

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
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
«Прикладне програмування мовою Python»,
викладається в межах ОПП «Інформатика»
першого (бакалаврського) рівня вищої освіти
для здобувачів зі спеціальності 122 Комп'ютерні науки

Львів 2024 р.

Назва дисципліни	Прикладне програмування мовою Python
Адреса викладання дисципліни	Львівський національний університет імені Івана Франка, вул. Університетська 1, м. Львів, Україна, 79000
Факультет та кафедра, за якою закріплена дисципліна	Факультет прикладної математики та інформатики, кафедра програмування
Галузь знань, шифр та назва спеціальності	Галузь знань: 12 Інформаційні технології Спеціальність: 122 Комп'ютерні науки Спеціалізація: Інформатика
Викладачі дисципліни	Черняхівський Володимир Вікторович, к. ф.-м. н., доц., доцент кафедри програмування
Контактна інформація викладачів	Електронна пошта: volodymyr.chernyakhivskyy@lnu.edu.ua веб-сторінка: https://ami.lnu.edu.ua/employee/cherniakhivskyy
Консультації з питань навчання по дисципліні відбуваються	Консультації проводять раз на тиждень згідно з оприлюдненим розкладом консультацій викладача. Можливі онлайн консультації через Zoom чи Microsoft Teams. Для погодження часу онлайн консультацій потрібно писати на електронну пошту викладача.
Сторінка курсу	https://ami.lnu.edu.ua/course/trykladne-prohamuvannya-movoyu-python-informatyka
Інформація про дисципліну	Курс “Прикладне програмування мовою Python” є вибірковою навчальною дисципліною зі спеціальності 122 Комп'ютерні науки (інформатика) для освітньої програми «Інформатика», яку викладають у сьомому семестрі в обсязі 4 кредитів (за Європейською кредитно-трансферною системою ECTS)
Коротка анотація дисципліни	Вивчення спеціалізованих бібліотек і засобів мови Python, необхідних для оволодіння прийомами прикладного програмування, орієнтованими на застосування за фахом в майбутній роботі. Формування системи знань про методи, алгоритми і прийоми програмування типових прикладних задач в рамках розробки програмних проєктів. Формування практичних навичок складання фахових прикладних програм, структур даних, моделей за розділами, викладеними в переліку тем курсу. Курс використовує низку знань і навиків, які були отримані студентами в інших раніше викладених курсах за розділами програмування і суміжних з ними.
Мета та цілі дисципліни	Метою вибіркової дисципліни «Прикладне програмування мовою Python» є: <ul style="list-style-type: none"> ● вивчення спеціалізованих бібліотек і засобів мови Python, пов'язаних з прикладним програмуванням типових частин реальних проєктів; ● розуміння предмета дисципліни, зв'язку з іншими суміжними засобами і методами програмування, необхідними для цілей майбутньої розробки програмних проєктів; ● вміння застосовувати алгоритмічну мову Python в обсязі задач розділів курсу; ● вивчення сучасних методів і алгоритмів програмування прикладних задач; ● отримання навиків розробки процедур проектування, програмування, тестування і налагодження типових прикладних програм мовою Python; ● вміння використовувати стандартні і тематичні бібліотеки Python, середовище програмування Python.
Література для вивчення дисципліни	<i>Основна література</i> <ol style="list-style-type: none"> 1. Python Software Foundation. The Python Tutorial [Електронний ресурс]. – Режим доступу: https://docs.python.org/3/tutorial/index.html 2. Python Software Foundation. Python 3.7.12 documentation [Електронний ресурс]. – Режим доступу: https://docs.python.org/3.7/ 3. Lutz M. Learning Python, 5th Edition. – O'Reilly Media Inc., 2013. – 1648 p. 4. Chun Wesley J. Core Python Application Programming. Third Edition. – Pearson Education, Inc., 2012. 5. Lutz M. Programming Python, Forth Edition. – O'Reilly Media Inc., 2011.

	<p>6. Prometheus: CS50. Вебпрограмування з Python та JavaScript CS50. – Prometheus. – 2021. [Електронний ресурс] – Режим доступу: https://courses.prometheus.org.ua/courses/course-v1:Prometheus+CS50+2021_T1/about</p> <p>7. Список структур даних. Матеріал з Вікіпедії — вільної енциклопедії. [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Список_структур_даних</p> <p>8. Перелік файлових форматів. Матеріал з Вікіпедії — вільної енциклопедії. [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Перелік_файлових_форматів</p> <p><i>Додаткова література</i></p> <p>9. Microsoft Ignite. Visual Studio documentation [Електронний ресурс]. – Режим доступу: https://docs.microsoft.com/en-us/visualstudio/windows/?f1url=%3FappId%3DDev15IDEF1%261%3Den-US%26k%3Dk(MSDNSTART)%26rd%3Dtrue&view=vs-2019</p> <p>10. Томас Г. Кормен, Чарлз Е. Лейзерсон, Роналд Л. Рівест, Кліфорд Стайн. Вступ до алгоритмів. — К. : К. І. С., 2019. — 1288 с. ISBN 978-617-684-239-2 [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Вступ_до_алгоритмів#Український_переклад</p>
Обсяг курсу	4 кредити ЄКТС – 120 годин. З них 16 годин лекцій, 32 годин лабораторних занять та 72 години самостійної роботи
Очікувані результати навчання	<p>Після завершення цього курсу студент буде:</p> <p><i>знати</i></p> <ul style="list-style-type: none"> ● сучасні методи і алгоритми програмування прикладних задач мовою Python; ● засоби спеціалізованих бібліотек мови Python, пов'язані з прикладним програмуванням типових частин реальних проєктів; ● процедури моделювання і застосування структур даних мовою Python, необхідних для реальних проєктів; ● методи організації виконання сценаріїв (Python-програм); ● технології роботи з internet-файлами, формати файлів; ● архітектурні особливості прикладних програмних проєктів мовою Python; <p><i>вміти</i></p> <ul style="list-style-type: none"> ● застосовувати засоби мови Python і спеціалізованих бібліотек для програмування типових прикладних задач; ● складати, тестувати і налагоджувати прикладні функції частин реальних проєктів мовою Python; ● реалізувати вимоги замовників до розробки програмних проєктів мовою Python; ● аналізувати і модифікувати прикладні програми до зміни вимог.
Компетентності	<p><i>Інтегральна:</i> Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми у галузі інформаційних технологій. Здатність застосовувати загальні принципи програмування задач на системному рівні мовою Python. Здатність застосовувати сучасні засоби програмування мовою Python задач різного призначення, проектувати прикладні програми, структури даних і алгоритмічні моделі за тематикою реальних програмних проєктів.</p> <p><i>Загальні (ЗК):</i></p> <p>ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.</p> <p>ЗК2. Здатність застосовувати знання у практичних ситуаціях.</p> <p>ЗК3. Знання та розуміння предметної області та розуміння професійної діяльності.</p> <p>ЗК7. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.</p> <p>ЗК8. Здатність генерувати нові ідеї (креативність).</p> <p>ЗК9. Здатність працювати в команді.</p> <p>ЗК11. Здатність приймати обґрунтовані рішення.</p> <p>ЗК12. Здатність оцінювати та забезпечувати якість виконуваних робіт.</p> <p><i>Спеціальні (фахові, предметні) компетентності (СК):</i></p> <p>СК3. Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та</p>

	<p>складності, розв'язності та нерозв'язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем.</p> <p>СК7. Здатність застосовувати теоретичні та практичні основи методології та технології моделювання для дослідження характеристик і поведінки складних об'єктів і систем, проводити обчислювальні експерименти з обробкою й аналізом результатів.</p> <p>СК8. Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.</p> <p>СК10. Здатність застосовувати методології, технології та інструментальні засоби для управління процесами життєвого циклу інформаційних і програмних систем, продуктів і сервісів інформаційних технологій відповідно до вимог замовника.</p> <p>СК12. Здатність забезпечити організацію обчислювальних процесів в інформаційних системах різного призначення з урахуванням архітектури, конфігурування, показників результативності функціонування операційних систем і системного програмного забезпечення.</p>																																			
<p>Програмні результати навчання</p>	<p>ПР1. Застосовувати знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу, обробки та синтезу інформації в предметній області комп'ютерних наук.</p> <p>ПР5. Проектувати, розробляти та аналізувати алгоритми розв'язання обчислювальних та логічних задач, оцінювати ефективність та складність алгоритмів на основі застосування формальних моделей алгоритмів та обчислюваних функцій.</p> <p>ПР9. Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук.</p> <p>ПР11. Володіти навичками управління життєвим циклом програмного забезпечення, продуктів і сервісів інформаційних технологій відповідно до вимог і обмежень замовника, вміти розробляти проектну документацію (техніко-економічне обґрунтування, технічне завдання, бізнес-план, угоду, договір, контракт).</p>																																			
<p>Ключові слова</p>	<p>Python, синтаксис, семантика, оператор, сценарій, структури даних, словник, стек, черга, файл, каталог, потік, процес, інтернет, сервер, веб-сторінка, формат xml, формат json, python+json, програмний проєкт, архітектура програмного проєкта, парадигма програмування, технологія програмування, документація програмного продукта, сервісна функція програми.</p>																																			
<p>Формат курсу</p>	<p>Очний: проведення лекцій, лабораторних робіт та консультацій в приміщеннях університету, а в умовах форсмажорних обставин – онлайн-овий на платформі Microsoft Teams.</p>																																			
<p>Теми</p>	<table border="1"> <thead> <tr> <th data-bbox="491 1630 571 1675">Тиж-день</th> <th data-bbox="571 1630 1098 1675">Тема, план, короткі тези</th> <th data-bbox="1098 1630 1249 1675">Форма заняття</th> <th data-bbox="1249 1630 1353 1675">Тривалість год</th> <th data-bbox="1353 1630 1528 1675">Термін виконання</th> </tr> </thead> <tbody> <tr> <td data-bbox="491 1675 571 1753">1</td> <td data-bbox="571 1675 1098 1753">Середовища розробки python-програм. Базове середовище IDLE. Параметри середовищ для розробки програм. Взаємодія з операційною системою.</td> <td data-bbox="1098 1675 1249 1753">Лекція</td> <td data-bbox="1249 1675 1353 1753">2</td> <td data-bbox="1353 1675 1528 1753"></td> </tr> <tr> <td data-bbox="491 1753 571 1832"></td> <td data-bbox="571 1753 1098 1832">Інструменти середовища для розробки програм. Операції редагування, виконання, перегляду, контролю.</td> <td data-bbox="1098 1753 1249 1832">Лабораторна робота</td> <td data-bbox="1249 1753 1353 1832">2</td> <td data-bbox="1353 1753 1528 1832">Наступне лабораторне заняття</td> </tr> <tr> <td data-bbox="491 1832 571 1910">2</td> <td data-bbox="571 1832 1098 1910">Базові типи даних мови Python. Принципи будови системи даних. Змінні величини, об'єкти, посилання. Поліморфізм операцій. Типові бібліотеки функцій.</td> <td data-bbox="1098 1832 1249 1910">Лабораторна робота</td> <td data-bbox="1249 1832 1353 1910">2</td> <td data-bbox="1353 1832 1528 1910">Наступне лабораторне заняття</td> </tr> <tr> <td data-bbox="491 1910 571 1955">3</td> <td data-bbox="571 1910 1098 1955">Алгоритми програмування задач типового змісту. Використання типів Python і засобів бібліотек.</td> <td data-bbox="1098 1910 1249 1955">Лекція</td> <td data-bbox="1249 1910 1353 1955">2</td> <td data-bbox="1353 1910 1528 1955"></td> </tr> <tr> <td data-bbox="491 1955 571 2033"></td> <td data-bbox="571 1955 1098 2033">Числові задачі. Особливості числових типів. Методи налагодження і тестування.</td> <td data-bbox="1098 1955 1249 2033">Лабораторна робота</td> <td data-bbox="1249 1955 1353 2033">2</td> <td data-bbox="1353 1955 1528 2033">Наступне лабораторне заняття</td> </tr> <tr> <td data-bbox="491 2033 571 2098">4</td> <td data-bbox="571 2033 1098 2098">Задачі опрацювання текстових даних і даних спеціальних форматів. Системи кодування літер.</td> <td data-bbox="1098 2033 1249 2098">Лабораторна робота</td> <td data-bbox="1249 2033 1353 2098">2</td> <td data-bbox="1353 2033 1528 2098">Наступне лабораторне заняття</td> </tr> </tbody> </table>	Тиж-день	Тема, план, короткі тези	Форма заняття	Тривалість год	Термін виконання	1	Середовища розробки python-програм. Базове середовище IDLE. Параметри середовищ для розробки програм. Взаємодія з операційною системою.	Лекція	2			Інструменти середовища для розробки програм. Операції редагування, виконання, перегляду, контролю.	Лабораторна робота	2	Наступне лабораторне заняття	2	Базові типи даних мови Python. Принципи будови системи даних. Змінні величини, об'єкти, посилання. Поліморфізм операцій. Типові бібліотеки функцій.	Лабораторна робота	2	Наступне лабораторне заняття	3	Алгоритми програмування задач типового змісту. Використання типів Python і засобів бібліотек.	Лекція	2			Числові задачі. Особливості числових типів. Методи налагодження і тестування.	Лабораторна робота	2	Наступне лабораторне заняття	4	Задачі опрацювання текстових даних і даних спеціальних форматів. Системи кодування літер.	Лабораторна робота	2	Наступне лабораторне заняття
Тиж-день	Тема, план, короткі тези	Форма заняття	Тривалість год	Термін виконання																																
1	Середовища розробки python-програм. Базове середовище IDLE. Параметри середовищ для розробки програм. Взаємодія з операційною системою.	Лекція	2																																	
	Інструменти середовища для розробки програм. Операції редагування, виконання, перегляду, контролю.	Лабораторна робота	2	Наступне лабораторне заняття																																
2	Базові типи даних мови Python. Принципи будови системи даних. Змінні величини, об'єкти, посилання. Поліморфізм операцій. Типові бібліотеки функцій.	Лабораторна робота	2	Наступне лабораторне заняття																																
3	Алгоритми програмування задач типового змісту. Використання типів Python і засобів бібліотек.	Лекція	2																																	
	Числові задачі. Особливості числових типів. Методи налагодження і тестування.	Лабораторна робота	2	Наступне лабораторне заняття																																
4	Задачі опрацювання текстових даних і даних спеціальних форматів. Системи кодування літер.	Лабораторна робота	2	Наступне лабораторне заняття																																

	5	Вбудовані структури даних мови Python. Словники. Базові операції над словниками. Приклади задач з використанням словників.	Лекція	2	
		Розробка макета проєкта опрацювання даних прогнозу погоди. Джерела даних про погоду. Формати вхідних даних, функції опрацювання, зображення результатів.	Лабораторна робота	2	Наступне лабораторне заняття
	6	Тестові файли вхідних даних задач прогнозу погоди. Програмна реалізація задач. Правила будови інструкції користувача.	Лабораторна робота	2	Наступне лабораторне заняття
	7	Типові архітектурні особливості будови інформаційно-довідкових систем. Задача будови системи про пересування містом трамваями у Львові.	Лекція	2	
		Програмне укладання трамвайних маршрутів. Вибір і заповнення структур даних. Форматування даних для функцій опрацювання. Формати запитів і відповідей.	Лабораторна робота	2	Наступне лабораторне заняття
	8	Програмування функцій опрацювання запитів і будови відповідей. Сервісні внутрішні функції проєкта. Протоколи виконання програми.	Лабораторна робота	2	Наступне лабораторне заняття
	9	Модельовані структури даних Python. Стек, функціонування, програмне моделювання. Черга, функціонування, програмне моделювання.	Лекція	2	
		Алгоритми і програмування задач з використанням структури стека і функцій стека.	Лабораторна робота	2	Наступне лабораторне заняття
	10	Програмна модель структури черги. Програмування задачі моделювання організації прикордонного митного контролю. Структури даних для задачі. Операції. Протоколи виконання.	Лабораторна робота	2	Наступне лабораторне заняття
	11	Робота з файлами і каталогами. Режими відкриття файлу. Сканери файлів. Модулі та інструменти для роботи з файлами і каталогами. Типові задачі.	Лекція	2	
		Задачі пошуку файлів за різними критеріями. Відображення характеристик файлу. Задачі операцій з файлами і каталогами. Інструменти розробника.	Лабораторна робота	2	Наступне лабораторне заняття
	12	Програмування сценаріїв виконання python-програми. Запуск файлу зі зв'язаною аплікацією. Передача керування іншій програмі. Дочірні процеси.	Лабораторна робота	2	Наступне лабораторне заняття
	13	Робота з internet-файлами. Доступ до веб-сайтів. Програмування сценаріїв веб-клієнтів. Доступ до файлів на веб-серверах. Опрацювання xml-файлів.	Лекція	2	
		Пошук url-адресів серверів і веб-файлів. Програмування сценаріїв веб-клієнтів. Опрацювання xml-файлів.	Лабораторна робота	2	Наступне лабораторне заняття
	14	Формат даних Json. Сервери і ресурси формату Json. Перетворення json-файлів в python-структури. Задачі опрацювання даних формату Json.	Лабораторна робота	2	Наступне лабораторне заняття
	15	Модель проєкта на основі Json-даних веб-серверів. Постановка задачі. Отримання json-даних. Дослідження структури json-файла. Приклади задач опрацювання json-даних. Архітектура проєкта.	Лекція	2	
	Планування архітектури проєкта на основі Json-даних серверів. Розробка головного сценарію мовою Python.	Лабораторна робота	2	Наступне лабораторне заняття	
16	Програмування, тестування і налагодження задач проєкта (C#, Python тощо). Комплексна перевірка цілого проєкта. Документування для користувача.	Лабораторна робота	2	До початку сесії	
Підсумковий контроль, форма	Залік в кінці семестру				
Пререквізити	Для вивчення курсу студенти потребують базових знань з дисциплін: програмування, бази даних та інформаційні системи, паралельні та розподілені обчислення, програмування та підтримка веб-застосовань, алгоритми і структури даних, програмна інженерія.				
Навчальні методи та техніки, які використовують під час викладання курсу	Лекції з мультимедійними презентаціями та з демонстрацією прийомів прикладного програмування мовою Python; лабораторні заняття у вигляді проєктування алгоритмів і програм, складання моделей реальних проєктів, виконання практичних завдань та проєктів; самостійне опрацювання навчальних матеріалів: підручників, конспектів лекцій, електронних ресурсів, готових програм, додаткових матеріалів, розміщених у хмарному сховищі (Microsoft Teams, Google Диск). Обговорення теоретичного та практичного матеріалу в онлайн сервісах, формулювання творчих завдань для студентів, виконання яких готує до вивчення нового теоретичного і практичного матеріалу.				
Необхідне обладнання	Для проведення лекцій: комп'ютер, проєктор, доступ до мережі інтернет.				

	<p>Для проведення лабораторних та виконання завдань: комп'ютер; ОС Windows/Linux; доступ до інтернету; середовища програмування мовою Python (IDLE Python тощо), мовою С#. Уся література і робочі матеріали, які 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>Критерії оцінювання (окремо для кожного виду навчальної діяльності)</p>	<p>Політика оцінювання:</p> <ul style="list-style-type: none"> • повнота виконання завдань відповідно до теми; • правильне оформлення звітів за виконання завдань і проєктів; • дотримання графіка виконання робіт; • допускається можливість доопрацювання завдань після першого оцінювання і повторна здача; • заохочується використання методик і програмних інструментів реальних проєктів; • використання до виконання завдань елементів фахової організації роботи. <p>Критерії оцінювання Оцінювання проводиться за 100-бальною шкалою. Упродовж семестру студент виконує 10 завдань, які оцінюють різною шкалою залежно від складності (конкретна шкала оцінювання і критерії надаються студентам на початку семестру). До роботи студента, виконання і оцінювання завдань застосовуємо комплексний підхід: кожне завдання є одночасно лабораторною роботою, домашнім завданням на самостійну роботу і проєктом. Основна частина завдань є індивідуальними. Окремі завдання з навчальною метою студент за бажанням може виконати в складі міні-команди, в цьому разі повідомляють додаткові умови виконання роботи. Для кожного завдання визначено термін виконання – зазвичай до наступного лабораторного заняття. Вчасно виконані завдання поточного контролю оцінюють так (у відсотках від максимальної оцінки):</p> <ul style="list-style-type: none"> • 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; • 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; • 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; • 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; • 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; • 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. <p>Запізнення терміну виконання завдання зменшує максимальну оцінку за завдання, ступінь зменшення визначає викладач до кожної роботи.</p>

	<p>Відвідання занять є важливою складовою навчання. Очікується, що всі студенти відвідають усі лекції і лабораторні заняття курсу. Активність під час проведення лекцій і лабораторних заохочується балами при оцінюванні відповідного лабораторного завдання чи проекту. У будь-якому випадку студенти зобов'язані дотримуватися усіх строків визначених для виконання усіх видів 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.</p> <p>Академічна доброчесність. 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.</p>
<p>Опитування</p>	<p>Анкету-оцінку з метою оцінювання якості курсу буде надано після завершення курсу.</p>