

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА

ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ

Кафедра теорії та технології програмування

«ЗАТВЕРДЖУЮ»

Заступник декана
з навчальної роботи

_____ Кашпур О.Ф.

«___» _____ 2018 року

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Основи програмування

для студентів

галузь знань	03 «Гуманітарні науки» <i>(шифр і назва)</i>
спеціальність	035 «Філологія» (035.10 «Прикладна лінгвістика») <i>(шифр і назва спеціальності)</i>
освітній рівень	бакалавр <i>(молодший бакалавр, бакалавр, магістр)</i>
освітня програма	«Прикладна (комп'ютерна) лінгвістика та англійська мова» <i>(назва освітньої програми)</i>
вид дисципліни	обов'язкова

Форма навчання	денна
Навчальний рік	2018 / 2019
Семестр	2-3
Кількість кредитів ECTS	6
Мова викладання, навчання та оцінювання	українська
Форма підсумкового контролю	екзамен

Викладачі: **к.ф.-м.н., ас. Федорова М.В.** (лекції, семінарські заняття)

Пролонговано: на 20__/20__ н. р. _____ (_____) «__» __ 20__ р.
на 20__/20__ н. р. _____ (_____) «__» __ 20__ р.

КИЇВ – 2018

Розробник: Федорова Марія Вікторівна, к.ф.-м.н., асистент кафедри «Теорії та технології програмування»

ЗАТВЕРДЖЕНО

В.о. зав. кафедри «Теорії та технології програмування»

_____ Панченко Т.В.
(підпис) (прізвище та ініціали)

Протокол № ____ від « ____ » _____ 20__ р.

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

Протокол від « ____ » _____ 20__ року № ____

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

Затверджено Вченою радою факультету комп'ютерних наук та кібернетики

Протокол від « ____ » _____ 20__ року № ____

Голова Вченої ради _____ Анісімов А.В.
(підпис) (прізвище та ініціали)

Погоджено з науково-методичною комісією Інституту філології

Протокол від « ____ » _____ 20__ року № ____.

Голова науково-методичної комісії _____ Ситдикова І.В.
(підпис) (прізвище та ініціали)

« ____ » _____ 20__ року.

ВСТУП

1. Мета дисципліни – опанування базових знань з основ програмування, базових структур даних, комп'ютерних алгоритмів, набуття прикладних навичок процедурного програмування.

2. Попередні вимоги до опанування або вибору навчальної дисципліни:

1. *Успішне опанування курсів:* основ інформатики, вищої математики та дискретної математики;

2. *Знання теоретичних основ* теорії алгоритмів, теорії графів, теорії автоматів;

3. *Володіння* техніками та методиками роботи з цифровими даними, елементарними навичками роботи із комп'ютером.

3. Анотація навчальної дисципліни:

Навчальна дисципліна «Основи програмування» є складовою освітньо-професійної програми підготовки фахівців за першим (бакалаврським) рівнем вищої освіти галузі знань 03 «Гуманітарні науки» зі спеціальності 035 «Філологія» (035.10 «Прикладна лінгвістика»), освітньо-професійної програми «Прикладна (комп'ютерна) лінгвістика та англійська мова».

Дана дисципліна є обов'язковою навчальною дисципліною за програмою «Прикладна (комп'ютерна) лінгвістика та англійська мова».

Викладається у 2 та 3 семестрах **1 та 2** курсів в обсязі – **180 год.**

6 кредитів (ECTS) зокрема: *лекції – 14+14 год., семінарські заняття – 30+28 год., самостійна робота – 46+48 год.* У курсі передбачено 4 частини та 4 контрольні роботи. Завершується дисципліна – **екзаменом у 3 семестрі.**

В результаті вивчення навчальної дисципліни студент повинен:

знати: основні поняття, засоби та методи програмування на мові програмування Python;

вміти: демонструвати практичні навички у знаходженні програмного рішення певного класу лінгвістичних задач.

4. Завдання (навчальні цілі) - набуття знань, умінь та навичок (компетентностей) на рівні новітніх досягнень у програмуванні відповідно до освітньої кваліфікації «Бакалавр філології».

Зокрема, розвивати:

ФК 9. Розуміння сутності й соціального значення майбутньої професії, основних проблем лінгвістичних, математичних дисциплін та дисциплін інформаційних технологій, що визначають прикладну (комп'ютерну) лінгвістику як окрему філологічну спеціалізацію у взаємозв'язку цілісної системи міждисциплінарних знань;

ФК 27. Базові знання фундаментальних розділів математики, в обсязі, необхідному для володіння математичними поняттями та методами у галузях математичної, структурної, комп'ютерної лінгвістики та інженерії програмного забезпечення.

5. Результати навчання за дисципліною:

Результат навчання (1. знати; 2. вміти; 3. комунікація)		Форми (та/або методи і технології) викладання і навчання	Методи оцінювання та пороговий критерій оцінювання (за необхідності)	Відсоток у підсумко вій оцінці з дисци пліни
Код	Результат навчання			
РН1.1	Знати основні типи даних та функції над ними у мові Python, інтегровані середовища для програмування на мові Python.	Лекція, семінарське заняття, сам. робота	Відповідь на семінарському занятті, контрольна та екзаменаційна робота з відкритими відповідями	10%
РН1.2	Знати принцип роботи різних типів умовних операторів та циклів.	Лекція, семінарське заняття, сам. робота	Відповідь на семінарському занятті, контрольна та екзаменаційна робота з відкритими відповідями	30%
РН1.3	Знати основні задачі web-програміста, опанувати теоретичні засади створення зручного для користувача інтерфейсу та дизайну web-додатків та сайтів.	Лекція, семінарське заняття, сам. робота	Відповідь на семінарському занятті, контрольна та екзаменаційна робота з відкритими відповідями	20%
РН2.1	Вміти використовувати автоматичні методи обробки рядків для розв'язання лінгвістичних задач.	Семінарське заняття, самостійна робота	Відповідь на семінарському занятті, контрольна та екзаменаційна робота з відкритими відповідями	10%
РН2.2	Вміти будувати програмні рішення із використанням циклів, умовних операторів та функцій, використовуючи регулярні вирази та роботу з файлами.	Семінарське заняття, самостійна робота	Відповідь на семінарському занятті, контрольна та екзаменаційна робота з відкритими відповідями	10%
РН 3.1	Використовувати інформаційні технології та комунікативні стратегії у пошуках оптимальних розв'язків лінгвістичних задач.	Семінарське заняття, самостійна робота,	Відповідь на семінарському занятті, контрольна та екзаменаційна робота з відкритими відповідями	5%
РН3.2	Обґрунтовувати власний погляд при пошуку шляхів розв'язання лінгвістичних задач за допомогою програмування, представляти складну комплексну інформацію у стислій формі усно та письмово, відповідально ставитися до опрацювання наукової літератури	Семінарське заняття, самостійна робота	Відповідь на семінарському занятті, контрольна та екзаменаційна робота з відкритими відповідями	15%
				100%

6. Співвідношення результатів навчання дисципліни із програмними результатами навчання

Програмні результати навчання (за описом освітньої програми)	Результати навчання за дисципліною «Структурна лінгвістика»						
	РН 1.1	РН 1.2	РН 1.3	РН 2.1	РН 2.2	РН 3.1	РН 3.2
ПРН 1. Організувати процес свого навчання й самоосвіти.			+	+	+	+	+
ПРН 2. Ефективно працювати з інформацією: шукати і			+	+	+	+	+

добирати для вирішення професійних завдань необхідні інформаційно-довідникові ресурси з урахуванням сучасних досягнень науки і техніки, критично аналізувати, інтерпретувати, впорядковувати, класифікувати й систематизувати інформацію.							
ПРН 7. Знати норми української мови та використовувати українську мову як державну в усіх сферах суспільного життя, зокрема у професійному спілкуванні та для організації ефективної міжкультурної комунікації.	+	+	+	+	+	+	+
ПРН 10. Демонструвати та застосовувати філологічні знання для розв'язання професійних завдань у галузі прикладної (комп'ютерної) лінгвістики та в інших сферах професійної реалізації.	+	+	+	+	+	+	+
ПРН 15. Аналізувати мовні явища і процеси методами структурної, математичної та комп'ютерної лінгвістики, представляти в алгоритмічному вигляді процеси аналізу і синтезу лінгвістичних об'єктів.	+			+	+	+	+
ПРН 16. Планувати і здійснювати дослідження в галузі прикладної (комп'ютерної) лінгвістики на належному рівні, уміти описувати і презентувати результати досліджень.				+	+	+	+
ПРН 19. Проводити передпроектне обстеження предметної лінгвістичної галузі, системний аналіз лінгвістичного об'єкта програмного проектування.				+	+	+	+
ПРН 20. Уміння вибирати та використовувати відповідно до лінгвістичної задачі методологію створення програмного забезпечення				+	+	+	+
ПРН 21. Знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізів та математичного моделювання, сучасні мови програмування, для розробки програмного забезпечення.				+	+		
ПРН 22. Створювати та супроводжувати програмне забезпечення лінгвістичних інформаційних технологій.	+	+	+	+	+		
ПРН 23. Вміти розробляти людино-машинний інтерфейс.	+	+	+	+	+		
ПРН 25. Знати та вміти застосовувати інформаційні технології обробки, зберігання та передачі текстових даних.	+	+	+	+	+		
ПРН 26. Знати підходи щодо оцінки та забезпечення якості програмного забезпечення.				+	+	+	+
ПРН 27. Уміння документувати та презентувати результати розробки програмного забезпечення.					+	+	+
ПРН 31. Використовувати математичні методи та математичний понятійний апарат у лінгвістичних дослідженнях та у створенні лінгвістичних інформаційних технологій.	+	+	+	+	+		

7. Схема формування оцінки.

7.1 Форми оцінювання студентів:

- семестрове оцінювання:

1. Контрольна робота (тест): РН 1.1., РН 1.2, РН 3.2 — 5 балів/3 бали.
2. Контрольна робота (тест): РН1.1, РН 1.2., РН 2. — 5 балів/3 бали.
3. Контрольна робота (тест): РН 1.1., РН 2.1 — 5 балів/3 бали.
4. Контрольна робота (тест): РН1.1, РН 1.3., РН 2.1, РН 2.2 — 5 балів/3 бали.
1. Лабораторна робота 1 (проект): РН 2.1 – 10 балів/6 балів.
2. Лабораторна робота 2 (проект): РН 2.1 – 10 балів/6 балів.
3. Лабораторна робота 3 (проект): РН 2.2 – 10 балів/6 балів.
4. Лабораторна робота 4 (проект): РН 2.2 – 10 балів/6 балів.

- підсумкове оцінювання (у формі екзамену) вказується:

- максимальна кількість балів які можуть бути отримані студентом: 40 балів;

- результати навчання які будуть оцінюватись: РН1.1, РН1.2, РН1.3, РН2.1, РН2.2;

- форма проведення і види завдань: письмова

Види завдань:

Структура екзаменаційної роботи та критерії оцінювання:

1. Теоретичне запитання.
2. Практичне завдання «регулярні вирази».
3. Практичне завдання «робота з файлами» або «програмування мовою Scratch»

Критерії оцінювання екзаменаційної роботи

Завдання	Вид завдання	Максимальний відсоток (бал)	Всього відсотків (балів)
Завдання 1	Письмове запитання з курсу	25% (10 балів)	25% (10 балів)
Завдання 2, 3	Практичне завдання	по 37,5% (15 балів)	по 37,5% (15 балів)
Всього			100% (40 балів)

Питання, що виносяться на екзамен

Теоретичні питання до екзамену:

1. Робота з файлами у Python: ім'я файлу, відкриття, запис та закриття файлів, порядкове зчитування, конструкція with open as, позиція курсора у файлі.
2. Життєвий цикл програмного забезпечення.
3. Склад команди виконавців проекту: особливості роботи та функціональні обов'язки технічних виконавців та управлінців.

4. Технічне завдання: означення, функції, необхідні складові ТЗ для виконавця та замовника.
5. Типи тестування: за знанням системи, за ступенем автоматизації, за часом проведення тестування, за ступенем підготовленості до тестування.
6. Культура коду. Оптимізація коду. Рефакторинг.

Практичні завдання до екзамену.

1. Регулярні вирази.

Всі завдання на вхід отримують файл, на вихід програми мають записувати відповідь у новий файл. У випадку, якщо задача реалізована тільки у консолі, ставиться дві треті виділених балів за завдання.

1. За допомогою регулярного виразу написати програму, що шукає в тексті усі посилання, видаляє теги та залишає тільки назви:

Було: `назва`

Стало: назва

2. За допомогою регулярного виразу написати програму, що шукає в тексті усі заголовки (h1-h6), видаляє теги та залишає тільки назви заголовків:

Було: `<h3>назва</h3>`

Стало: назва

3. За допомогою регулярного виразу написати програму, що шукає в тексті усі абзаци, видаляє теги та залишає тільки текст:

Було: `<p>текст</p>`

Стало: текст

4. За допомогою регулярного виразу написати програму, що шукає в тексті усі картинки, видаляє їх та залишає їх підписи:

Було: ``

Стало: підпис

5. За допомогою регулярного виразу написати програму, що видаляє в тексті усі коментарі `<!-- Appl -->`

Було: текст `<!-- Appl -->` продовження

Стало: текст продовження

6. За допомогою регулярного виразу написати програму, що шукає в тексті усі роки життя та видаляє їх:

Було: Алан Тюрінг (1912 - 1954), Джон фон Нейман (1903 - 1957)

Стало: Алан Тюрінг, Джон фон Нейман

7. За допомогою регулярного виразу написати програму, що шукає в тексті усі дати у форматі 2016-11-23 та замінює їх на формат 23.11.16

Було: Ми вивчали це 2016-11-23.

Стало: Ми вивчали це 23.11.16.

8. За допомогою регулярного виразу написати програму, що шукає в тексті усі власні назви (послідовність слів, що починається з великої літери), та виводить в файл список цих назв через кому:

Було: Ми цікавимося доробками Алана Тюрінга та Оскара Моргенштерна, вивчаючи теорію алгоритмів та теорію ймовірностей.

Стало: Алана Тюрінга, Оскара Моргенштерна,

9. За допомогою регулярного виразу написати програму, що шукає в тексті усі таблиці та прибирає їх:

Було: текст перед таблицею `<table>вміст таблиці</table>` та після

Стало: текст перед таблицею та після

10. За допомогою регулярного виразу написати програму, що шукає в тексті усі вирази у круглих дужках та прибирає їх:

Було: Цікаво вивчати програмування (особливо в жовтому корпусі)!

Стало: Цікаво вивчати програмування!

11. За допомогою регулярного виразу написати програму, що шукає в тексті усі вирази у фігурних дужках та прибирає їх:

Було: Цікаво вивчати програмування {особливо в жовтому корпусі}!

Стало: Цікаво вивчати програмування!

10. За допомогою регулярного виразу написати програму, що шукає в тексті усі вирази у квадратних дужках та прибирає їх:

Було: Цікаво вивчати програмування [особливо в жовтому корпусі]!

Стало: Цікаво вивчати програмування!

2. Робота з файлами. Scratch.

1. Написати програму, яка:

- створює файл із назвою 1.txt
- записує у нього текст "Мене звать. Я навчаюся в."
- питає у користувача як його звать та де він навчається
- записує відповідь в потрібному місці у файлі 1.txt:

Було: Мене звать. Я навчаюся в.

Стало: Мене звать Марія. Я навчаюся в університеті імені Т. Шевченка.

2. Створити проект у Scratch у якому виконуються наступні умови:

- змінюється костюм спрайта
- змінюється фон
- є мінімум 3 різних спрайта
- хоча б один спрайт рухається залежно від дій користувача
- використовуються цикли, розгалуження, умови та змінні
- у спрайтів змінюються як горизонтальні, так і вертикальні координати
- всі спрайти не виходять за межі екрану
- хоча б один спрайт зникає чи з'являється.

Студент не допускається до екзамену, якщо під час семестру набрав менше ніж 24 бали. Студент допускається до екзамену за умови виконання 70% передбачених планом лабораторних робіт.

7.2 Організація оцінювання:

Терміни проведення форм оцінювання:

1. Контрольна робота (тест): до 7 тижня 2 семестру.
2. Контрольна робота (тест): до 15 тижня 2 семестру.
3. Контрольна робота (тест): до 7 тижня 3 семестру.
4. Контрольна робота (тест): до 14 тижня 3 семестру.
5. Лабораторна робота 1 (проект): до 7 тижня 2 семестру.
6. Лабораторна робота 2 (проект): до 15 тижня 2 семестру.
7. Лабораторна робота 3 (проект): до 7 тижня 3 семестру.
8. Лабораторна робота 4 (проект): до 14 тижня 3 семестру.

Студент має право на одне перескладання контрольної роботи із можливістю отримання максимально 3 бали за кожен раз. Термін перескладання визначається викладачем.

У випадку відсутності студента з поважних причин відпрацювання та перездачі контрольної роботи здійснюються у відповідності до „Положення про

порядок оцінювання знань студентів при кредитно-модульній системі організації навчального процесу” від 1 жовтня 2010 року.

У разі неякісного виконання лабораторної роботи, викладач має право не зарахувати лабораторну роботу, або знизити за неї бали.

Студент має право здавати лабораторні роботи після закінчення визначеного для них терміну, але з втратою одного балу за кожен тиждень, який пройшов з моменту закінчення терміну її здачі.

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 ЛЕКЦІЙНИХ І СЕМІНАРСЬКИХ ЗАНЯТЬ

№ п/п	Назва теми лекції / практичного заняття	Кількість годин		
		лекції	семінарські зан.	самостійна робота
1.	Основні поняття, пов'язані з алгоритмами та мовами програмування. Scratch. Інтерпритовані мови програмування на прикладі мови програмування Python.	2	1	4
2.	Інтегрована середа розробки на прикладі Visual Studio Code та PyCharm.		1	4
3.	Типи даних. Дії над даними. Робота з числами у Python.	2	2	4
4.	Рядки у Python. Перетворення рядка на число та навпаки. Операції над рядками у Python.	2	4	4
5.	Умовні оператори у Python. Порівняння блок-схем та коду програми на мові програмування Python.	2	4	4
6.	Створення інтелектуального чат-боту на мові програмування Python.		4	8
7.	Контрольна робота		2	
8.	Цикли на мові програмування Python. Конструкція «якщо, то».	1	2	4
9.	Цикли на мові програмування Python. Конструкція «поки».	1	2	4
10.	Особливості роботи циклів з різними типами даних на мові програмування Python.	2	2	4
11.	Функції у мові програмування Python. Глобальні та локальні змінні. Фактичні та формальні аргументи функції.	2	4	6

12.	Контрольна робота		2	
Усього за 1 семестр		14	30	46
13.	Розв'язання лінгвістичних задач за допомогою мови програмування Python.		2	6
14.	Принципи розробки програмного забезпечення. Професійні ролі. Життєвий цикл програмного забезпечення.	2		
15.	Регулярні вирази у мові програмування Python.	2		
16.	Використання веб-ресурсів для роботи з регулярними виразами мови програмування Python.		2	2
17.	Робота з регулярними виразами Python в інтегрованій середі розробки.		4	4
18.	Контрольна робота		2	
19.	Робота з файлами у Python. Режими роботи з файлами. Відкриття, читання та закриття файлів.	2	2	4
20.	Запис у файл за допомогою мови програмування Python. Метод write.	2	4	6
21.	Запис у файл та читання з файлу за допомогою мови програмування Python. Конструкція with open as.	2	4	8
22.	Застосування роботи з файлами та регулярні вирази для розв'язання лінгвістичних задач.	2	6	8
23.	Вступ до web-технологій, web-дизайну та UI UX.	2	2	10
24.	Контрольна робота		2	
Усього за 2 семестр		14	28	48

Загальний обсяг 180 год.

У тому числі:

Лекцій – 28 год.

Практичних – 58 год.

Самостійна робота – 94 год.

9. Рекомендована література

Основна:

1. Ascher, D., & Lutz, M. (1999). Learning Python. O'Reilly.
2. Shaw, Z. A. (2010). Learn Python The Hard Way. A Very Simple Introduction to the Terrifyingly Beautiful World of Computers and Code.
3. Zlobin, G. (2013). Learning Python design patterns. Packt Publishing Ltd.

Додаткова:

1. Бауэр А. Гооз Г. Информатика. - М.:Мир.-1976.
2. Вирт Н. Систематическое программирование. Введение.- М.:Мир.-1987.
3. Вирт Н. Алгоритмы и структуры данных. - М.:Мир.-1989.
4. Кнут Д. Искусство программирования.-Т.1. Получисленные алгоритмы.- 3-е изд. М.- 2001.
5. Зубенко В.В., Омельчук Л.Л. Програмування. Поглиблений курс. – К.: Видавнично-поліграфічний центр «Київський інститут», 2011. – 623 с.
6. Зубенко В.В. Науково-методичні матеріали до нормативного курсу «Програмування та обчислювальні системи», факультет кібернетики КНУ ім.Тараса Шевченка, 1 курс, спеціальність «Інформатика», 2004, <http://www.unicyb.kiev.ua/~vvz>
7. Белов Ю.А. , Проценко В.С., Чаленко П.Й Інструментальні засоби програмування.- К.: Либідь.-1993.
8. Майерс Г. Надежность программного обеспечения. . - М.:Мир,1980.
9. Ставровский А.Б. Первые шаги в программировании. Самоучитель. – М.: “Вильямс”.- 2003.
10. Манакова О.І., Редько В.Н. Елементи редуційного програмування. Навчальний посібник.-Київ:КНУ, 2001, 57 с13.