

**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 2024 р.)



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

**Силабус навчальної дисципліни**  
**«Програмування» (IV семестр),**  
що викладається в межах  
**ОПП «Середня освіта (Інформатика)»**  
*для здобувачів першого (бакалаврського) рівня вищої освіти*  
з предметної спеціальності **014.09 Середня освіта (Інформатика)**  
галузі знань **01 Освіта/Педагогіка**

Львів 2024 р.

|  |   |
|--|---|
| <b>Назва дисципліни</b>  | Програмування   |
| <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 <i>01 Освіта/Педагогіка</i><br>Предметна спеціальність <i>014.09 Середня освіта (Інформатика)</i>  |
| <b>Викладачі дисципліни</b>                                      | Ярошко Сергій Адамович, к. ф.-м. н., доцент, завідувач кафедри програмування; Пасічник Тимофій Васильович, к. ф.-м. н., доцент, доцент кафедри програмування  |
| <b>Контактна інформація викладачів</b>                           | Електронна пошта: <a href="mailto:serhiy.yaroshko@lnu.edu.ua">serhiy.yaroshko@lnu.edu.ua</a> , <a href="mailto:tymofiy.pasichnyk@lnu.edu.ua">tymofiy.pasichnyk@lnu.edu.ua</a><br>веб-сторінки: <a href="https://ami.lnu.edu.ua/employee/yaroshko">https://ami.lnu.edu.ua/employee/yaroshko</a><br><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/prohramuvannia-chastyna-2-python">https://ami.lnu.edu.ua/course/prohramuvannia-chastyna-2-python</a>   |
| <b>Інформація про дисципліну</b>                                 | Курс "Програмування" є нормативною дисципліною зі освітньо-професійної програми «Середня освіта (Інформатика)» першого (бакалаврського) рівня вищої освіти з предметної спеціальності 014.09 Середня освіта (Інформатика). У четвертому семестрі її викладають в обсязі 4 кредити (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 «Програмування» є ознайомити студента з базовими можливостями платформи .Net, навчити його використовувати мову C# для створення ефективних застосунків, приготувати до вивчення курсу "Програмна інженерія".  |

|  |   |
|--|---|
| <p><b>Література для вивчення дисципліни</b></p> | <p><i>Основна література</i></p> <ol style="list-style-type: none"> <li>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></li> <li>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></li> <li>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></li> <li>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></li> <li>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></li> <li>6. Troelsen Andrew, Japikse Phil. Pro C# 10 with .NET 6: Foundational Principles and Practices in Programming – APress, 2022 – 1640 pp.</li> <li>7. Christian Nagel Professional C# and .NET, 2021st Edition – Wrox, 2021 – 1800 pp.</li> </ol> <p><i>Додаткова література</i></p> <ol style="list-style-type: none"> <li>8. Mark J. Price C# 10 and .NET 6 – Modern Cross-Platform Development – Packt Publishing, 2021 – 826 pp.</li> <li>9. Joseph Albahari C# 10 in a Nutshell: The Definitive Reference – O’Reilly Media, 2022 – 1000 pp.</li> <li>10. Ian Griffiths C# 10 in a Nutshell – O’Reilly Media, 2022 – 833 pp.</li> <li>11. Rob Miles Exam Ref 70-483 Programming in C#, Second Edition – Microsoft Press, 2018 – 433 pp.</li> </ol> |
| <p><b>Обсяг курсу</b></p>                        | <p>4 кредити ЄКТС – 120 годин. З них 32 години лекцій, 32 години лабораторних занять та 56 годин самостійної роботи</p>   |
| <p><b>Очікувані результати навчання</b></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><b>Компетентності</b></p>                | <p><i>Загальні (ЗК):</i></p> <p>ЗК1. Здатність до абстрактного мислення, аналізу та синтезу, до застосування знань у практичних ситуаціях.</p> <p>ЗК2. Знання й розуміння предметної області та професійної діяльності.</p> <p>ЗК4. Здатність орієнтуватися в інформаційному просторі, здійснювати пошук, аналіз та обробку інформації з різних джерел, ефективно використовувати цифрові ресурси та технології в освітньому процесі.</p> <p>ЗК6. Здатність до міжособистісної взаємодії та роботи у команді у сфері професійної діяльності, спілкування з представниками інших професійних груп різного рівня.</p> <p>ФК4. Здатність формувати і розвивати в учнів ключові та предметні компетентності засобами навчального предмету та інтегрованого навчання; формувати в них ціннісне ставлення, розвивати критичне мислення.</p> <p>ПК2 Володіння методами інформаційного моделювання; здатність реалізовувати інформаційну модель засобами інформаційнокомунікаційних технологій; проводити комп'ютерний експеримент, інтерпретувати, аналізувати та узагальнювати його результати.</p> <p>ПК3. Здатність до використання сучасних методів розробки та дослідження алгоритмів розв'язування задач у моделюванні об'єктів і процесів та реалізації цих алгоритмів сучасними мовами програмування.</p> <p>ПК4. Здатність використовувати програмні засоби загального та спеціального призначення для розв'язання прикладних задач з інформатики</p> <p>ПК6. Здатність розв'язувати задачі шкільного курсу інформатики різного рівня складності, аналізувати та оцінювати ефективність розв'язку та формувати відповідні вміння в учнів.</p> |
| <p><b>Програмні результати навчання</b></p> | <p>ПРН7. Демонструє знання основ фундаментальних і прикладних наук інформатики та програмування, оперує базовими категоріями та поняттями предметної області спеціальності.</p> <p>ПРН13. Визначає структуру предметної галузі інформатики, її місце в системі наук, пояснює перспективи розвитку інформатики та інформаційних технологій, їхнє суспільне значення.</p> <p>ПРН14. Знає та розуміє фізичні, логічні та математичні основи інформаційних технологій; пояснює та застосовує способи двійкового кодування текстової, числової, графічної, звукової та відеоінформації.</p> <p>ПРН17. Визначає та застосовує методи розроблення та дослідження алгоритмів розв'язування задач з інформатики, описує і застосовує методи оцінювання ефективності алгоритмів.</p> <p>ПРН21. Уміє реалізувати алгоритми розв'язання задач мовами програмування, вибирати й застосовувати інформаційно-комунікаційні технології; розв'язує задачі шкільного курсу інформатики різного рівня складності.</p>  |

|                      |  |  |                    |                 |                              |
|----------------------|--|--|--------------------|-----------------|------------------------------|
| <b>Ключові слова</b> | Платформа .Net, мова програмування C#, клас, інтерфейс, властивість, подія, делегат, LINQ, серіалізація, багатопотоковість, Windows Forms. |  |                    |                 |                              |
| <b>Формат курсу</b>  | Очний.   |  |                    |                 |                              |
| <b>Теми</b>          |  |  |                    |                 |                              |
|                      | Тижд   | Тема, план, короткі тези   | Форма заняття      | Тривалість, год | Термін виконання             |
| 1                    |  | 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. | Лекція             | 2               |                              |
|                      |  | Побудова консольних програм в середовищі Visual Studio засобами мови C#. Форматоване виведення. Послідовні, галужені, циклічні алгоритми з використанням вбудованих типів.   | Лабораторна робота | 2               | Наступне лабораторне заняття |
|                      |  | Ознайомлення з загальною характеристикою платформи .Net  | Самостійна робота  | 3               |                              |
| 2                    |  | Оголошення класів і структур мовою C#: поля, методи, властивості. Передавання аргументів методам. Конструювання, копіювання об'єктів. Статичні члени класу, статичні класи. Методи, що розширяють клас.  | Лекція             | 2               |                              |
|                      |  | Опрацювання масивів, у тому числі зубчастих. Оголошення та використання переліків.   | Контрольна робота  | 2               |                              |
|                      |  | Побудова консольних застосунків зі складними циклічними алгоритмами. Використання масивів.   | Самостійна робота  | 4               |                              |
| 3                    |  | Операції мови C#: is, as, поглинання null та інші спеціальні. Перетворення типів. Перевірка рівності об'єктів. Перевантаження операцій. Методи-індексатори.  | Лекція             | 2               |                              |
|                      |  | Оголошення та використання структур. Оголошення та використання простих класів. Побудова та використання властивостей.   | Лабораторна робота | 2               | Наступне лабораторне заняття |
|                      |  | Оголошення та використання класів. Використання нових (порівняно з C++) можливостей класів.  | Самостійна робота  | 4               |                              |
| 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  | 3               |                              |
| 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  | 3               |                              |
| 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  | 4               |                              |
| 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 події.   | Самостійна робота  | 4               |                              |
| 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  | 3 |                              |
| 9                                       | Файлове введення-виведення і серіалізація об'єктів. Доступ до файлової системи. Маніпулювання файлами і директоріями. Потіки даних. Формати серіалізації: двійковий, XML, SOAP.  | Лекція             | 2 |                              |
|   | Використання інструментів рефлексії для програмного дослідження класів. Побудова та використання атрибутів користувача.  | Лабораторна робота | 2 | Наступне лабораторне заняття |
|   | Динамічне завантаження бібліотек на етапі виконання програми. Дослідження типів бібліотеки.  | Самостійна робота  | 3 |                              |
| 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  | 4 |                              |
| 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  | 4 |                              |
| 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  | 4 |                              |
| 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 потоці, синхронізація.  | Самостійна робота  | 3 |                              |
| 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  | 4 |                              |
| 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  | 3 |                              |
| 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  | 3 |                              |
| <b>Підсумковий контроль, форма</b>      | іспит в кінці семестру   |                    |   |                              |
| <b>Пререквізити</b>                     | Для вивчення курсу студенти потребують знань з попередніх семестрів навчання: основи ООП, використання винятків, взаємодія з потоками, використання контейнерів, модульне тестування. Одночасно з вивченням програмування студенти проходять навчальну обчислювальну практику, впродовж якої виконують завдання з програмування, поглиблюють набуті в курсі знання та удосконалюють навички. |                    |   |                              |
| <b>Навчальні методи та техніки, які</b> | Лекції з мультимедійними презентаціями та з демонстрацією прийомів практичного використання середовища програмування; лабораторні заняття у  |                    |   |                              |

|   |  |
|---|--|
| <p><b>використовують під час викладання курсу</b></p>                             | <p>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.</p>  |
| <p><b>Необхідне обладнання</b></p>  | <p>Для проведення лекцій: комп'ютер, проектор, доступ до мережі інтернет.<br/> Для проведення лабораторних та виконання завдань: комп'ютер, ОС Windows, доступ до інтернету, середовище програмування мовою C# (Microsoft Visual Studio).<br/> 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.</p>   |
| <p><b>Критерії оцінювання (окремо для кожного виду навчальної діяльності)</b></p> | <p><b>Оцінювання</b> проводиться за 100-бальною шкалою. 50 балів нараховують за виконання лабораторних завдань і контрольних робіт, ще 50 балів – за виконання екзаменаційного завдання. Лабораторні завдання можуть бути індивідуальні та командні. Упродовж семестру студент виконує не менше 10 лабораторних робіт, кожна з яких оцінюють 3-5 балів залежно від складності.<br/> Для кожного завдання визначено термін виконання: зазвичай до закінчення навчального тижня. Вчасно виконані завдання оцінюють так (у відсотках від максимальної оцінки):<br/> 100% – умови завдання виконано повністю, алгоритми складено правильно, програма містить належні коментарі, роботу програми перевірено на достатньому наборі тестових даних, автор відповідає на всі запитання щодо використаних підходів, чітко інтерпретує отримані результати, немає ознак недоброчесності;<br/> 80% – наведено логічно правильну послідовність розв'язування, алгоритми складено правильно, бракує окремих коментарів чи тестів, автор не досить повно пояснює використані підходи, немає ознак недоброчесності;<br/> 60% – у правильній послідовності розв'язування допущено окремі помилки, які автор уміє виправити після зауваження викладача, бракує коментарів чи тестів, на запитання щодо використаних підходів автор відповідає з помилками, немає ознак недоброчесності;<br/> 40% – у правильній послідовності розв'язування пропущено окремі етапи, завдання виконано частково, автор не розуміє недоліків поданої роботи, не вміє їх виправити, немає ознак недоброчесності;<br/> 20% – завдання виконано частково, немає тестів, програма працює правильно для окремих наборів вхідних даних, автор не може самостійно інтерпретувати отримані результати, виправити помилки, немає ознак недоброчесності;<br/> 0% – завдання не виконано, написана програма не відповідає умові, або ж виявлено ознаки недоброчесності: запозичення, фрагменти коду, дію яких автор пояснити не може, автор не володіє відповідним теоретичним матеріалом тощо; можуть бути нараховані додаткові бали за повністю виконане завдання, яке містить кілька способів розв'язування, використовує особливо ефективний спосіб, демонструє креативність автора тощо.<br/> Запізнення зменшує максимальну оцінку за завдання: кожного наступного після терміну виконання тижня оцінка зменшується удвічі.<br/> Оцінка за екзаменаційне завдання може бути поділена на дві частини: 20 балів за засвоєння теоретичного матеріалу, виставлені після опитувань упродовж семестру (у формі тестувань, колоквиумів тощо) та 30 за написання комп'ютерної програми (декількох програм). Завдання до курсу можуть передбачати самостійне опрацювання актуальних публікацій щодо програмування для .Net.<br/> <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> |

|                   |   |
|-------------------|---|
|                   | <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 для її незарахування викладачем, незалежно від масштабів плагіату чи обману. Жодні форми порушення академічної доброчесності не толеруються. |
| <b>Опитування</b> | Анкету-оцінку з метою оцінювання якості курсу буде надано після завершення курсу.   |

## Запитання до іспиту

1. Охарактеризуйте склад середовища .NET.
2. Засоби середовища .NET компіляції та виконання програм.
3. Система типів мови програмування C#.
4. Консольне введення мовою C#. Як вводити числові дані?
5. Консольне виведення мовою C#. Які засоби форматування ви знаєте?
6. Тип перелік мови C#. Приклад оголошення, використання, що пояснює можливості типу.
7. Тип перелік мови C#. Введення-виведення значень типу перелік.
8. Типи одновимірний масив, багатовимірний масив мови C#. Наведіть приклади оголошення, ініціалізації, використання.
9. Зубчастий масив мови C#. Поясніть структуру, можливості, наведіть приклад оголошення, ініціалізації, використання.
10. Оператор перебору послідовності мови C#. Приклад. Умови, за яких його можна використовувати.
11. Укажіть всі способи перебирання елементів одновимірного та двовимірного масивів у мові програмування C#.
12. Копіювання, впорядкування масивів у мові програмування C#.
13. Методи класу System.Object у мові C#.
14. Що таке кортеж у мові C#? Як його оголошують, використовують?
15. Охарактеризуйте тип структуру мови C#
16. З якою метою застосовують блок ітератора? Наведіть приклад використання, поясніть алгоритм його функціонування.
17. Охарактеризуйте члени-дані та члени-функції класу у мові C# (вичерпний перелік).
18. Охарактеризуйте модифікатори доступу класу у мові C#.
19. Охарактеризуйте модифікатори оголошення (не доступу) у мові C#.
20. Охарактеризуйте конструктори класу у мові C#. Як з одного конструктора викликати інший?
21. Що ви знаєте про параметри методів у мові C# (оголошення та використання)?
22. Охарактеризуйте властивості (property) класу у мові C#.
23. Що ви знаєте про автоматичні властивості у мові C#?
24. Статичні класи у мові C#: призначення оголошення, використання.
25. Розширення функціональності класу у мові C# (без доступу до самого класу).
26. Синтаксис наслідування класів у мові C#. Конструктори підкласу.
27. Особливості оголошення та використання абстрактного класу мовою C#.
28. Як оголошують та перевизначають віртуальні методи мовою C#?
29. Сумісність і приведення споріднених типів у мові C#. Перевірка типу.
30. Охарактеризуйте поняття інтерфейсу у мові C#. Наведіть приклад наслідування інтерфейсу класом мови C#.
31. Охарактеризуйте інтерфейси IEnumerable, IEnumerator мови C#.
32. Охарактеризуйте інтерфейси ICloneable, IComparable мови C#.
33. Типи, що допускають null мови C#. Операція поглинання нуля мови C#.
34. Перевірка рівності об'єктів мовою C#. Опишіть усі варіанти.
35. Як перевантажи операцію мовою C#? Наведіть приклад для одної операції.
36. Методи індексатори, методи перетворення типу мовою C#.

37. Охарактеризуйте тип делегата мови C#. Члени класу System.MulticastDelegate
38. Використання групових делегатів, анонімних делегатів, лямбда-виразів мовою C#
39. Охарактеризуйте поняття події мовою C#. Які кроки потрібно виконати, щоб оголосити подію у власному класі? Як її використати?
40. Наведіть приклад взаємодії об'єктів власних класів через механізм подій (наприклад, подія зміни властивості).
41. Які можливості є у мови C# для побудови контейнера елементів довільного типу?
42. До чого можна застосувати узагальнення в мові C#? Перелічіть переваги узагальнень.
43. Використання обмежень для оголошення узагальнених класів мовою C#.
44. З якою метою застосовують винятки? Які класи винятків мови програмування C# визнаєте?
45. Опишіть засоби мови C# для перехоплення та опрацювання винятків.
46. Огляд контейнерів простору імен System.Collections мови C#.
47. Огляд контейнерів простору імен System.Collections.Specialized мови C#.
48. Огляд контейнерів простору імен System.Collections.Generic мови C#.
49. Індексована колекція змінного розміру мови C#.
50. Контейнери черга, стек мови C#.
51. Контейнери списки мови C#.
52. Контейнери словники мови C#.
53. Огляд класів простору імен System.IO мови C#.
54. Опишіть, для чого і як використовують Класи Directory, DirectoryInfo мови C#.
55. Опишіть, для чого і як використовують Класи File, FileInfo мови C#.
56. Опишіть, як мовою C# вивести інформацію до текстового файла.
57. Опишіть, як мовою C# прочитати дані з текстового файла.
58. Опишіть, як мовою C# використовують двійкові файли, для чого.
59. Опишіть один зі способів серіалізації об'єктів мовою C#.
60. Що ви знаєте про клас GarbageCollector мови C#?
61. Як оголошують та як використовують метод-фіналізатор мовою C#? У яких випадках потрібні фіналізатори?
62. Охарактеризуйте інтерфейс IDisposable мови C#. Які класи наслідують його?
63. Опишіть шаблон використання фіналізатора в класі, що реалізує IDisposable мови C#.
64. Наведіть приклад «лінивого» створення об'єктів мовою C#.
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?
66. Які можливості класу Thread мови C# ви знаєте? Наведіть приклад використання.
67. З якою метою використовують клас C# BackgroundWorker? Опишіть його можливості та налаштування.
68. Які можливості класу Task мови C# ви знаєте? Наведіть приклад використання.
69. Як отримати доступ до типу довільного об'єкта C# програми? Поясніть на прикладах. Які властивості типу можна дослідити?
70. Як програмно отримати перелік методів, конструкторів довільного об'єкта C# програми? Наведіть приклад.
71. Як програмно отримати перелік полів, властивостей довільного об'єкта C# програми? Наведіть приклад.
72. Як програмно отримати доступ до C# збірки з метою дослідження? Які властивості збірки можна одразу ж перевірити?
73. Як програмно отримати колекцію типів, визначених у C# збірці?
74. Що таке атрибут програмного коду C#? Для чого їх використовують? Наведіть приклад
75. Як визначити власні атрибути програмного коду C#? Як їх використати?
76. Які види застосунків WPF ви знаєте? Охарактеризуйте стисло особливості кожного.
77. Охарактеризуйте призначення та можливості основних класів, які використовують для побудови WPF-застосунку.
78. Яку роль відіграє XAML для побудови WPF-застосунку? Назвіть основні синтаксичні елементи XAML, наведіть приклади.
79. Охарактеризуйте елементи керування WPF: кнопки, перемикачі, списки.
80. Охарактеризуйте елементи керування WPF: поля введення, таблиці, написи.

81. Охарактеризуйте елементи керування WPF: меню, інструментальні панелі.
82. Охарактеризуйте діалогові вікна WPF.
83. Охарактеризуйте LINQ to XML провайдер.
84. Охарактеризуйте типи XElement, XContainer, XDocument.
85. Як виконується побудова XML-дерев?