова: [4] Інформацій- ні ресурси: [7]
<i>Тема 3.</i> Програмування обчислюваль- них процесів	Вивчення лекційного мате- ріалу, підготовка до лабо- раторного заняття. Розроблення консольної програми, яка реалізує об- числення математичного виразу з розгалуженням згідно з індивідуальним варіантом	5	Презентація результатів	Основна: [2; 3]. Додаткова: [5] Інформацій- ні ресурси: [9]
<i>Тема 3.</i> Програмування обчислюваль- них процесів	Вивчення лекційного мате- ріалу, підготовка до лабо- раторного заняття. Розроблення консольної програми, яка реалізує створення таблиці обчис- лення значень математич- ного виразу згідно з інди- відуальним варіантом	5	Презентація результатів	Основна: [3]. Додаткова: [6,7] Інформацій- ні ресурси: [4]
Усього за змістовим модулем 1		27		
Змістовий модуль 2				
Організація і оброблення складених типів даних				
<i>Тема 4. Масиви</i>	Вивчення лекційного мате- ріалу, підготовка до лабо- раторного заняття. Розроблення консольної програми, яка реалізує згідно з індивідуальним варіантом оброблення масиву	25	Презентація результатів	Основна: [2; 3]. Додаткова: [4] Інформаційні ресурси: [7]:

1	2	3	4	5
<i>Тема 5.</i> Структури	Вивчення лекційного матеріалу, підготовка до лабораторного заняття. Розроблення консольної програми, яка реалізує згідно з індивідуальним варіантом оброблення структури	6	Презентація результатів	Основна: [2; 3]. Додаткова: [6] Інформаційні ресурси: [9]
<i>Тема 6.</i> Функції	Вивчення лекційного матеріалу, підготовка до лабораторного заняття. Розроблення консольної програми, яка реалізує згідно з індивідуальним варіантом оброблення масиву з застосуванням певних функцій	40	Презентація результатів	Основна: [1; 3]. Додаткова: [5] Інформаційні ресурси: [8]
Усього за змістовим модулем 2		71		
Усього за змістовим модулем 1 і 2		98		
Змістовий модуль 3 Базові концепції об'єктно-орієнтованого програмування (ООП)				
<i>Тема 7.</i> Особливості об'єктно-орієнтованого стилю програмування	Розроблення консольної програми, яка створює певний клас з властивостями та здійснює подальше застосування екземплярів даного класу	5	Презентація результатів	Основна: [2; 3]. Додаткова: [4] Інформаційні ресурси: [9]
<i>Тема 8.</i> Методи класів	Підготовка до лабораторного заняття. Розроблення консольної програми, яка створює певний клас з властивостями і методами та здійснює подальше застосування екземплярів даного класу	25	Презентація результатів	Основна: [1]. Додаткова: [5] Інформаційні ресурси: [7]

1	2	3	4	5
<i>Тема 9.</i> Відносини між класами	Вивчення лекційного матеріалу, підготовка до лабораторного заняття. Розроблення консольної програми, яка створює згідно з індивідуальним варіантом об'єктно-орієнтовану програму з ієрархією класів та здійснює подальше застосування похідних екземплярів даного класу	10	Презентація результатів	Основна: [1]. Додаткова: [6] Інформаційні ресурси: [7]
Усього за змістовим модулем 3		40		
Змістовий модуль 4 Організація мультимедійних програм і даних				
<i>Тема 10.</i> Принципи створення візуальних інтерфейсів	Вивчення лекційного матеріалу, підготовка до лабораторного заняття. Розроблення програми, яка створює згідно з індивідуальним варіантом типовий каркас графічного Windows-додатка	10	Презентація результатів	Основна: [1]. Додаткова: [5] Інформаційні ресурси: [7]
<i>Тема 11.</i> Програмування графіки	Вивчення лекційного матеріалу, підготовка до лабораторного заняття. Розроблення програми, яка створює згідно з індивідуальним варіантом MDI- і SDI-додатки за допомогою компонентів Designer Forms	10	Презентація результатів	Основна: [1]. Додаткова: [5] Інформаційні ресурси: [7]
<i>Тема 11.</i> Програмування графіки	Вивчення лекційного матеріалу, підготовка до лабораторного заняття. Розроблення програми, яка створює згідно з індивідуальним варіантом Windows-додаток з елементами графіки	10	Презентація результатів	Основна: [1; 2]. Додаткова: [4] Інформаційні ресурси: [7]

1	2	3	4	5
Тема 12. Програмування додатків з елементами мультимедіа	Вивчення лекційного матеріалу, підготовка до лабораторного заняття. Розроблення програми, яка створює згідно з індивідуальним варіантом мультимедійний Windows-додаток з елементами звука, відео та анімації	18	Презентація результатів	Основна: [1; 2]. Додаткова: [4] Інформаційні ресурси: [7]
<i>Усього за змістовим модулем 4</i>		48		
<i>Усього за змістовим модулем 3 і 4</i>		88		
<i>Підготовка до екзамену і екзамен</i>		12		
<i>Усього за змістовими модулями 1 – 4</i>		198		

6.1. Контрольні запитання для самодіагностики

Тема 1. Теоретичні та методологічні засади організації програм і даних

1. Укажіть місце дисципліни "Програмування засобів мультимедіа" в навчальному процесі напряму підготовки 6.051501 "Видавничо-поліграфічна справа".
2. У чому сутність принципу програмного управління комп'ютером?
3. Поясніть такі стилі програмування: процедурний, логічний, функціональний, об'єктно-орієнтований. Наведіть приклади.
4. Вкажіть місце С-образних мов програмування у галузі розроблення сучасних мультимедійних продуктів.
5. Дайте огляд основних етапів розроблення програмного забезпечення.
6. Що розуміється під специфікацією програми?
7. Наведіть приклади визначення вимог до програми.
8. Дайте перелік етапів проектування програми.
9. Що таке технологія .NET?
10. Які особливості інфраструктури .NET-платформи?
11. Що розуміється під традиційним процесом компіляції?
12. Як здійснюється компіляція у .NET початкового коду C#?
13. Дайте перелік лексичних елементів мови C#.

Тема 2. Поняття типу даних

1. Наведіть класифікацію типів даних.
2. Дайте огляд основних типів C#.
3. Наведіть приклади простих типів даних.
4. Наведіть приклади переліків.
5. Що таке типи – структури?
6. Коли доцільно застосовувати типи – класи?
7. Коли доцільно застосовувати тип string?
8. Коли доцільно застосовувати типи – масиви?
9. Коли доцільно застосовувати типи – інтерфейси?
10. Коли доцільно застосовувати типи – делегати?
11. Що розуміється під єдиною системою типів .NET (Common Type System – CTS)?
12. Наведіть приклади синтаксису оголошення змінних.
13. Коли доцільно застосовувати перетворення вбудованих типів?
14. У чому особливість операцій над числами з плаваючою крапкою?
15. 15. Наведіть класифікацію константних величин.
16. Дайте приклади числових, символічних та іменованих констант.

Тема 3. Програмування обчислювальних процесів

1. Що розуміється під алгоритмічними структурами?
2. Дайте огляд основних операцій C#.
3. Наведіть приклади арифметичних і логічних операцій.
4. Наведіть приклади операцій відносин.
5. Що розуміється під синтаксичними блоками?
6. Що розуміється під пріоритетом операцій? Наведіть приклади.
7. Яке призначення простора імен?
8. Що розуміється під областю видимості змінних?
9. Яка різниця між областю видимості і часом існування змінних?
10. Наведіть приклади форматування числових значень.
11. Наведіть приклади перетворення типів даних.
12. Яка особливість структури лінійної програми?
13. У чому суть поняття потоку управління програмою?
14. Що розуміється під структурою вибору альтернатив?
15. Структура вибору if. Структура вибору if / else. Множинний вибір – структура switch. Умовне вираження. Наведіть приклади.
16. Що розуміється під структурою повторення?

17. Наведіть приклади синтаксису операторів з передумовою (while).

18. Наведіть приклади синтаксису операторів з післяумовою (do while).

19. Наведіть приклади синтаксису операторів циклу for і foreach.

20. Наведіть приклади синтаксису управляючих операторів у циклах.

21. Дайте перелік рекомендацій з вибору циклів.

Тема 4. Масиви

1. Коли доцільно застосовувати масиви?

2. Як забезпечується призначення, оголошення і визначення масиву?

3. Яким чином здійснюється доступ до окремих елементів масиву?

4. Наведіть можливі варіанти ініціалізація масиву.

5. Як здійснюється перебір елементів масиву за допомогою оператора foreach.

6. Що таке ранг масиву?

7. Наведіть приклади алгоритмів пошуку та перетворення масивів.

8. У чому суть алгоритму сортування елементів масиву.

9. Як здійснюється доступ до елементів двовимірного масиву.

10. Наведіть приклади оброблення матриць.

11. Як здійснюється операції над строками?

Тема 5. Структури

1. Коли доцільно застосовувати структури?

2. Як здійснюється призначення, оголошення і визначення структур?

3. Коли доцільно застосовувати масиви структур?

4. У чому особливості оброблення елементів структур?

5. Наведіть приклади оброблення масиву структур.

Тема 6. Функції

1. Коли доцільно застосовувати функції?

2. Як здійснюється опис функцій? Що таке сигнатура функції?

3. Що таке формальні параметри функції? Чим вони відрізняються від фактичних?

4. Що розуміється під відповідністю параметрів функції?

5. Коли доцільно застосовувати масиви параметрів?

6. Як здійснюється передача параметрів по посиланням і по значенням? У чому різниця?

7. Опишіть варіанти вихідних параметрів функцій.

8. Дайте опис параметрів і значень, що повертаються за порівнянням з глобальними даними.

9. Що таке перевантаження функцій?

10. У чому суть застосування делегатів?

11. Що розуміється під винятковими ситуаціями?

12. Як здійснюється генерування виняткових ситуацій?

13. Наведіть приклади оброблення виняткових ситуацій.

14. Як здійснюється оброблення файлів?

15. У чому особливість використання бінарних і текстових файлів?

Тема 7. Особливості об'єктно-орієнтованого стилю програмування

1. У чому суть парадигми ООП?

2. Дайте визначення об'єкта.

3. Наведіть основні принципи ООП.

4. Наведіть приклади мультимедійних сценаріїв, де застосовуються абстрагування, інкапсуляція і спадкування.

5. У чому сутність уніфікованої мови моделювання (UML) для опису класів і об'єктів?

6. Як здійснюється визначення класу?

7. Наведіть приклад синтаксису класу.

8. Що таке поля класу?

9. Як здійснюється доступ до полів класу?

10. Дайте порівняння властивостей і полів класу.

11. Як здійснюється доступ до стану об'єкта?

12. Що розуміється під життєвим циклом об'єкта?

Тема 8. Методи класів

1. Як здійснюється доступ до методів?

2. Що таке методи-властивості?

3. Коли застосовуються індексатори?

4. Що таке статичні поля, і який їх зв'язок із методами класу?

5. Призначення конструкторів і деструкторів.

6. Наведіть приклади конструкторів із параметрами.

7. Чим відрізняються конструктори від методів?

8. Як здійснюється обмін інформацією між методами?

9. Коли доцільно застосовувати рекурсивний виклик методів?

Тема 9. Відносини між класами

1. Назвіть можливі типи відносин між класами.

2. У чому суть відношення “клієнт – постачальник”

3. У чому суть відношення “батько – спадкоємець”?
4. Наведіть приклад простої ієрархії класів.
5. Як здійснюється доступ до елементів класу у разі наявності спадкоємства?
6. У чому суть використання захищеного доступу елементів класу?
7. Назвіть особливості застосування конструктори у разі наявності спадкоємства.
8. Дайте визначення абстрактних класів.
9. Що таке віртуальні методи і коли їх доцільно застосовувати?
10. Як здійснюється перевизначення віртуальних методів?
11. Що означає ключове слово this?
12. Яке призначення інтерфейсів?
13. Наведіть синтаксис визначення інтерфейсів.

Тема 10. Принципи створення візуальних інтерфейсів

1. Назвіть основні етапи розроблення графічного додатку.
2. Дайте призначення класу Application.
3. Опишіть спадкоємець Form для головної форми.
4. Що розуміється під ресурсами програми? Наведіть приклади.
5. Наведіть перелік основних властивостей форми.
6. Наведіть перелік основних методів форми.
7. Дайте визначення модального вікна.
8. Яка особливість немодального вікна.
9. Наведіть приклад визначення діалогового вікна.
10. Що таке колекція? Дайте опис її компонентів.
11. У чому суть подій і як забезпечується їх виклик?
12. Який зв'язок між подіями і власними делегатами?
13. Дайте перелік можливостей застосування Майстра Visual Studio для роботи з елементами класу.

Тема 11. Програмування графіки

1. Що розуміється під технологією GDI+?
2. Як створюється поверхня малювання Graphics?
3. Які графічні структури застосовуються в GDI+?
4. Наведіть приклад синтаксису застосування пера Pen.
5. Наведіть приклад синтаксису застосування пензля Brush.
6. Що таке матриця кольору? Де і коли вона застосовується?

7. У чому суть трансформацій графічних об'єктів?
8. Дайте опис координатної системи GDI+.
9. Наведіть перелік колірних моделей та формати графічних файлів.
10. Опишіть способи малювання прямих ліній.
11. Опишіть способи малювання геометричних примітивів.
12. Що таке крива Безьє? Як вона формується і коли її доцільно застосовувати?
13. Опишіть алгоритм роботи з картинками.
14. Як здійснюється малювання елементів списку ListBox?

Тема 12. Програмування додатків з елементами мультимедіа

1. Дайте перелік особливості розроблення мультимедійного контенту в Windows Forms графічних додатках.
2. Опишіть алгоритм додавання звука в графічний додаток.
3. Наведіть синтаксис виклику відеофайла в графічний додаток.
4. Як додати анімаційні ефекти в графічний додаток?
5. Дайте порівняльний огляд можливостей та галузей застосування графічних додатків Windows Form Application, WPW Application, SilverlightApplication.
6. Які перспективи застосування С-образних мов програмування в галузі поліграфії?
7. Дайте порівняльний огляд сучасних авторських середовищ розроблення мультимедіа.

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.

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 та групові;

в) для комплексного оцінювання засвоєння програмного матеріалу: індивідуальне здавання виконаних робіт.

8. Методи навчання

Методика викладення дисципліни передбачає використання сучасних технічних засобів, включаючи комп'ютери, мультимедійні проектори та комунікаційні пристрої. Для індивідуалізації навчання студентам видаються диференційовані індивідуальні завдання, завдання на лабораторні заняття, а також завдання для самостійної роботи.

У процесі викладання навчальної дисципліни для активізації навчально-пізнавальної діяльності студентів зазвичай передбачене застосування як активних, так і інтерактивних навчальних технологій, серед яких: лекції проблемного характеру, міні-лекції, робота в малих групах, семінари-дискусії, мозкові атаки, кейс-метод, презентації, ознайомлювальні (початкові) ігри, метод проектної роботи, комп'ютерні симуляції, метод Дельфі, метод сценаріїв, банки візуального супроводу (табл. 8.1).

Таблиця 8.1

Розподіл форм та методів активізації процесу навчання за темами навчальної дисципліни

Тема	Практичне застосування навчальних технологій
<i>Тема 1.</i> Теоретичні та методологічні засади організації програм і даних	Робота в малих групах, презентація результатів.
<i>Тема 2.</i> Поняття типу даних	Лекція проблемного характеру з питання "Синтаксичні блоки", робота в малих групах
<i>Тема 3.</i> Програмування обчислювальних процесів	Робота в малих групах, презентація результатів
<i>Тема 4.</i> Масиви	Робота в малих групах, презентація результатів
<i>Тема 5.</i> Структури	Робота в малих групах, презентація результатів.
<i>Тема 6.</i> Функції	Лекція проблемного характеру з питання "Обмін інформацією між функціями", робота в малих групах

1	2
<i>Тема 7. Особливості об'єктно-орієнтованого стилю програмування</i>	Лекція проблемного характеру з питання "Властивості об'єктної моделі програмування", робота в малих групах
<i>Тема 8. Методи класів</i>	Робота в малих групах, презентація результатів
<i>Тема 9. Відносини між класами</i>	Робота в малих групах, презентація результатів
<i>Тема 10. Принципи створення візуальних інтерфейсів</i>	Робота в малих групах, презентація результатів
<i>Тема 11. Програмування графіки</i>	Лекція проблемного характеру з питання "Особливості програмування графіки", робота в малих групах
<i>Тема 12. Програмування додатків з елементами мультимедіа</i>	Робота в малих групах, презентація результатів

Лекції проблемного характеру – один із найважливіших елементів проблемного навчання студентів. Вони передбачають поряд із розглядом основного лекційного матеріалу встановлення та розгляд кола проблемних питань дискусійного характеру, які недостатньо розроблені в науці й мають актуальне значення для теорії та практики. Лекції проблемного характеру відрізняються поглибленою аргументацією матеріалу, що викладається. Вони сприяють формуванню у студентів самостійного творчого мислення, прищеплюють їм пізнавальні навички. Студенти стають учасниками наукового пошуку та вирішення проблемних ситуацій.

Презентації – виступи перед аудиторією, що використовуються для представлення певних досягнень, результатів роботи групи, звіту про виконання індивідуальних завдань, проектних робіт. Презентації можуть бути як індивідуальними, наприклад виступ одного слухача, так і колективними, тобто виступи двох та більше слухачів.

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.

Використання методик активізації процесу навчання

Тема навчальної дисципліни	Практичне застосування методик	Методики активізації процесу навчання
<i>Тема 2. Поняття типу даних</i>	<i>Завдання 1. Обґрунтувати перелік даних відповідно певного сценарію</i>	Робота в малих групах, мозкові атаки
<i>Тема 3. Програмування обчислювальних процесів</i>	<i>Завдання 2. Написати програму, яка реалізує певний сценарій</i>	Робота в малих групах, мозкові атаки
<i>Тема 6. Функції</i>	<i>Завдання 2. Розробити опис функції, яка реалізує певну дію</i>	Робота в малих групах, мозкові атаки
<i>Тема 7. Особливості об'єктно-орієнтованого стилю програмування</i>	<i>Завдання 3. Розробити ОО-програму, згідно з певним сценарієм</i>	Робота в малих групах, мозкові атаки
<i>Тема 12. Програмування додатків з елементами мультимедіа</i>	<i>Завдання 1. Розробити додаток з елементами мультимедіа</i>	Робота в малих групах, мозкові атаки, презентації

9. Методи контролю

Система оцінювання сформованих компетентностей (див. табл. 2.1) у студентів ураховує види занять, які згідно з програмою навчальної дисципліни передбачають лекційні, лабораторні заняття, а також виконання самостійної роботи. Оцінювання сформованих компетентностей у студентів здійснюється за накопичувальною 100-бальною системою. Відповідно до Тимчасового положення "Про порядок оцінювання результатів навчання студентів за накопичувальною бально-рейтинговою системою" ХНЕУ ім. С. Кузнеця, контрольні заходи включають наступні види.

Поточний контроль, що здійснюється протягом семестру під час проведення лекційних та лабораторних занять і оцінюється сумою набраних балів. Для першого семестру максимальна сума балів складає 100, для отримання заліку достатнє отримати 60 балів. Для другого семестру максимальна сума балів – 60, а мінімальна сума, що дозволяє студенту скласти іспит, – 35 балів.

Модульний контроль, що проводиться з урахуванням поточного контролю за відповідний змістовий модуль і має на меті *інтегровану* оцінку результатів навчання студента після вивчення матеріалу з логічно завершеної частини дисципліни – змістового модуля.

Підсумковий/семестровий контроль, що проводиться у формі заліку, або семестрового екзамену, відповідно до графіка навчального процесу.

Порядок проведення поточного оцінювання знань студентів.

Поточний контроль з даної навчальної дисципліни проводиться у таких формах:

- активна робота на лекційних заняттях;
- активна робота під час лабораторних занять;
- захист індивідуального завдання для поточних лабораторних робіт;
- проведення поточного тестування;
- проведення письмової контрольної роботи;
- експрес-опитування.

Оцінювання знань студента під час лекційних, лабораторних занять та виконання індивідуальних лабораторних завдань проводиться за накопичувальною 100-бальною системою за такими критеріями:

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

ступінь засвоєння фактичного матеріалу навчальної дисципліни;

ознайомлення з рекомендованою літературою, а також із сучасною літературою з питань, що розглядаються;

вміння поєднувати теорію з практикою під час розгляду технологічних засобів, розв'язання задач, проведення розрахунків у процесі виконання індивідуальних завдань та завдань, винесених на розгляд в аудиторії.

Максимально можливий бал за конкретним завданням ставиться за умови відповідності виконання завдання або усної відповіді всім зазначеним критеріям. Відсутність тієї або іншої складової знижує кількість балів. Під час оцінюванні індивідуальних завдань увага також приділяється якості, самостійності та своєчасності здачі виконаних завдань викладачу, згідно з графіком навчального процесу. Якщо якась із вимог не буде виконана, то бали будуть знижені.

Поточний контроль роботи студентів на лабораторних заняттях здійснюється у формі індивідуального опитування за звітами з лабо-

раторних робіт, що передбачає ґрунтовні, розгорнуті відповіді студентів на питання, що належить до матеріалу лабораторної роботи. Питання індивідуального опитування стимулюють студентів логічно мислити, порівнювати, аналізувати, доводити, підбирати переконливі приклади, встановлювати причинно-наслідкові зв'язки, робити обґрунтовані висновки.

Критерії оцінювання позааудиторної самостійної роботи студентів. Загальними критеріями, за якими здійснюється оцінювання позааудиторної самостійної роботи студентів, є: глибина і міцність знань, рівень мислення, вміння систематизувати знання за окремими темами, вміння робити обґрунтовані висновки, володіння категорійним апаратом, навички і прийоми виконання практичних завдань, вміння знаходити необхідну інформацію, здійснювати її систематизацію та оброблення, використовувати її на лабораторних заняттях та під час виконання індивідуального завдання.

Поточний контроль роботи студентів у рамках самостійної роботи здійснюється за практичне виконання індивідуального завдання.

Підсумковий контроль знань та компетентностей студентів з навчальної дисципліни здійснюється на підставі проведення заліку у першому семестрі, та семестрового екзамену у другому.

Порядок підсумкового контролю з навчальної дисципліни у першому семестрі. Підсумковий контроль знань та компетентностей студентів здійснюється за результатами поточного контролю. Підсумкова оцінка розраховується з урахуванням балів, отриманих під час поточного контролю за накопичувальною системою:

активна робота на лекції (0,2 бала за 2 години занять) – максимум 1,6 бала; виконання та здача (захист) лабораторних робіт – максимум 94,4 балів; виконання завдань контрольних робіт – максимум 4 бали.

Сумарний результат у балах за семестр складає: *"60 і більше балів – зараховано"*, *"59 і менше балів – не зараховано"* та заноситься у залікову *"Відомість обліку успішності"* навчальної дисципліни.

Зразок завдань для контрольних робіт.

Змістовий модуль 1. Організація процедурно орієнтованих програм
Рівень 1 (діагностичне завдання).

1. Наведіть схему структури типової C# програми.
2. Наведіть приклади оголошення і визначення змінної. У чому різниця?

3. Дайте визначення константи. Які типи констант ви знаєте? Як визначити константу t зі значенням 512?
4. Що надрукує наступна програма?

```
using System;
class Program
{
    static void Main(string[] args)
    {
        int x = 2, y, z;
        x *= 3 + 2;
        Console.WriteLine("x = {0}\n", x);
        x *= y = z = 4;
        Console.WriteLine("x = {0}\n", x);
        x = y / ++z;
        Console.WriteLine("x = {0}\n", x);
        x = (2 * y % --z);
        Console.WriteLine("x = {0}\n", ++x);
        Console.WriteLine("z = {0}\n", z--);
        Console.WriteLine("y = {0}\n", y = ++x);
    }
}
```

Рівень 2 (евристичне завдання).

Задано одновимірний масив з десяти цілих чисел (ім'я – mas).

Необхідно:

намалювати графічну схему алгоритму пошуку максимального елемента масиву кратного трьом (maxKrat3) і значення його індексу (indMaxKrat3);

словесно описати алгоритм, використовуючи ключові фрази типу: "... початковий стан змінних перед початком циклу ...", "... на кожній ітерації проходження масиву виконуються наступні дії ...", "... як результат отримуємо ...";

дати чисельний приклад десяти значень вихідного масиву і результату його обробки згідно з наведеним алгоритмом.

Змістовий модуль 2. Організація і оброблення складених типів даних

Рівень 1 (діагностичне завдання).

Задано одновимірний масив з десяти цілих чисел (ім'я – mas). Необхідно написати і реалізувати на ПК C# програму, яка реалізує алгоритм пошуку суми чисел (sum) розташованих між максимальним (max) і мінімальним (min) елементами масиву.

Рівень 2 (евристичне завдання).

Задано матриця з цілих чисел, розмір і елементи якої вводяться з клавіатури ПК. Необхідно обчислити ранг матриці і номер строки, в якій сума елементів має найбільше значення.

Порядок підсумкового контролю з навчальної дисципліни у другому семестрі. У другому семестрі з дисципліни передбачений екзамен. Екзаменаційний білет охоплює програму дисципліни і передбачає визначення рівня знань та ступеня опанування студентами компетентностей (див. табл. 2.1).

Завданням екзамену є перевірка розуміння студентом програмного матеріалу в цілому, логіки та взаємозв'язків між окремими розділами, здатності творчого використання накопичених знань, вміння формулювати своє ставлення до певної проблеми навчальної дисципліни тощо. Екзамен оцінює рівень засвоєння студентом компетентностей, що передбачені кваліфікаційними вимогами. Кожен екзаменаційний білет складається із 3 практичних завдань, які передбачають вирішення типових професійних задач фахівця та дозволяють діагностувати рівень підготовки студента і рівень його компетентності з матеріалу навчальної дисципліни.

Екзаменаційний білет включає одне діагностичне та два евристичних завдання, які оцінюються відповідно до Тимчасового положення "Про порядок оцінювання результатів навчання студентів за накопичувальною бально-рейтинговою системою" ХНЕУ ім. С. Кузнеця.

Студент, який із поважних причин, підтверджених документально, не мав можливості брати участь у формах поточного контролю, тобто не склав змістовий модуль, має право на його відпрацювання у двотижневий термін після повернення до навчання за розпорядженням декана факультету відповідно до встановленого терміну.

Студент **не може бути допущений** до складання екзамену, якщо кількість балів, одержаних за результатами перевірки успішності під час поточного та модульного контролю відповідно до змістового модуля впродовж семестру, в сумі не досягла 35 балів. Після екзаменаційної сесії декан факультету видає розпорядження про ліквідацію академічної заборгованості. У встановлений термін студент добирає залікові бали.

Студента слід **уважати атестованим**, якщо сума балів, одержаних за результатами підсумкової/семестрової перевірки успішності, дорівнює або перевищує 60. Мінімально можлива кількість балів за поточний і модульний контроль упродовж семестру – 35 та мінімально можлива кількість балів, набраних на екзамені, – 25.

Результат семестрового екзамену оцінюється в балах (максимальна кількість – 40 балів, мінімальна кількість, що зараховується, – 25 балів).

Підсумкова оцінка з навчальної дисципліни розраховується з урахуванням балів, отриманих під час екзамену, та балів, отриманих під час поточного контролю за накопичувальною системою:

активна робота на лекції (0,2 бала за 2 години занять) – максимум 1,8 бала;

активна робота на лабораторному занятті – максимум 3,4 бали;

виконання та здача (захист) лабораторних робіт – максимум 52,8 балів;

виконання завдань контрольних робіт – максимум 2 бали.

Сумарний результат у балах за семестр складає: *"60 і більше балів – зараховано"*, *"59 і менше балів – не зараховано"*. У випадку отримання менше 60 балів студент обов'язково здає залік після закінчення екзаменаційної сесії у встановлений деканом факультету термін. У випадку повторного отримання менше 60 балів декан факультету призначає комісію у складі трьох викладачів на чолі із завідувачем кафедри та визначає термін перескладання заліку, після чого приймається рішення відповідно до чинного законодавства: *"зараховано"* – студент продовжує навчання за графіком навчального процесу, а якщо *"не зараховано"*, тоді декан факультету пропонує студенту повторне вивчення навчальної дисципліни протягом наступного навчального періоду самостійно.

У додатку А наведено "Технологічну карту накопичувальних рейтингових балів з навчальної дисципліни "Програмування засобів мультимедіа" та "Рейтинг-план дисципліни "Програмування засобів мультимедіа".

Зразок екзаменаційного білета

Форма № Н-5.05

Харківський національний економічний університет імені Семена Кузнеця
Освітній ступінь "бакалавр"

Напрямок підготовки: "Видавничо-поліграфічна справа". Семестр V
Навчальна дисципліна "Програмування засобів мультимедіа "

ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ 1

Завдання 1 (діагностичне, 10 балів, виконати на папері).

Задано одновимірний масив з десяти цілих чисел (ім'я – mas). Необхідно:

намалювати графічну схему алгоритму пошуку максимального (max) елемента масиву розташованого після першого негативного елемента (indFirstNegativo);

словесно описати алгоритм, використовуючи ключові фрази типу: "... початковий стан змінних перед початком циклу ...", "... на кожній ітерації проходження масиву виконуються наступні дії ...", "... як результат отримуємо ...";

дати чисельний приклад десяти значень вихідного масиву і результату його оброблення згідно наведеним алгоритмом.

Завдання 2 (евристичне, 15 балів, виконати за комп'ютером).

Задано матрицю з цілих чисел, розмір і елементи якої вводяться з клавіатури ПК. Обчислити номер строки матриці, в якій сума елементів має найменш значення.

Завдання 3 (евристичне, 15 балів, виконати за комп'ютером).

Розробити графічний, який містить дві сторінки, на першій з якої необхідно розмістити у фігурній рамці свої студентські реквізити: П.І.Б., група, e-mail, а також фотографію. Друга сторінка повинна містити поверхню малювання, на якій потрібно за допомогою кривих Без'є створити кольоровий силует людини.

Затверджено на засіданні

кафедри комп'ютерних систем і технологій ХНЕУ ім. С. Кузнеця.

Протокол № ___ від "___" _____ 20___р.

Зав. кафедри _____ Екзаменатор _____
(підпис) (підпис)

Підсумкові бали за екзамен складаються із суми балів за виконання всіх завдань. Окремі завдання оцінюються відокремлено один від одного таким чином.

Максимальна оцінка за повне і правильне виконання кожного з завдань така: перше – 10 балів, друге та третє – по 15 балів; невиконання завдання – 0.

Діагностичне завдання вважається невиконаним, якщо відповідь не дає уявлення про суть алгоритму, який пропонує студент. Оцінка зменшується на 2 бали в таких випадках:

помилки в графічній схемі виконання алгоритму;

неповний словесний опис дії алгоритму, який пропонується студентом;

помилки в чисельному прикладі виконання алгоритму.

Евристичне завдання вважається невиконаним, якщо не створено працездатний Windows додаток. Оцінка зменшується в таких випадках:

створений додаток не задовольняє висунуті вимоги (на 5 балів);

неповне виконання завдання, за якого створений продукт реалізує лише частку функцій (на 3 бали);

помилка, що не дозволяє отримати правильні результати (на 2 бали).

10. Розподіл балів, які отримують студенти

Система оцінювання рівня сформованості професійних компетентностей студентів денної форми навчання наведена в табл. 10.1.

Таблиця 10.1

Система оцінювання рівня сформованості професійних компетентностей

Професійні компетентності	Навчальний тиждень	Години	Форми навчання	Рівень сформованості компетентностей	
				Форми контролю	Макс. Бал
1	2	3	4	5	6
1. Організація процедуро-орієнтованих програм					33,4
ПТ*, ЗТ*, ФТ*** Обґрунтовано визначати оптимальні типи даних для реалізації мультимедійних сценаріїв.	1	Ауд. 2	Лекція Тема 1. Теоретичні та методологічні засади організації програм і даних	Робота на лекції	0,2

Продовження табл. 10.1

1	2	3	4		5	6	
		2	Лабораторне заняття	<i>Лабораторне заняття</i> 1. Інтегроване середовище системи програмування Visual Studio.NET	Робота на занятті. Захист не передбачений	0,2	
		СРС	2	Підготовка до занять	Вивчення лекційного матеріалу, підготовка до лабораторного заняття. Самостійне вивчення питань "Використання середовища Visual Studio для створення консольних додатків"	Використання на лабораторних роботах	–
		Ауд.	2	Лабораторне заняття	<i>Лабораторне заняття</i> 1. Інтегроване середовище системи програмування Visual Studio.NET	Робота на занятті. Захист не передбачений	0,2
	2	СРС	4	Підготовка до занять	Вивчення лекційного матеріалу, підготовка до лабораторного заняття. Самостійне вивчення питань "Використання середовища Visual Studio для створення консольних додатків"	Використання на лабораторних роботах	–
		Ауд.	2	Лекція	<i>Тема 2. Поняття типу даних</i>	Робота на лекції	0,2
		Ауд.	2	Лабораторне заняття	<i>Лабораторне заняття</i> 2. Програмування лінійних обчислювальних процесів	Робота на занятті. Захист лабораторної роботи 1	7,2
	3	СРС	3	Підготовка до занять	Вивчення лекційного матеріалу, підготовка до лабораторного заняття. Самостійне вивчення питань: "Особливості застосування вбудованих типів даних"	Використання на лабораторних роботах	–

1	2	3	4		5	6	
<p style="text-align: center;">ПТ*, ЗТ**, ФТ***</p> <p>Здійснювати обґрунтований вибір середовища програмної розробки мультимедійних продуктів. Розробляти програми з застосуванням структур вибору альтернатив</p>	4	Ауд.	2	Лабораторне заняття	<i>Лабораторне заняття 2.</i> Програмування лінійних обчислювальних процесів	Робота на занятті. Захист не передбачений	0,2
		СРС	3	Підготовка до занять	Вивчення лекційного матеріалу, підготовка до лабораторного заняття. Самостійне вивчення питань: "Особливості застосування вбудованих типів даних"	Використання на лабораторних роботах	—
		Ауд.	2	Лекція	<i>Тема 3.</i> Програмування обчислювальних процесів	Робота на лекції	0,2
			2	Лабораторне заняття	<i>Лабораторне заняття 3.</i> Програмування обчислювальних процесів що розгалужуються	Робота на занятті. Захист лабораторної роботи 2	10,2
	5	СРС	7	Підготовка до занять	Вивчення лекційного матеріалу, підготовка до лабораторного заняття. Самостійне вивчення питань: "Розробка консольної програми, яка реалізує обчислення математичного виразу з розгалуженням"	Використання на лабораторних роботах	—
		Ауд	2	Лабораторне заняття	<i>Лабораторне заняття 3.</i> Програмування обчислювальних процесів що розгалужуються	Робота на занятті. Захист не передбачений	0,2
			8	Підготовка до занять	Вивчення лекційного матеріалу, підготовка до лабораторного заняття. Самостійне вивчення питань: "Розроблення консольної програми, яка реалізує обчислення математичного виразу з розгалуженням"	Використання на лабораторних роботах	—
		6	СРС	8	Підготовка до занять	Вивчення лекційного матеріалу, підготовка до лабораторного заняття. Самостійне вивчення питань: "Розроблення консольної програми, яка реалізує обчислення математичного виразу з розгалуженням"	Використання на лабораторних роботах

1	2	3	4		5	6		
<p>Розробляти програми з застосуванням структур вибору альтернатив. Розробляти програми, які здійснюють оброблення масивів.</p>	7	Ауд.	2	Лекція	Тема 4. Масиви	Робота на лекції Перевірка контрольної роботи	2,2	
			2	Лабораторне заняття	Лабораторне заняття 4. Програмування циклічних обчислювальних процесів	Робота на занятті. Захист лабораторної роботи 3	12,2	
		СРС	5	Підготовка до занять	Вивчення лекційного матеріалу, підготовка до лабораторного заняття. Самостійне вивчення питань: "Обробка масивів" Підготовка до контрольної роботи	Використання на лабораторних роботах	—	
			8	Ауд.	2	Лабораторне заняття	Лабораторне заняття 4. Програмування циклічних обчислювальних процесів	Робота на занятті. Захист не передбачений
	СРС	5			Підготовка до занять	Вивчення лекційного матеріалу, підготовка до лабораторного заняття. Самостійне вивчення питань: "Оброблення масивів"	Використання на лабораторних роботах	—
		2. Організація і оброблення складених типів даних						66,6
	<p>Розробляти програми, які здійснюють обробку масивів. Реалізувати типові алгоритми пошуку, перетворення та сортування елементів масиву. Розробляти програми, які здійснюють опрацювання структур</p>	9		Ауд.	2	Лекція	Тема 4. Масиви	Робота на лекції
			2		Лабораторне заняття	Лабораторне заняття 5. Оброблення одновимірних масивів і матриць	Робота на занятті. Захист лабораторної роботи 4	10,2
СРС			7	Підготовка до занять	Вивчення лекційного матеріалу, підготовка до лабораторного заняття. Самостійне вивчення питань: "Оброблення масивів"	Використання на лабораторних роботах	—	

Продовження табл. 10.1

1	2	3	4		5	6	
	10	Ауд.	2	Лабо- раторне заняття	<i>Лабораторне занят- тя 5. Оброблення од- новимірних масивів і матриць</i>	Робота на занятті. Захист не перед- бачений	0,2
		СРС	8	Підго- товка до занять	Вивчення лекційного матеріалу, підготовка до лабораторного за- няття. Самостійне вивчення питань: "Оброблення масивів"	Використан- ня на лабо- раторних роботах	—
		Ауд.	2	Лекція	<i>Тема 5. Структури</i>	Робота на лекції	0,2
			2	Лабо- раторне заняття	<i>Лабораторне занят- тя 5. Оброблення од- новимірних масивів і матриць</i>	Робота на занятті. Захист не перед- бачений	0,2
		СРС	3	Підго- товка до занять	Вивчення лекційного матеріалу, підготовка до лабораторного за- няття. Самостійне вивчення питань: "Оброблення матриць"	Використан- ня на лабо- раторних роботах	—
		12	Ауд.	2	Лабо- раторне заняття	<i>Лабораторне занят- тя 6. Обробка структур</i>	Робота на занятті. Захист лабо- раторної ро- боти 5
	СРС		3	Підго- товка до занять	Вивчення лекційного матеріалу, підготовка до лабораторного за- няття Самостійне вивчення питань: "Оброблення структур"	Використан- ня на лабо- раторних роботах	—

1	2	3	4		5	6		
ПТ*, ЗТ**, ФТ***	Розробляти програми, які використовують функції	13	Ауд.	2	Лекція	Тема 6. Функції	Робота на лекції	0,2
				2	Лабораторне заняття	Лабораторне заняття 6. Оброблення структур	Робота на занятті. Захист не передбачений	0,2
			СРС	10	Підготовка до занять	Вивчення лекційного матеріалу, підготовка до лабораторного заняття. Самостійне вивчення питань: "Визначення функцій"	Використання на лабораторних роботах	—
		14	Ауд.	2	Лабораторне заняття	Лабораторне заняття 7. Використання функцій	Робота на занятті. Захист лабораторної роботи 6	8,2
			СРС	10	Підготовка до занять	Вивчення лекційного матеріалу, підготовка до лабораторного заняття. Самостійне вивчення питань "Використання функцій"	Використання на лабораторних роботах	—
		ПТ*, ЗТ**, ФТ***	Розробляти програми, які використовують функції	15	Ауд.	2	Лекція	Тема 6. Функції
2	Лабораторне заняття					Лабораторне заняття 7. Використання функцій	Робота на занятті. Захист не передбачений	0,2
СРС	8				Підготовка до занять	Вивчення лекційного матеріалу, підготовка до лабораторного заняття. Самостійне вивчення питань "Використання функцій" Підготовка до контрольної роботи	Використання на лабораторних роботах	—
16	Ауд.			2	Лабораторне заняття	Лабораторне заняття 7. Використання функцій	Робота на занятті. Захист не передбачений	0,2

1		2	3		4		5	6
ПТ*, ЗТ**, ФТ***	Розробляти програми, які використовують функції	16	СРС	6	Підготовка до занять	Вивчення лекційного матеріалу, підготовка до лабораторного заняття. Самостійне вивчення питань "Використання функцій"	Використання на лабораторних роботах	–
			Ауд.	2	Лабораторне заняття	<i>Лабораторне заняття 7.</i> Використання функцій	Робота на занятті. Захист лабораторної роботи 7	24,2
		17	СРС	6	Підготовка до занять	Вивчення лекційного матеріалу, підготовка до лабораторного заняття. Самостійне вивчення питань "Використання функцій"	Використання на лабораторних роботах	–
Годин за семестр			148	Максимальна кількість балів за семестр			100	
3. Базові концепції об'єктно-орієнтованого програмування (ООП)								14,4
ПТ*, ЗТ**, ФТ***	Обґрунтовувати перелік і структуру класів, екземпляри яких найбільш адекватно забезпечують реалізацію заданого мультимедійного сценарію	1	Ауд.	2	Лекція	<i>Тема 7.</i> Особливості об'єктно-орієнтованого стилю програмування	Робота на лекції	0,2
				2	Лабораторне заняття	<i>Лабораторне заняття 8.</i> Використання простих класів	Робота на занятті. Захист не передбачений	0,2
			СРС	2	Підготовка до занять	Вивчення лекційного матеріалу, підготовка до лабораторного заняття. Самостійне вивчення питань "Особливості об'єктно-орієнтованого стилю програмування, виконання індивідуального завдання"	Використання на лабораторних роботах	–

1		2	3		4		5	6							
ПТ*, ЗТ**, ФТ*** Обґрунтовувати перелік і структуру класів, екземпляри яких найбільш адекватно забезпечують реалізацію заданого мультимедійного сценарію	2	СРС	3	Підготовка до занять	Вивчення лекційного матеріалу, підготовка до лабораторного заняття. Самостійне вивчення питань "Особливості об'єктно-орієнтованого стилю програмування, виконання індивідуального завдання	Використання на лабораторних роботах	–	–							
									Ауд.	2	Лабораторне заняття	Лабораторне заняття 8. Використання простих класів	Робота на занятті. Захист не передбачений	0,2	
		3	Ауд.	2	2	Лекція	Тема 8. Методи класів	Робота на лекції	0,2						
						Лабораторне заняття	Лабораторне заняття 9. Використання простих методів	Робота на занятті. Захист лабораторної роботи 8	3,2						
			СРС	5	5	Підготовка до занять	Вивчення лекційного матеріалу, підготовка до лабораторного заняття. Самостійне вивчення питань "Методи класів", виконання індивідуального завдання	Використання на лабораторних роботах	–						
										Ауд.	2	Лабораторне заняття	Лабораторне заняття 10. Визначення особливостей застосування конструкторів	Робота на занятті. Захист лабораторної роботи 9	3,2
ПТ*, ЗТ**, ФТ*** Застосувати методи-властивості, методи класу і майстри Visual Studio	4	СРС	5	Підготовка до занять	Вивчення лекційного матеріалу, підготовка до лабораторного заняття. Самостійне вивчення питань "Методи класів", виконання індивідуального завдання	Використання на лабораторних роботах	–								
								Ауд.	2	Лекція	Тема 8. Методи класів	Робота на лекції	0,2		
	5	Ауд.	2	2	Лабораторне заняття	Лабораторне заняття 10. Визначення особливостей застосування конструкторів	Робота на занятті. Захист не передбачений	0,2							
									4	СРС	5	Підготовка до занять	Вивчення лекційного матеріалу, підготовка до лабораторного заняття. Самостійне вивчення питань "Методи класів", виконання індивідуального завдання	Використання на лабораторних роботах	–

1		2	3		4		5	6
ПТ*, ЗТ**, ФТ***	Застосувати методи-властивості, методи класу і майстри Visual Studio	5	СРС	7	Підготівка до занять	Вивчення лекційного матеріалу, підготовка до лабораторного заняття. Самостійне вивчення питань "Методи класів", виконання індивідуального завдання	Використання на лабораторних роботах	–
						6		
		СРС	8	Підготівка до занять	Вивчення лекційного матеріалу, підготовка до лабораторного заняття. Самостійне вивчення питань "Відносини між класами", виконання індивідуального завдання		Використання на лабораторних роботах	–
					ПТ*, ЗТ**, ФТ***	Об'єднувати перелік і структуру класів, екземпляри яких найбільш адекватно забезпечують реалізацію заданого мультимедійного сценарію	7	Ауд.
СРС	3	Підготівка до занять	Вивчення лекційного матеріалу, підготовка до лабораторного заняття. Самостійне вивчення питань "Відносини між класами", виконання індивідуального завдання	Використання на лабораторних роботах				
			Ауд.	2			Лабораторне заняття	Лабораторне заняття 11. Розроблення програм з ієрархією класів
СРС	5	Підготівка до занять						Вивчення лекційного матеріалу, підготовка до лабораторного заняття. Самостійне вивчення питань "Відносини між класами", виконання індивідуального завдання
			8	Ауд.	2	Лабораторне заняття	Лабораторне заняття 11. Розроблення програм з ієрархією класів	Робота на занятті. Захист не передбачений
СРС	5	Підготівка до занять					Вивчення лекційного матеріалу, підготовка до лабораторного заняття. Самостійне вивчення питань "Відносини між класами", виконання індивідуального завдання	Використання на лабораторних роботах

1	2	3	4	5	6			
4. Організація мультимедійних програм і даних					85,6			
ПТ*, ЗТ**, ФТ***	Розробляти об'єктно-орієнтовані бізнес-додатки з елементами мультимедіа	9	Ауд.	2	Лекція	Тема 10. Принципи створення візуальних інтерфейсів	Робота на лекції	0,2
				2	Лабораторне заняття	Лабораторне заняття 12. Розроблення типового каркаса графічного Windows-додатка	Робота на занятті. Захист лабораторної роботи 11	5,2
			СРС	5	Підготовка до занять	Вивчення лекційного матеріалу, підготовка до лабораторного заняття. Самостійне вивчення питань "Принципи створення візуальних інтерфейсів". Виконання індивідуального завдання	Використання на лабораторних роботах	—
			Ауд.	2	Лабораторне заняття	Лабораторне заняття 12. Розробка типового каркаса графічного Windows-додатка	Робота на занятті. Захист передбачений	0,2
		11	Ауд.	2	Лекція	Тема 11. Програмування графіки	Робота на лекції	0,2
				2	Лабораторне заняття	Лабораторне заняття 13. Розроблення MDI- і SDI-додатків за допомогою компонентів Designer Forms	Робота на занятті. Захист лабораторної роботи 12	6,2

1		2	3		4		5	6
			СРС	5	Підгото- вка до занять	Вивчення лекційного мате- ріалу, підготовка до лабо- раторного заняття. Самостійне вивчення пи- тань "Програмування гра- фіки" Виконання індивідуа- льного завдання	Використан- ня на лабо- раторних ро- ботах	—
ПТ*, ЗТ**, ФТ***	Розробляти об'єктно-орієнтовані бізнес- додатки з елементами мультимедіа	12	Ауд.	2	Лабо- раторне заняття	<i>Лабораторне заняття 13.</i> Розроблення MDI- і SDI- додатків за допомогою компонентів Designer Forms	Робота на занятті. Захист не перед- бачений	0,2
			СРС	5	Підгото- вка до занять	Вивчення лекційного мате- ріалу, підготовка до лабо- раторного заняття. Самостійне вивчення пи- тань "MDI- і SDI-додатків". Виконання індивідуального завдання	Використан- ня на лабо- раторних ро- ботах	—
ПТ*, ЗТ**, ФТ***	Обґрунтовувати структуру класів, екземпляри яких забезпе- чують реалізацію мультимедійного сценарію	13	Ауд.	2	Лабо- раторне заняття	<i>Лабораторне заняття 14.</i> Розроблення Windows- додатків з елементами графіки	Робота на занятті. Захист лабо- раторної ро- боти 13	8,2
			СРС	3	Підгото- вка до занять	Вивчення лекційного мате- ріалу, підготовка до лабо- раторного заняття. Само- стійне вивчення питань "MDI- і SDI-додатків" Виконання індивідуального завдання	Використан- ня на лабо- раторних ро- ботах	—
		14	Ауд.	2	Лабо- раторне заняття	<i>Лабораторне заняття 14.</i> Розроблення Windows- додатків з елементами графіки	Робота на занятті. Захист не перед- бачений	0,2
			СРС	5	Підгото- вка до занять	Вивчення лекційного мате- ріалу, підготовка до лабо- раторного заняття. Само- стійне вивчення питань "MDI- і SDI-додатків" Виконання індивідуального завдання	Використан- ня на лабо- раторних ро- ботах	—

1	2	3	4	5	6		
ПТ*, ЗТ**, ФТ*** Розробляти об'єктно-орієнтовані бізнес-додатки з елементами мультимедіа	15	Ауд.	2	Лекція	Тема 12. Програмування додатків з елементами мультимедіа	Поточний контроль. Робота на лекції	1.2
			2	Лабораторне заняття	Лабораторне заняття 15. Розроблення мультимедійних Windows-додатків (звук, відео, анімація)	Робота на занятті. Захист лабораторної роботи 14	10, 2
		СРС	5	Підготовка до занять	Вивчення лекційного матеріалу, підготовка до лабораторного заняття. Самостійне вивчення питань "Розроблення мультимедійних Windows-додатків". Виконання індивідуального завдання	Використання на лабораторних роботах	—
	16	Ауд.	2	Лабораторне заняття	Лабораторне заняття 15. Розроблення мультимедійних Windows-додатків (звук, відео, анімація)	Робота на занятті. Захист не передбачений	0,2
		СРС	5	Підготовка до занять	Вивчення лекційного матеріалу, підготовка до лабораторного заняття. Самостійне вивчення питань "Розроблення мультимедійних Windows-додатків". Виконання індивідуального завдання	Використання на лабораторних роботах	—
	17	Ауд.	2	Лекція	Тема 12. Програмування додатків з елементами мультимедіа	Робота на лекції	0,2
			2	Лабораторне заняття	Лабораторне заняття 15. Розроблення мультимедійних Windows-додатків (звук, відео, анімація)	Робота на занятті. Захист лабораторної роботи 15	13
		СРС	5	Підготовка до занять	Вивчення лекційного матеріалу. Самостійне вивчення питань "Розроблення мультимедійних Windows-додатків" Виконання індивідуального завдання	Перевірка індивідуального завдання	—

1	2	3	4		5	6
СЕСІЯ	СРС	12	Передекзаменаційна консультація	Вирішення практичних завдань, що входять до підсумкового контролю	Підсумковий контроль	40
			Екзамен	Виконання завдань екзаменаційного білета		
			Підготовка до екзамену	Повторення матеріалів змістових модулів		
Годин за семестр		152	Максимальна кількість балів за семестр			100
з них						
Аудиторні		52	34 %	поточний контроль		60
самостійна робота		100	66 %	підсумковий контроль		40

* Предметна технологія.

** Забезпечувана технологія.

*** Функціональна технологія.

Розподіл балів у межах тем змістових модулів наведено в табл. 10.2.

Таблиця 10.2

Розподіл балів за темами

Поточне тестування та самостійна робота						Сума	
Перший семестр							
Змістовий модуль 1			Змістовий модуль 2			100	
T1	T2	T3	T4	T4	T5		T6
0,6	7,6	10,6	14,6	10,6	20,6		35,4
Другий семестр							
Змістовий модуль 3			Змістовий модуль 4			60	
T7	T8	T9	T10	T11	T12		
0.6	12,2	1,6	5,6	15,2	24,8		

Максимальну кількість балів, яку може накопичити студент протягом тижня за формами та методами навчання, наведено в табл. 10.3.

Розподіл балів за тижнями

Теми змістового модуля			Лек. заняття	Лаб. заняття	Контр. робота	Усього
1			2	3	4	5
1. Організація процедуро – орієнтований програма	Тема 1	1 тиждень	0,2	0,2		0,4
		2 тиждень		0,2		0,2
	Тема 2	3 тиждень	0,2	7,2		7,4
		4 тиждень		0,2		0,2
	Тема 3	5 тиждень	0,2	10,2		10,4
		6 тиждень		0,2		0,2
	Тема 4	7 тиждень	0,2	12,2	2	14,4
		8 тиждень		0,2		0,2
2. Організація і оброблення складених типів даних	Тема 4	9 тиждень	0,2	10,2		10,4
		10 тиждень		0,2		0,2
	Тема 5	11 тиждень	0,2	0,2		0,4
		12 тиждень		20,2		20,2
	Тема 6	13 тиждень	0,2	0,2		0,4
		14 тиждень		8,2		8,2
		15 тиждень	0,2	0,2	2	2,4
		16 тиждень		0,2		0,2
		17 тиждень		24,2		24,2
Усього за перший семестр			1,6	94,4	4	100
3. Базові концепції об'єктно-орієнтованого програмування (ООП)	Тема 7	1 тиждень	0,2	0,2		0,4
		2 тиждень		0,2		0,4
	Тема 8	3 тиждень	0,2	3,2		3,4
		4 тиждень		3,2		3,2
	Тема 8	5 тиждень	0,2	0,2		0,4
		6 тиждень		5,2		5,2
	Тема 9	7 тиждень	0,2	0,2	1	1,4
		8 тиждень		0,2		0,2
4. Організація мультимедійних програма і даних	Тема 10	9 тиждень	0,2	5,2		5,4
		10 тиждень		0,2		0,2
	Тема 11	11 тиждень	0,2	6,2		6,4
		12 тиждень		0,2		0,2
	Тема 11	13 тиждень	0,2	8,2		8,4
		14 тиждень		0,2		0,2
	Тема 12	15 тиждень	0,2	10,2	1	11,4

1		2	3	4	5
	16 тиждень		0,2		0,2
	17 тиждень	0,2	13		13,2
Екзамен					40
Усього за другій семестр		1,8	56.2	2	100

Підсумкова оцінка з навчальної дисципліни визначається відповідно до Тимчасового положення "Про порядок оцінювання результатів навчання студентів за накопичувальною бально-рейтинговою системою" ХНЕУ ім. С. Кузнеця (табл. 12.4).

Оцінки за цією шкалою заносяться до відомостей обліку успішності, індивідуального навчального плану студента та іншої академічної документації.

Таблиця 10.4

Шкала оцінювання: національна та ЄКТС

Сума балів за всі види навчальної діяльності	Оцінка ЄКТС	Оцінка за національною шкалою	
		для екзамену, курсового проєкту (роботи), практики	для заліку
90 – 100	A	відмінно	Зараховано
82 – 89	B	добре	
74 – 81	C		
64 – 73	D	задовільно	
60 – 63	E		
35 – 59	FX	Незадовільно	не зараховано
1 – 34	F		

11. Рекомендована література

11.1. Основна

1. Браткевич В. В. Методичні рекомендації до виконання лабораторних робіт з навчальної дисципліни "Основи програмування" для студентів напряму підготовки 6.051501 "Видавничо-поліграфічна справа" всіх форм навчання / В. В. Браткевич. – Х. : Вид. ХНЕУ, 2015. – 118 с.

2. Гаврилов В. П. Основи програмування. Конспект лекцій для студентів напряму підготовки 0927 "Видавничо-поліграфічна справа" усіх форм навчання / В. П. Гаврилов, В. В. Браткевич, І. О. Бондар. – Х. : Вид. ХНЕУ, 2007. – 172 с.

3. Петцольд Ч. Программирование для Microsoft Windows на C#. В 2-х томах. / Ч. Петцольд ; пер. с англ. – М. : Издательско-торговый дом "Русская Редакция", 2009. – 576 с. : ил.

11.2. Додаткова

4. Программист – программисту. C# / К. Ватсон и др. ; пер. с англ. – М. : Изд. "Лори", 2010. – 862 с.

5. C# без лишних слов / У. Робинсон ; пер. с англ. – М. : ДМК Пресс, 2010. – 352 с.

6. Си Шарп: Создание приложений для Windows / В. В. Лабор. – Мн. : Харвест, 2011 – 384 с.

11.3. Інформаційні ресурси

7. Полный справочник по C# [Электронный ресурс]. – Режим доступа : <http://skillcoding.com/Books.aspx?id=50>.

8. Полный справочник по C# [Электронный ресурс]. – Режим доступа : <http://www.kodges.ru/komp/program/9424-shildt-g.-polnyjj-spravochnik-ro-c.html>.

9. Справочник по C# [Электронный ресурс]. – Режим доступа : <https://msdn.microsoft.com/ru-ru/library/618ayhy6.aspx>.

Додатки

Додаток А
Таблиця А.1

Структура складових професійних компетентностей з навчальної дисципліни "Програмування засобів мультимедіа" за Національною рамкою кваліфікацій України

54

Складові компетентності, яка формується в рамках теми	Мінімальний досвід	Знання	Вміння	Комунікації	Автономність і відповідальність
1	2	3	4	5	6
Тема 1. Теоретичні та методологічні засади організації програм і даних					
Обирати технологію створення Windows-додатків	Принцип програмного управління комп'ютером. Стилi програмування	Зміст етапів створення Windows-додатків	Планувати роботу по створенню Windows-додатків	Взаємодія з замовником при прийомі замовлення на розроблення	Пошук шляхів вирішення проблем, що виникають при взаємодії з замовником
Обґрунтування вибору мови програмування С#	Характеристика С-образних мов програмування	Місце С-образних мов програмування в галузі розробки сучасних мультимедійних продуктів	Обґрунтування вибору мови програмування	Аргументовано обґрунтовувати вибір мови програмування у ході обговорення	Пошук альтернатив
Тема 2. Поняття типу даних					
Використовувати прості типи даних	Концепція типу даних	Характеристики і особливостей застосування простих типів	Створювати Windows-додатки з застосування простих типів даних	Аргументовано обґрунтовувати вибір простих типів даних у ході обговорення	Підготовка декількох варіантів можливих рішень щодо вибору простих типів даних та вибір найбільш доцільного
Використовувати вбудовані типи даних	Характеристика і особливості застосування вбудованих типів даних	Основні типи вбудованих типів даних та їх атрибути	Перетворювати вбудовані типи даних згідно поточного алгоритму	Обґрунтовувати рішення, що пропонуються	Використовувати альтернативні рішення

1	2	3	4	5	6
Тема 3. Програмування обчислювальних процесів					
Використовувати стандартні алгоритмічні структури для програмування обчислювальних процесів	Механізм дії базових структур створення алгоритмів програмування	Алгоритмічні структури управління програмою та варіанти їх реалізації мовою С#:	Застосовувати алгоритмічні структури управління програмою	Приймати зважене з урахуванням думки інших розробників рішення про зміст і місце використання алгоритмічних структур управління програмою	Здатність розробляти програми з урахуванням особливостей використання алгоритмічних структур управління програмою
Тема 4. Масиви					
Здійснювати оброблення масивів	Доступ до окремих елементів масиву. Ініціалізація масивів. Перебір елементів масиву	Загальні відомості про масиви. Особливості обробки масивів даних	Обробляти одномірні і багатомірні масиви	Надання допомоги учасникам проекту у виборі засобів оброблення масивів	Пошук альтернативних засобів обробки масивів
Тема 5. Структури					
Здійснювати оброблення структур	Доступ до окремих елементів масиву структур. Ініціалізація масиву структур. Перебір елементів масиву структур	Загальні відомості про структури та їх масиви. Особливості оброблення масивів структур	Обробляти масиви структур	Надання допомоги учасникам проекту у виборі засобів оброблення структур	Пошук альтернативних засобів оброблення структур
Тема 6. Функції					
Здійснювати застосування функцій	Призначення, опис та виклик функцій	Механізми опису і виклику функцій. Способи обміну інформацією з функцією	Розробляти програми з застосування стандартних функцій та функцій користувача	Надання допомоги учасникам проекту у виборі переліку функцій	Пошук альтернативних засобів опису функцій

1	2	3	4	5	6
Тема 7. Особливості об'єктно-орієнтованого стилю програмування					
Використовувати об'єктно-орієнтований стиль програмування для реалізації певних сценаріїв замовника	Особливості об'єктно-орієнтованого стилю програмування	Визначення об'єкта. Принципи ООП: абстрагування, інкапсуляція, спадкування	Здійснювати доступ до полів класу	Консультації учасників проекту щодо доцільності застосування певних полів класу	Здатність знаходити компромісні рішення
Тема 8. Методи класів					
Використовувати методи для опису взаємодії об'єктів під час реалізації певних сценаріїв замовника	Особливості застосування методів. Порівняння методів і функцій	Визначення методів. Механізм обміну інформацією між методами. Виклики методів. Рекурсивні методи	Здійснювати опис методу та його відповідний виклик. Опрацьовувати методи-властивості. Індексатори. Статичні поля і методи класу	Консультації учасників проекту щодо доцільності застосування певних методів класу	Пошук альтернативних засобів реалізації подібних методів
Тема 9. Відносини між класами					
Використовувати відношення "клієнт – постачальник" та відношення "батько – спадкоємець" під час реалізації певних сценаріїв замовника	Типи відносин між класами	Проста ієрархія класів. Використовування захищеного доступу. Конструктори і спадкоємство. Визначення абстрактних класів. Віртуальні методи і їх перевизначення. Інтерфейси. Синтаксис визначення інтерфейсів. Порівняння інтерфейсів і абстрактних класів	Здійснювати розроблення об'єктно-орієнтованих додатків з застосуванням відносин між класами	Консультації учасників проекту щодо доцільності застосування певних відносин між класами	Пошук альтернативних засобів реалізації відносин між класами

Закінчення додатка А
Закінчення табл. А.1

1	2	3	4	5	6
Тема 10. Принципи створення візуальних інтерфейсів					
Застосування Майстра Visual Studio для роботи з елементами класу	Спадкоємець Form для головної форми. Ресурси програми. Файл для логіки модуля	Простори імен. Потoki. Клас Application. Аналіз коду, згенерований дизайнером. Властивості форми. Методи форми. Діалогові вікна. Немодальні вікна. Події та їх виклик. Події на прикладі форми. Використання власних делегатів	Здійснювати розроблення об'єктно-орієнтованих додатків, в яких застосовуються: методи форми, діалогові вікна, немодальні вікна, багатодокументний інтерфейс, компоненти .NET, шаблони, колекції	Консультації учасників проекту щодо доцільності застосування певних компонентів візуальних інтерфейсів	Здатність знаходити компромісні рішення
Тема 11. Програмування графіки					
Застосування технології GDI+ для створення графічних додатків	Поверхня малювання Graphics. Класи та методи GDI+	Графічні структури, перо Pen, пензель Brush (одноколірні, лінійні складні градієнтні, текстурні і штрихові), колір Color, матриці кольору і трансформацій. Координатна система GDI+. Колірні моделі та формати графічних файлів	Розробляти мультимедійний контент у Windows Forms графічних додатках	Консультації учасників проекту щодо доцільності застосування певного мультимедійного контенту в Windows Forms графічних додатках	Здатність розробляти програми з урахуванням особливостей застосування певного мультимедійного контенту в Windows Forms графічних додатках
Тема 12. Програмування додатків з елементами мультимедіа					
Використовувати Середовище Visual Studio для розробки графічних додатків зі звуком, відео та анімацією	Здатність уявити сценарій роботи мультимедійного застосування у вигляді послідовності процедур (дій), які здійснюють обробку відповідних даних	Принципи створення мультимедійних ігор (стратегії, головоломки, "бродилки", "стрілялки" тощо). Порівняльний огляд можливостей та галузей застосування графічних додатків Windows Form Application, WPW Application, Silverlight Application	Розробляти графічні додатки зі звуком, відео та анімацією	Взаємодія з замовником під час прийому замовлення на розробку	Пошук компромісів

Зміст

Вступ.....	3
1. Опис навчальної дисципліни	4
2. Мета та завдання навчальної дисципліни	4
3. Програма навчальної дисципліни	8
4. Структура навчальної дисципліни.....	13
5. Теми лабораторних занять.....	15
6. Самостійна робота.....	19
6.1. Контрольні запитання для самодіагностики.....	23
7. Індивідуально-консультативна робота	28
8. Методи навчання	29
9. Методи контролю	31
10. Розподіл балів, які отримують студенти	38
11. Рекомендована література.....	53
11.1. Основна.....	53
11.2. Додаткова	53
11.3. Інформаційні ресурси.....	53
Додатки.....	54

НАВЧАЛЬНЕ ВИДАННЯ

Робоча програма
навчальної дисципліни
**"ПРОГРАМУВАННЯ ЗАСОБІВ
МУЛЬТИМЕДІА"**

для студентів напряму підготовки
6.051501 "Видавничо-поліграфічна справа"
денної форми навчання

Укладач **Браткевич** Вячеслав Вячеславович

Відповідальний за видання *О. І. Пушкар*

Редактор *В. О. Бутенко*

Коректор *В. О. Бутенко*

План 2016 р. Поз. № 160 ЕВ. Обсяг 59 с.

Видавець і виготовлювач – ХНЕУ ім. С. Кузнеця, 61166, м. Харків, просп. Науки, 9-А

*Свідоцтво про внесення суб'єкта видавничої справи до Державного реєстру
ДК № 4853 від 20.02.2015 р.*