

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Львівський національний університет імені Івана Франка**  
**Факультет прикладної математики та інформатики**  
**Кафедра програмування**

**Затверджено**

На засіданні кафедри програмування  
факультету прикладної математики та  
інформатики  
Львівського національного університету  
імені Івана Франка  
(протокол № 1 від 29 серпня 2023 р.)



Зав. кафедри к. ф.-м. н., доц. Ярошко С. А.

**Силабус навчальної дисципліни**  
**«Програмне забезпечення»,**  
що викладається в межах першого (бакалаврського) рівня  
вищої освіти для здобувачів  
з спеціальності 113 Прикладна математика

Львів 2023 р.

<b>Назва дисципліни</b>	Програмне забезпечення
<b>Адреса викладання дисципліни</b>	Львівський національний університет імені Івана Франка, вул. Університетська 1, м. Львів, Україна, 79000
<b>Факультет та кафедра, за якою закріплена дисципліна</b>	Факультет прикладної математики та інформатики, кафедра програмування
<b>Галузь знань, шифр та назва спеціальності</b>	Галузь знань 11 Математика та статистика Предметна спеціальність 113 Прикладна математика
<b>Викладачі дисципліни</b>	Пасічник Тимофій Васильович, кандидат фізико-математичних наук, ст. наук. співробітник, доцент кафедри програмування
<b>Контактна інформація викладачів</b>	Електронна пошта: <a href="mailto:tymofiy.pasichnyk@lnu.edu.ua">tymofiy.pasichnyk@lnu.edu.ua</a> веб-сторінка: <a href="https://ami.lnu.edu.ua/employee/pasichnyk-t-v">https://ami.lnu.edu.ua/employee/pasichnyk-t-v</a>
<b>Консультації з питань навчання по дисципліні відбуваються</b>	Консультації проводять раз на тиждень згідно з оприлюдненим розкладом консультацій викладача. Можливі он-лайн консультації через Microsoft Teams. Для погодження часу он-лайн консультацій слід писати на електронну пошту викладача.
<b>Сторінка курсу</b>	<a href="https://ami.lnu.edu.ua/course/software-applied-mathematics">https://ami.lnu.edu.ua/course/software-applied-mathematics</a>
<b>Інформація про дисципліну</b>	Курс «Програмне забезпечення» є нормативною дисципліною з спеціальності 113 Прикладна математика. У четвертому семестрі її викладають в обсязі 5 кредитів (за Європейською кредитно-трансферною системою ECTS)
<b>Коротка анотація дисципліни</b>	Фокус уваги курсу спрямовано на оволодіння базовими інструментами середовища .Net та особливості мови програмування C#: властивості, події, індикатори, методи розширення, використання інтерфейсів, серіалізація, життєвий цикл об'єктів, засоби рефлексії, LINQ, багатопотокове програмування. Розглянуто засоби Windows Forms для побудови застосунків.
<b>Мета та цілі дисципліни</b>	Метою нормативної дисципліни «Програмне забезпечення» є ознайомити студента з базовими можливостями платформи .Net, навчити його використовувати мову C# для створення ефективних застосунків, приготувати до вивчення курсу "Програмне забезпечення II".
<b>Література для вивчення дисципліни</b>	<i>Основна література</i> 1. Learn C#. OFFICIAL COLLECTION: <a href="https://learn.microsoft.com/enus/users/dotnet/collections/yz26f8y64n7k07?WT.mc_id=dotnet-35129-website">https://learn.microsoft.com/enus/users/dotnet/collections/yz26f8y64n7k07?WT.mc_id=dotnet-35129-website</a> 2. C# programming guide <a href="https://learn.microsoft.com/en-us/dotnet/csharp/programmingguide/?WT.mc_id=dotnet-35129-website">https://learn.microsoft.com/en-us/dotnet/csharp/programmingguide/?WT.mc_id=dotnet-35129-website</a> 3. Object-oriented programming (C#) <a href="https://learn.microsoft.com/enus/dotnet/csharp/fundamentals/tutorials/oop?source=recommendations">https://learn.microsoft.com/enus/dotnet/csharp/fundamentals/tutorials/oop?source=recommendations</a> 4. Inheritance in C# and .NET <a href="https://learn.microsoft.com/enus/dotnet/csharp/fundamentals/tutorials/inheritance">https://learn.microsoft.com/enus/dotnet/csharp/fundamentals/tutorials/inheritance</a> 5. Language Integrated Query (LINQ) <a href="https://learn.microsoft.com/en-us/dotnet/csharp/programmingguide/concepts/linq">https://learn.microsoft.com/en-us/dotnet/csharp/programmingguide/concepts/linq</a> 6. Troelsen Andrew, Japikse Phil. Pro C# 10 with .NET 6: Foundational Principles and Practices in Programming – APress, 2022 – 1640 pp. 7. Christian Nagel Professional C# and .NET, 2021st Edition – Wrox, 2021 – 1800 pp. <i>Додаткова література</i> 8. Mark J. Price C# 10 and .NET 6 – Modern Cross-Platform Development – Packt Publishing, 2021 – 826 pp. 9. Joseph Albahari C# 10 in a Nutshell: The Definitive Reference – O'Reilly Media, 2022 – 1000 pp. 10. Ian Griffiths C# 10 in a Nutshell – O'Reilly Media, 2022 – 833 pp. 11. Rob Miles Exam Ref 70-483 Programming in C#, Second Edition – Microsoft Press, 2018 – 433 pp.
<b>Обсяг курсу</b>	5 кредитів ЄКТС – 150 годин. З них 32 години лекцій, 32 години лабораторних

<p><b>Очікувані результати навчання</b></p>	<p>занять та 86 годин самостійної роботи</p> <p>Після завершення цього курсу студент буде:</p> <p><i>знати</i></p> <ul style="list-style-type: none"> <li>• засади функціонування платформи .Net;</li> <li>• синтаксис мови програмування C#, прийоми об'єктно-орієнтованого програмування мовою C#;</li> <li>• можливості та правила використання стандартних класів, зокрема, контейнерів, стандартних інтерфейсів;</li> <li>• призначення та влаштування компонент бібліотеки Windows Forms.</li> </ul> <p><i>вміти</i></p> <ul style="list-style-type: none"> <li>• оголошувати та використовувати у програмах мовою C# ієрархії класів, реалізовувати стандартні та власні інтерфейси;</li> <li>• налагоджувати взаємодію об'єктів через події;</li> <li>• серіалізувати об'єкти, програмно взаємодіяти з файловою системою;</li> <li>• використовувати LINQ для опрацювання колекцій даних;</li> <li>• будувати застосунки за допомогою Windows Forms;</li> <li>• створювати багатопотокові застосунки.</li> </ul> <p>Курс забезпечує набуття таких компетентностей та програмних результатів навчання:</p> <p>Фахові компетентності спеціальності:</p> <p>ФК04. Здатність розробляти алгоритми та структури даних, програмні засоби та програмну документацію.</p> <p>ФК05. Здатність проектувати бази даних, інформаційні системи та ресурси.</p> <p>ФК08. Здатність використовувати сучасні технології програмування та тестування програмного забезпечення.</p> <p>ФК21. Здатність аналізувати можливості наявних технологій розробки програмного забезпечення, обирати та застосовувати найбільш ефективний інструментарій відповідно до специфіки завдання.</p> <p>Програмні результати навчання:</p> <p>РН11. Вміти застосовувати сучасні технології програмування та розроблення програмного забезпечення, програмної реалізації чисельних і символічних алгоритмів.</p> <p>РН23. Порівнювати, обирати та застосовувати інструментарій для розробки програмного забезпечення, виходячи з міркувань ефективності відповідно до специфіки завдання.</p>				
<p><b>Ключові слова</b></p>	<p>Платформа .Net, мова програмування C#, клас, інтерфейс, властивість, подія, делегат, LINQ, серіалізація, багатопотоковість, Windows Forms.</p>				
<p><b>Формат курсу</b></p>	<p>Очний.</p>				
<p><b>Теми</b></p>	<p>Тижд.</p>	<p>Тема, план, короткі тези</p>	<p>Форма заняття</p>	<p>Тривалість, год</p>	<p>Термін виконання</p>
	1	<p>Архітектура, призначення, можливості платформи .Net. Система типів, операції мови C#. Типи-значення та типи-посилання C#. Консольне введення-виведення. Конструкції керування мови C#. Методи System. Object. Перелік - найпростіший тип користувача. Масиви C#: одно- та багатовимірні, зубчасті.</p>	Лекція	2	
		<p>Побудова консольних програм в середовищі Visual Studio засобами мови C#. Форматоване виведення. Послідовні, галужені, циклічні алгоритми з використанням вбудованих типів.</p>	Лабораторна робота	2	Наступне лабораторне заняття
		<p>Ознайомлення з загальною характеристикою платформи .Net</p>	Самостійна робота	5	
	2	<p>Оголошення класів і структур мовою C#: поля, методи, властивості. Передавання аргументів методам. Конструювання, копіювання об'єктів. Статичні члени класу, статичні класи. Методи, що розширюють клас.</p>	Лекція	2	
		<p>Опрацювання масивів, у тому числі зубчастих. Оголошення та використання переліків.</p>	Контрольна робота	2	
		<p>Побудова консольних застосунків зі складними циклічними алгоритмами. Використання масивів.</p>	Самостійна робота	5	
	3	<p>Операції мови C#: is, as, поглинання null та інші спеціальні. Перетворення типів. Перевірка рівності об'єктів. Перевантаження операцій. Методи-індексатори.</p>	Лекція	2	
		<p>Оголошення та використання структур. Оголошення та використання простих класів. Побудова та використання властивостей.</p>	Лабораторна робота	2	Наступне лабораторне заняття
		<p>Оголошення та використання класів. Використання нових (порівняно з C++) можливостей класів.</p>	Самостійна робота	5	

4	Інтерфейси: означення, можливості використання. Стандартні інтерфейси IComparable, IEnumerable, ICloneable. Наслідування класів та інтерфейсів. Абстрактні класи. Віртуальні та перекриті методи. Конструктори підкласів. Закриті класи і методи. Часткові класи.	Лекція	2	
	Перевантаження операцій. Використання власних індексаторів. Порівняння об'єктів. Перетворення типів.	Лабораторна робота	2	Наступне лабораторне заняття
	Вивчення вимог стандартних інтерфейсів.	Самостійна робота	5	
5	Делегати, події в мові C#: визначення типу делегата, використання простого та групового делегатів, масиву делегатів, анонімні методи і лямбда-вирази. Визначення події, взаємодія об'єктів через подію.	Лекція	2	
	Оголошення та використання інтерфейсів. Оголошення класів, що реалізують стандартні інтерфейси.	Лабораторна робота	2	Наступне лабораторне заняття
	Вивчення правил оголошення та використання узагальнених типів. Обмеження.	Самостійна робота	5	
6	Розробка узагальнених типів. Узагальнення класів. Узагальнення інтерфейсів. Обмеження узагальнень. Засоби узагальнених класів. Питання обсягу коду та продуктивності.	Лекція	2	
	Побудова ієрархії класів. Оголошення і використання конструкторів, фіналізаторів, віртуальних методів.	Контрольна робота	2	
	Проектування та побудова ієрархій класів. Побудова поліморфних колекцій.	Самостійна робота	5	
7	Колекції .Net, узагальнені колекції. Списки (зв'язні, сортовані), стек, черга, словник, бітовий масив. Перехоплення та опрацювання винятків. Класи винятків користувача, запуск винятків.	Лекція	2	
	Синхронний та асинхронний виклики методів через делегата. Визначення події, організація взаємодії об'єктів через подію.	Лабораторна робота	2	Наступне лабораторне заняття
	Експерименти з використанням делегатів. Налаштування взаємодії об'єктів через подію.	Самостійна робота	5	
8	Простір System.Reflection. Програмний доступ до метаданих. Дослідження збірки, типу. Атрибути користувача та їх використання. Приклади. Доступ до приватних членів класу. Динамічний виклик методу.	Лекція	2	
	Розробка узагальнених класів. Створення класів винятків. Запуск, перехоплення та опрацювання винятків.	Лабораторна робота	2	Наступне лабораторне заняття
	Експерименти з запуском і перехопленням винятків.	Самостійна робота	5	
9	Файлове введення-виведення і серіалізація об'єктів. Доступ до файлової системи. Маніпулювання файлами і директоріями. Потoki даних. Формати серіалізації: двійковий, XML, SOAP.	Лекція	2	
	Використання інструментів рефлексії для програмного дослідження класів. Побудова та використання атрибутів користувача.	Лабораторна робота	2	
	Динамічне завантаження бібліотек на етапі виконання програми. Дослідження типів бібліотеки.	Самостійна робота	5	
10	Побудова Windows-аплікацій з використанням WPF. Стандартні елементи керування. Діалоги. Приклади побудови застосунків: перша програма, модель світлофора.	Лекція	2	
	Використання колекцій .Net. Робота з рядками, текстом.	Лабораторна робота	2	Наступне лабораторне заняття
	Програмна взаємодія з файловою системою. Серіалізація, десеріалізація колекцій об'єктів.	Самостійна робота	5	
11	Приклади побудови застосунків: тестування, редактор текстів, багатівіконні програми. Меню, діалоги, панелі інструментів, рядок стану.	Лекція	2	
	Програмний пошук папок, файлів. Зберігання до файла (у різних форматах) та завантаження об'єктів з файла.	Лабораторна робота	2	Наступне лабораторне заняття
	Вивчення основних прийомів проектування графічного інтерфейсу користувача.	Самостійна робота	6	
12	Створення складних Windows Forms User Control, забезпечення підтримки етапу проектування аплікації	Лекція	2	
	Побудова Windows-аплікацій зі стандартними елементами керування WPF. Файлові діалоги.	Лабораторна робота	2	Наступне лабораторне заняття
	Побудова віконних застосунків, що ведуть діалог з користувачем.	Самостійна робота	6	
13	Багатопотоковість і синхронізація. Асинхронні делегати. Класи Thread, Task. Синхронізація. Багатопотокові аплікації	Лекція	2	
	Побудова багатопотокових Windows-аплікацій: окремий потік для взаємодії з GUI, окремий для обчислень тощо.	Контрольна робота	2	
	Експерименти з запуском обчислень у окремому	Самостійна	6	

		потоці, синхронізація.	робота		
	14	Засоби мови, що підтримують LINQ. LINQ to objects. Блок ітератора. Приклади.	Лекція	2	
		LINQ: розширення типів, лямбда вирази, LINQ-запити	Лабораторна робота	2	Наступне лабораторне заняття
		Тренування щодо застосування типових прийомів використання засобів LINQ	Самостійна робота	6	
	15	LINQ to XML провайдер. Типи XElement, XContainer, XDocument. Побудова XML-дерева.	Лекція	2	
		Побудова колекцій об'єктів на основі XML описів, опрацювання засобами LINQ.	Лабораторна робота	2	Наступне лабораторне заняття
		Експерименти з опрацюванням XML-документів.	Самостійна робота	6	
	16	Життєвий цикл об'єктів. Finalize. IDisposable. Менеджер керованої купи, збирання сміття.	Лекція	2	
		Взаємодія з менеджером купи. Реалізація фіналізаторів. Реалізація інтерфейсу IDisposable.	Контрольна робота	2	
		Підготовка до контрольної роботи	Самостійна робота	6	
<b>Підсумковий контроль, форма</b>	залік				
<b>Пререквізити</b>	Для вивчення курсу студенти потребують знань з попередніх семестрів навчання: основи ООП, використання винятків, взаємодія з потоками, використання контейнерів, модульне тестування.				
<b>Навчальні методи та техніки, які використовують під час викладання курсу</b>	Лекції з мультимедійними презентаціями та практичного використання середовища програмування; лабораторні заняття у вигляді проектування алгоритмів і програм, виконання практичних завдань, у тому числі у команді з 3-4 осіб; самостійне опрацювання навчальних матеріалів: підручників, конспектів лекцій, готових програм мовою C++, додаткових навчальних посібників, розміщених у хмарному сховищі (Moodle, Microsoft Teams, Google Classroom). Обговорення теоретичного та практичного матеріалу в онлайн сервісах, формулювання творчих завдань для студентів, виконання яких готує до вивчення нового теоретичного матеріалу.				
<b>Необхідне обладнання</b>	Для проведення лекцій: комп'ютер, проектор, доступ до мережі інтернет. Для проведення лабораторних та виконання завдань: комп'ютер, ОС Windows, доступ до інтернету, середовище програмування мовою C# (Microsoft Visual Studio). Уся література, яку студенти не зможуть знайти самостійно, буде надана викладачем виключно в освітніх цілях без права її передачі третім особам. Студенти заохочуються до використання також й іншої літератури та джерел, яких немає серед рекомендованих.				
<b>Критерії оцінювання (окремо для кожного виду навчальної діяльності)</b>	<p><b>Оцінювання</b> проводиться за 100-бальною шкалою. 80 балів нараховують за виконання лабораторних завдань і контрольних робіт, ще 20 балів – за тестування в системі MOODLE, або виконання комплексного індивідуального завдання. Лабораторні завдання можуть бути індивідуальні та командні. Упродовж семестру студент виконує не менше 10 лабораторних робіт, кожна з яких оцінюють 5-10 балів залежно від складності. Оцінка за тестування, колоквиуми тощо та за написання ускладненої комп'ютерної програми (декількох програм) оцінюється в 20 балів. Завдання до курсу можуть передбачати самостійне опрацювання актуальних публікацій щодо програмування для .Net.</p> <p><b>Відвідання занять</b> є важливою складовою навчання. Очікується, що всі студенти відвідають усі лекції і лабораторні заняття курсу. Активність під час проведення лекцій і лабораторних заохочується балами. У будь-якому випадку студенти зобов'язані дотримуватися усіх строків визначених для виконання усіх видів письмових робіт, передбачених курсом. Виконані роботи завантажують у відповідне хмарне сховище. Альтернативою відвідування лабораторних занять в університеті може бути дистанційна онлайн робота за розкладом проведення занять. Активність на лекціях і лабораторних ураховують при оцінюванні відповідного лабораторного завдання.</p> <p><b>Академічна доброчесність:</b> очікується, що роботи студентів будуть їхнім оригінальними дослідженнями чи міркуваннями. Відсутність посилань на використані джерела, фабрикування джерел, списування, втручання в роботу інших студентів, здавання чужих комп'ютерних програм як своїх становлять, але не обмежують, приклади можливої академічної недоброчесності. Виявлення ознак академічної недоброчесності в письмовій роботі студента є підставою</p>				

	<p>для її незарахування викладачем, незалежно від масштабів плагіату чи обману. Жодні форми порушення академічної доброчесності не толеруються.</p>
<p><b>Запитання до колоквиуму</b></p>	<ol style="list-style-type: none"> <li>1. Охарактеризуйте склад середовища .NET.</li> <li>2. Засоби середовища .NET компіляції та виконання програм.</li> <li>3. Система типів мови програмування C#.</li> <li>4. Консольне введення мовою C#. Як вводити числові дані?</li> <li>5. Консольне виведення мовою C#. Які засоби форматування ви знаєте?</li> <li>6. Тип перелік мови C#. Приклад оголошення, використання, що пояснює можливості типу.</li> <li>7. Тип перелік мови C#. Введення-виведення значень типу перелік.</li> <li>8. Типи одновимірний масив, багатовимірний масив мови C#. Наведіть приклади оголошення, ініціалізації, використання.</li> <li>9. Зубчастий масив мови C#. Поясніть структуру, можливості, наведіть приклад оголошення, ініціалізації, використання.</li> <li>10. Оператор перебору послідовності мови C#. Приклад. Умови, за яких його можна використовувати.</li> <li>11. Укажіть всі способи перебирання елементів одновимірного та двовимірного масивів у мові програмування C#.</li> <li>12. Копіювання, впорядкування масивів у мові програмування C#.</li> <li>13. Методи класу System.Object у мові C#.</li> <li>14. Що таке кортеж у мові C#? Як його оголошують, використовують?</li> <li>15. Охарактеризуйте тип структури мови C#</li> <li>16. З якою метою застосовують блок ітератора? Наведіть приклад використання, поясніть алгоритм його функціонування.</li> <li>17. Охарактеризуйте члени-дані та члени-функції класу у мові C# (вичерпний перелік).</li> <li>18. Охарактеризуйте модифікатори доступу класу у мові C#.</li> <li>19. Охарактеризуйте модифікатори оголошення (не доступу) у мові C#.</li> <li>20. Охарактеризуйте конструктори класу у мові C#. Як з одного конструктора викликати інший?</li> <li>21. Що ви знаєте про параметри методів у мові C# (оголошення та використання)?</li> <li>22. Охарактеризуйте властивості (property) класу у мові C#.</li> <li>23. Що ви знаєте про автоматичні властивості у мові C#?</li> <li>24. Статичні класи у мові C#: призначення оголошення, використання.</li> <li>25. Розширення функціональності класу у мові C# (без доступу до самого класу).</li> <li>26. Синтаксис наслідування класів у мові C#. Конструктори підкласу.</li> <li>27. Особливості оголошення та використання абстрактного класу мовою C#.</li> <li>28. Як оголошують та перевизначають віртуальні методи мовою C#?</li> <li>29. Сумісність і приведення споріднених типів у мові C#. Перевірка типу.</li> <li>30. Охарактеризуйте поняття інтерфейсу у мові C#. Наведіть приклад наслідування інтерфейсу класом мови C#.</li> <li>31. Охарактеризуйте інтерфейси IEnumerable, IEnumerator мови C#.</li> <li>32. Охарактеризуйте інтерфейси ICloneable, IComparable мови C#.</li> <li>33. Типи, що допускають null мови C#. Операція поглинання нуля мови C#.</li> <li>34. Перевірка рівності об'єктів мовою C#. Опишіть усі варіанти.</li> <li>35. Як перевага операцію мовою C#? Наведіть приклад для однієї операції.</li> <li>36. Методи індексатори, методи перетворення типу мовою C#.</li> <li>37. Охарактеризуйте тип делегата мови C#. Члени класу System.MulticastDelegate</li> <li>38. Використання групових делегатів, анонімних делегатів, лямбда-виразів мовою C#</li> <li>39. Охарактеризуйте поняття події мовою C#. Які кроки потрібно виконати, щоб оголосити подію у власному класі? Як її використати?</li> <li>40. Наведіть приклад взаємодії об'єктів власних класів через механізм подій (наприклад, подія зміни властивості).</li> <li>41. Які можливості є у мови C# для побудови контейнера елементів довільного типу?</li> <li>42. До чого можна застосувати узагальнення в мові C#? Перелічіть переваги узагальнень.</li> <li>43. Використання обмежень для оголошення узагальнених класів мовою C#.</li> </ol>

	<p>44. З якою метою застосовують винятки? Які класи винятків мови програмування C# визнаєте?</p> <p>45. Опишіть засоби мови C# для перехоплення та опрацювання винятків.</p> <p>46. Огляд контейнерів простору імен System.Collections мови C#.</p> <p>47. Огляд контейнерів простору імен System.Collections.Specialized мови C#.</p> <p>48. Огляд контейнерів простору імен System.Collections.Generic мови C#.</p> <p>49. Індексована колекція змінного розміру мови C#.</p> <p>50. Контейнери черга, стек мови C#.</p> <p>51. Контейнери списки мови C#.</p> <p>52. Контейнери словники мови C#.</p> <p>53. Огляд класів простору імен System.IO мови C#.</p> <p>54. Опишіть, для чого і як використовують Класи Directory, DirectoryInfo мови C#.</p> <p>55. Опишіть, для чого і як використовують Класи File, FileInfo мови C#.</p> <p>56. Опишіть, як мовою C# вивести інформацію до текстового файлу.</p> <p>57. Опишіть, як мовою C# прочитати дані з текстового файлу.</p> <p>58. Опишіть, як мовою C# використовують двійкові файли, для чого.</p> <p>59. Опишіть один зі способів серіалізації об'єктів мовою C#.</p> <p>60. Що ви знаєте про клас GarbageCollector мови C#?</p> <p>61. Як оголошують та як використовують метод-фіналізатор мовою C#? У яких випадках потрібні фіналізатори?</p> <p>62. Охарактеризуйте інтерфейс IDisposable мови C#. Які класи наслідують його?</p> <p>63. Опишіть шаблон використання фіналізатора в класі, що реалізує IDisposable мови C#.</p> <p>64. Наведіть приклад «лінивого» створення об'єктів мовою C#.</p> <p>65. Що таке програмний потік ОС Windows? Для чого програмі декілька потоків? Поясніть загальні засади написання таких програм. Які проблеми багатопотокових програм ви знаєте? Як їх вирішують?</p> <p>66. Які можливості класу Thread мови C# ви знаєте? Наведіть приклад використання.</p> <p>67. З якою метою використовують клас C# BackgroundWorker? Опишіть його можливості та налаштування.</p> <p>68. Які можливості класу Task мови C# ви знаєте? Наведіть приклад використання.</p> <p>69. Як отримати доступ до типу довільного об'єкта C# програми? Поясніть на прикладах. Які властивості типу можна дослідити?</p> <p>70. Як програмно отримати перелік методів, конструкторів довільного об'єкта C# програми? Наведіть приклад.</p> <p>71. Як програмно отримати перелік полів, властивостей довільного об'єкта C# програми? Наведіть приклад.</p> <p>72. Як програмно отримати доступ до C# збірки з метою дослідження? Які властивості збірки можна одразу ж перевірити?</p> <p>73. Як програмно отримати колекцію типів, визначених у C# збірці?</p> <p>74. Що таке атрибути програмного коду C#? Для чого їх використовують? Наведіть приклад</p> <p>75. Як визначити власні атрибути програмного коду C#? Як їх використати?</p> <p>76. Які види застосунків WPF ви знаєте? Охарактеризуйте стисло особливості кожного.</p> <p>77. Охарактеризуйте призначення та можливості основних класів, які використовують для побудови WPF-застосунку.</p> <p>78. Яку роль відіграє XAML для побудови WPF-застосунку? Назвіть основні синтаксичні елементи XAML, наведіть приклади.</p> <p>79. Охарактеризуйте елементи керування WPF: кнопки, перемикачі, списки.</p> <p>80. Охарактеризуйте елементи керування WPF: поля введення, таблиці, написи.</p> <p>81. Охарактеризуйте елементи керування WPF: меню, інструментальні панелі.</p> <p>82. Охарактеризуйте діалогові вікна WPF.</p> <p>83. Охарактеризуйте LINQ to XML провайдер.</p> <p>84. Охарактеризуйте типи XElement, XContainer, XDocument.</p> <p>85. Як виконується побудова XML-дерева?</p>
<p><b>Опитування</b></p>	<p>Анкету-оцінку з метою оцінювання якості курсу буде надано після завершення курсу.</p>