

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ

КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ

«ЗАТВЕРДЖУЮ»

Заступник декана
з навчальної роботи

_____ Кашпур О.Ф.

«__» _____ 2018 року

**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
МЕТАПРОГРАМУВАННЯ**

для студентів

галузь знань **12 «Інформаційні технології»**
спеціальність **121 «Інженерія програмного забезпечення»**
освітній рівень **бакалавр**
освітня програма **«Програмна інженерія»**

вид дисципліни вільного вибору студента

Форма навчання	денна
Навчальний рік	2018/2019
Семестр	7
Кількість кредитів ECTS	4
Мова викладання, навчання та оцінювання	українська
Форма заключного контролю	екзамен

Викладач: **к. ф.-м. н., асистент Терлецький Д.О.** (лекції, лабораторні заняття)

Пролонговано: на 20__/20__ н. р. _____ (_____) «__» 20__ р.
(підпис, ПІБ, дата)

на 20__/20__ н. р. _____ (_____) «__» 20__ р.
(підпис, ПІБ, дата)

КИЇВ – 2018

Розробник: Терлецький Дмитро Олександрович, к. ф.-м. н., асистент кафедри інформаційних систем.

ЗАТВЕРДЖЕНО

В. о. Зав. кафедри інформаційних систем

_____ (Іванов Є.О.)
(підпис) (прізвище та ініціали)

Протокол № 8 від «22» травня 2018 р.

Схвалено науково-методичною комісією факультету комп'ютерних наук та кібернетики

Протокол від «18» червня 2018 року №__

Голова науково-методичної комісії _____ (Хусаїнов Д.Я.)
(підпис) (прізвище та ініціали)

« ____ » _____ 2018 року

1. Мета дисципліни

Метою дисципліни «Метапрограмування» – є вивчення базових концепцій та механізмів динамічного метапрограмування в рамках мови програмування Python 3 з метою проектування та розробки програмного забезпечення для динамічного аналізу, модифікації та генерації програмних кодів.

2. Попередні вимоги до опанування або вибору навчальної дисципліни

Для успішного вивчення дисципліни «Метапрограмування» студент повинен відповідати наступним вимогам:

1. Успішне опанування курсів:
 1. Дискретна математика.
 2. Основи програмування.
 3. Програмування.
 4. Основи об'єктно-орієнтованого програмування.
 5. Об'єктно-орієнтоване програмування.
 6. Системне програмування та операційні системи.
2. Знання:
 1. Основних концепцій процедурного та об'єктно-орієнтованого програмування.
 2. Процедурного та об'єктно-орієнтованого програмування мовою C++.
 3. Об'єктно-орієнтованого програмування мовою Java.
 4. Базових шаблонів проектування програмного забезпечення.
 5. Основних понять теорії компіляторів.
 6. Предметно-орієнтованих мов SQL, HTML, XHTML, DHTML, XML.
3. Вміти:
 1. Програмувати у процедурному та об'єктно-орієнтованому стилях використовуючи мову C++.
 2. Програмувати у об'єктно-орієнтованому стилі використовуючи мову Java.
 3. Проектувати класові ієрархії на мовах C++ та Java.
 4. Реалізовувати мовах C++ та Java та адаптувати під власні потреби базові шаблони проектування програмного забезпечення.
 5. Використовуючи мови C++ та Java, розробляти програмне забезпечення для лексичного аналізу програм на різних мовах програмування.
 6. Використовуючи мови C++ та Java, розробляти програмне забезпечення для примітивного синтаксичного та структурного аналізу програм на різних мовах програмування.
4. Володіти:
 1. Базовими навичками використання інтегрованих середовищ розробки програмного забезпечення CLion, IntelliJIDEA, Eclipse, NetBeans.
 2. Англійською мовою на рівні не нижче Intermediate.

3. Анотація навчальної дисципліни

Навчальна дисципліна «Метапрограмування» є складовою освітньо-професійної програми підготовки фахівців за першим (бакалаврським) рівнем вищої освіти у галузі знань 12 «Інформаційні технології» зі спеціальності 121 «Інженерія програмного забезпечення», освітньо-професійної програми «Програмна інженерія»

Дана дисципліна належить до переліку дисциплін вільного вибору студента, спеціалізація «Програмна інженерія». Викладається у 1 семестрі 4 курсу в обсязі – 120 год.,

(4 кредити ECTS) зокрема: лекції – 28 год., лабораторні – 26 год., консультації – 2 год., самостійна робота – 64 год. У курсі передбачено 2 змістових модулі, 2 модульні контрольні роботи та 2 лабораторні роботи. Завершується дисципліна – **екзаменом**.

В результаті вивчення навчальної дисципліни студент повинен:

Знати:

1. Основи програмування мовою Python 3.
2. Основні поняття та концепції динамічного метапрограмування.
3. Механізми метапрограмування в рамках мови Python 3.

Вміти:

1. Проектувати та розробляти на мові Python 3 програмне забезпечення для динамічного аналізу, модифікації та генерації програмних кодів.

Дисципліна «Метапрограмування» є базовою для засвоєння дисциплін «Типи у мовах програмування», «Моделе-орієнтована побудова програмних систем», «Актуальні проблеми об'єктно-орієнтованого програмування», «Проектування мультиагентних систем» другого (магістерського) рівня вищої освіти у галузі знань 12 «Інформаційні технології» зі спеціальності 121 «Інженерія програмного забезпечення», освітньо-професійної програми «Програмне забезпечення систем».

4. Завдання (навчальні цілі)

Основними завданнями дисципліни «Метапрограмування» є набуття знань, умінь та навичок (компетентностей) відповідно до кваліфікації фахівець з інформаційних технологій. Зокрема, розвивати:

- Здатність до абстрактного мислення, аналізу та синтезу.
- Здатність застосовувати знання у практичних ситуаціях.
- Здатність до пошуку, оброблення та аналізу інформації з різних джерел.
- Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.
- Здатність розробляти архітектури, модулі та компоненти програмних систем.
- Здатність до алгоритмічного та логічного мислення.

5. Результати навчання за дисципліною

Результат навчання (1. знати; 2. вміти; 3. комунікація; 4. автономність та відповідальність)		Форми (та/або методи і технології) викладання і навчання	Методи оцінювання та пороговий критерій оцінювання (за необхідності)	Відсоток у підсумковій оцінці з дисципліни
Код	Результат навчання			
PH1.1	Знати мову програмування Python 3 (базовий рівень).	Лекції, лабораторні заняття, самостійна робота	Модульна контрольна робота 1, екзамен	10%
PH1.2	Знати основні поняття та концепції динамічного метапрограмування.	Лекції, лабораторні заняття, самостійна робота	Модульна контрольна робота 1 та 2, екзамен	5%

PH1.3	Знати механізми метапрограмування в рамках мови Python 3.	Лекції, лабораторні заняття, самостійна робота	Модульна контрольна робота 2, екзамен	35%
PH2.1	Вміти проектувати та розробляти на мові Python 3 програмне забезпечення для динамічного аналізу, модифікації та генерації програмних кодів.	Лабораторні заняття, самостійна робота	Здача лабораторних робіт, екзамен	20%
PH3.1	Консультуватися з викладачем стосовно питань що виникають у ході вивчення теоретичного матеріалу.	Лекції, лабораторні заняття, години консультацій	Поточне оцінювання, здача лабораторних робіт	5%
PH3.2	Обговорювати з колегами та викладачем проблеми питання що виникають у ході виконання лабораторних робіт.	Лабораторні заняття	Здача лабораторних робіт	5%
PH3.3	Чітко та послідовно обґрунтовувати власні архітектурні та програмні рішення в рамках виконання лабораторних робіт.	Лабораторні заняття	Здача лабораторних робіт	5%
PH4.1	Закріплення та поглиблення набутих на лекціях теоретичних знань проектування та розробки метапрограм.	Самостійна робота, опрацювання рекомендованих інформаційних джерел	Поточне оцінювання, здача лабораторних робіт	5%
PH4.2	Закріплення та поглиблення набутих на лабораторних заняттях практичних навичок проектування та розробки метапрограм.	Самостійна робота, реалізація рекомендованих завдань	Поточне оцінювання, здача лабораторних робіт	5%
PH4.3	Сумлінне і вчасне виконання та здача лабораторних робіт.	Лабораторні заняття	Здача лабораторних робіт	5%

6. Співвідношення результатів навчання дисципліни із програмними результатами навчання

Результати навчання дисципліни Програмні результати навчання	PH 1.1	PH 1.2	PH 1.3	PH 2.1	PH 3.1	PH 3.2	PH 3.3	PH 4.1	PH 4.2	PH 4.3
ПР-5. Знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізів та математичного моделювання для розробки програмного забезпечення.	+		+	+	+	+	+			
ПР-7. Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи	+		+	+	+	+				

функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.										
ПР-10. Проводити передпроектне обстеження предметної області, системний аналіз об'єкта проектування.				+	+	+		+	+	
ПР-12. Знати ефективні підходи щодо проектування програмного забезпечення.			+	+					+	+
ПР-13. Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.	+		+	+						
ПР-23. Уміння документувати та презентувати результати розробки програмного забезпечення.				+					+	+
ВПР-3. Знати та мати навички реалізації основних алгоритмів та структур даних програмування.	+		+	+						
ВПР-4. Знати та вміти застосовувати сучасні технології та методи проектування та програмування.		+	+	+				+	+	

7. Схема формування оцінки

7.1 Форми оцінювання студентів

Семестрове оцінювання:

1. Контрольна робота 1: РН 1.1, РН 1.2, РН 3.1, РН 4.1, РН 4.2 – 15 балів/9 балів.
2. Контрольна робота 2: РН1.2, РН 1.3, РН 3.1, РН 4.1, РН 4.2 – 15 балів/9 балів.
3. Лабораторна робота 1: РН1.1, РН 1.2, РН 2.1, РН3.1, РН3.2, РН3.3, РН 4.1, РН 4.2, РН 4.3 – 10 балів/6 балів.
4. Лабораторна робота 2: РН1.2, РН1.3, РН 2.1, РН3.1, РН3.2, РН3.3, РН 4.1, РН 4.2, РН 4.3 – 20 балів/12 балів.

Підсумкове оцінювання (у формі екзамену):

- максимальна кількість балів які можуть бути отримані студентом: 40 балів;
- результати навчання які будуть оцінюватись: РН1.1, РН1.2, РН1.3, РН2.1;
- форма проведення і види завдань: письмова;
- види завдань: 5 письмових завдань (2 теоретичних питання та 3 практичних завдання);
- для отримання загальної позитивної оцінки з дисципліни оцінка за екзамен повинна бути не меншою ніж 24 бали;
- студент не допускається до екзамену, якщо протягом семестру він набрав менше ніж 36 балів;
- студент не допускається до екзамену, якщо протягом семестру він не виконав та не здав 100 % лабораторних робіт передбачених планом.

Критерії оцінювання на екзамені

Завдання	Тема завдання	Максимальний відсоток від 40 балів	Всього відсотків
Завдання 1	Питання по теоретичному матеріалу курсу	15%	15%
Завдання 2		22,5%	22,5%
Завдання 3	Практичне завдання на основі теоретичного матеріалу курсу	25%	25%
Завдання 4		20%	20%
Завдання 5		17,5%	17,5%
			100%

Запитання для підготовки до екзамену

1. Базові поняття метапрограмування

1. Основні поняття та приклади застосування метапрограмування.
2. Різновиди метапрограмування та їх особливості. Приклади.
3. Метадані та метазнання. Приклади.
4. Ієрархія метарівнів. Приклади.

2. Керовані атрибути у мові Python 3

1. Поняття, призначення, різновиди та приклади властивостей класів.
2. Поняття, призначення, різновиди та приклади дескрипторів.
3. Поняття, призначення, різновиди та приклади спеціальних методів управління атрибутами класів.

3. Декоратори функцій у мові Python 3

1. Поняття декораторів функцій та їх застосування для делегування викликів декорованих функцій. Приклади.
2. Поняття декораторів функцій та їх застосування для додаткової обробки результатів викликів декорованих функцій. Приклади.
3. Поняття декораторів функцій та їх застосування для трасування та фіксації викликів декорованих функцій. Приклади.
4. Поняття декораторів функцій та їх застосування для делегування викликів декорованих методів класів. Приклади.
5. Поняття декораторів функцій та їх застосування для додаткової обробки результатів викликів декорованих методів класів. Приклади.
6. Поняття декораторів функцій та їх застосування для трасування та фіксації викликів декорованих методів класів. Приклади.
7. Поняття вкладених декораторів функцій та їх застосування для делегування викликів декорованих функцій. Приклади.
8. Поняття вкладених декораторів функцій та їх застосування для додаткової обробки результатів викликів декорованих функцій. Приклади.
9. Поняття вкладених декораторів функцій та їх застосування для делегування викликів декорованих методів класів. Приклади.

10. Поняття вкладених декораторів функцій та їх застосування для додаткової обробки результатів викликів декорованих методів класів. Приклади.

4. Декоратори класів у мові Python 3

1. Поняття декораторів класів та їх застосування для делегування викликів декорованих класів. Приклади.
2. Поняття декораторів класів та їх застосування для додаткової обробки результатів викликів декорованих класів. Приклади.
3. Поняття декораторів класів та їх застосування для модифікації декорованих класів. Приклади.
4. Поняття декораторів класів та їх застосування для управління створенням екземплярів декорованих класів. Приклади.
5. Поняття вкладених декораторів класів та їх застосування для делегування викликів декорованих класів. Приклади.
6. Поняття вкладених декораторів класів та їх застосування для додаткової обробки результатів викликів декорованих класів. Приклади.
7. Поняття вкладених декораторів класів та їх застосування для модифікації декорованих класів. Приклади.

5. Метакласи в мові Python 3

1. Поняття метакласів та їх застосування для модифікації класів. Приклади.
2. Поняття метакласів та їх застосування для управління створенням екземплярів класів. Приклади.
3. Поняття транзитивних метакласів та їх застосування для модифікації класів. Приклади.
4. Поняття метакласів на основі функцій та їх застосування для модифікації класів. Приклади.

7.2 Організація оцінювання

Терміни проведення форм оцінювання:

1. Контрольна робота 1: до 7 тижня семестру.
2. Контрольна робота 2: до 14 тижня семестру.
3. Лабораторна робота 1: до 7 тижня семестру.
4. Лабораторна робота 2: до 14 тижня семестру.

Студент має право на одне перескладання кожної контрольної роботи у визначений викладачем термін.

У випадку відсутності студента з поважних причин відпрацювання та перездачі контрольних робіт здійснюються у відповідності до «Положення про порядок оцінювання знань студентів при кредитно-модульній системі організації навчального процесу» від 1 жовтня 2010 року.

Студент має право здавати лабораторні роботи протягом усього навчального семестру.

7.3 Шкала відповідності оцінок

Відмінно / Excellent	90-100
Добре / Good	75-89
Задовільно / Satisfactory	60-74
Незадовільно / Fail	0-59
Зараховано / Passed	60-100
Не зараховано / Fail	0-59

8. Структура навчальної дисципліни. Тематичний план лекцій і лабораторних занять

№ лекції	Назва лекції	Кількість годин		
		Лекції	Лабораторні заняття	Самостійна робота
Частина 1. Програмування мовою Python 3				
1	Тема 1. Вступ до курсу. Загальні особливості синтаксису. Базові оператори. Умовні оператори. Циклічні оператори.	2		4
2	Тема 2. Базові типи даних. Числа. Рядки. Списки. Словники. Кортежі. Файли. Множини.	4		8
3	Тема 3. Функції. Області видимості та простори імен. Аргументи функцій. Розширені можливості функцій.	2	2	8
4	Тема 4. Ітератори. Генератори. Функції-генератори. Вирази-генератори. Документування. Модулі.	4	4	8
5	Тема 5. Класи. Інкапсуляція. Поліморфізм. Успадкування. Класові ієрархії.	2	4	6
	Контрольна робота 1			
Всього по частині 1		14	10	34
Частина 2. Метапрограмування мовою Python 3				
6	Тема 6. Основні поняття та різновиди метапрограмування. Ієрархія метарівнів. Метадані та метазнання.	2		4
7	Тема 7. Керовані атрибути. Властивості. Дескриптори. Спеціальні методи.	4	6	10
8	Тема 8. Декоратори функцій та класів.	4	6	8
9	Тема 9. Метакласи. Метакласи на основі класів. Метакласи на основі функцій.	4	4	8
	Контрольна робота 2			
Всього по частині 2		14	16	30
	Консультація		2	
	Екзамен			
ВСЬОГО		28	26	64

Загальний обсяг – 120 год., в тому числі:

Лекцій – **28 год.**

Лабораторні заняття – **26 год.**

Консультації – **2 год.**

Самостійна робота – **64 год.**

Лабораторні роботи:

Лабораторна робота 1: Примітивний аналіз, модифікація та генерація програмних кодів.

Лабораторна робота 2: Структурний аналіз, модифікація та генерація програмних кодів.

9. Рекомендовані джерела

Основний перелік

1. [The official home of the Python Programming Language](#)
2. M. Lutz, *Learning Python: Powerful Object-Oriented Programming, 5th ed.*, O'Reilly Media, Inc., 2013.

3. М. Лутц, *Изучаем Python, 4-е издание*, СИМВОЛ-Плюс, 2011.
4. D. Beazley, B. K. Jones, *Python Cookbook: Recipes for Mastering Python 3, 3rd ed.*, O'Reilly Media, Inc., 2013.
5. L. Ramalho, *Fluent Python: Clear, Concise, and Effective Programming*, O'Reilly Media, Inc., 2015.
6. М. Lutz, *Python Pocket Reference: Python in Your Pocket, 5th ed.*, O'Reilly Media, Inc., 2014.
7. М. Лутц, *Python Карманный справочник 5-е издание*, Издательский дом "Вильямс", 2015.
8. J. Zimmerman, *Python Descriptors: Understanding and Using the Descriptor Protocol*, 2nd ed., Apress, 2018.
9. V. Stukys, R. Damasevicius, *Meta-Programming and Model-Driven Meta-Program Development: Principles, Processes and Techniques*, Springer, 2013.

Додатковий перелік

1. B. Lubanovic, *Introducing Python: Modern Computing in Simple Packages*, O'Reilly Media, Inc., 2015.
2. A. Martelli, A. Ravenscroft, S. Holden, *Python in a Nutshell: The Definitive Reference, 3rd ed.*, O'Reilly Media, Inc., 2017.
3. M. Summerfield, *Python in Practice: Create Better Programs Using Concurrency, Libraries, and Patterns*, Addison-Wesley, 2014.
4. D. Hellmann, *The Python 3 Standard Library by Example*, Pearson Education, Inc., 2017.
5. D. Phillips, C. Giridhar, S. Kasampalis, *Python: Master the Art of Design Patterns*, Packt Publishing, 2016.
6. C. Giridhar, *Learning Python Design Patterns* 2nd ed., Packt Publishing, 2016.
7. D. C. Hay, *Data Model Patterns: A Metadata Map*, Morgan Kaufmann Publishers, 2006.
8. J. Pomerantz, *Metadata*, The MIT Press, 2015.
9. R. Gartner, *Metadata: Shaping Knowledge from Antiquity to the Semantic Web*, Springer, 2016.
10. R. L. Lubas, A. S. Jackson, I. Schneider, *The Metadata Manual: A Practical Workbook*, Chandos Publishing, 2013.
11. M. Voelter, *DSL Engineering: Designing, Implementing and Using Domain-Specific Languages*, CreateSpace Independent Publishing Platform, 2013.
12. R. Lammel, *Software Languages: Syntax, Semantics and Metaprogramming*, Springer, 2018.
13. P. Caplan, *Metadata Fundamentals for All Librarians*, American Library Association, 2003.
14. P. Riti, *Practical Scala DSLs: Real-World Applications Using Domain Specific Languages*, Apress, 2018.
15. R. D. Kelker, *Clojure for Domain-Specific Languages: Learn How to Use Clojure Language with Examples and Develop Domain-Specific Languages on the Go*, Packt Publishing, 2013.