

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
ПРОГРАМНА ІНЖЕНЕРІЯ**

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	магістр <i>(молодший бакалавр, бакалавр, магістр)</i>
освітня програма	«Прикладна лінгвістика: редакторсько-перекладацька та експертна діяльність» <i>(назва освітньої програми)</i>

вид дисципліни **вибіркова (спеціалізація «Автоматичне оброблення природної мови – NLP»)**

Форма навчання	денна
Навчальний рік	2018/2019
Семестр	3
Кількість кредитів ECTS	3
Мова викладання, навчання та оцінювання	українська
Форма заключного контролю	залік

Викладачі: **к.п.н., ас. Русіна Н.Г.** (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: Омельчук Людмила Леонідівна, к.ф.-м.н., доцент кафедри «Теорії та технології програмування», Русіна Наталія Геннадіївна, к.п.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.

1. Мета дисципліни – засвоєння базових знань з основ інженерії програмування, оцінювання економічних показників продукту і діяльності його виконавців, керування програмними проектами та промислове виготовлення складних програмних систем. Підготовка професіоналів, здатних ставити і розв'язувати завдання, що пов'язані з аналізом та формулюванням вимог, створенням, супроводженням та забезпеченням якості програмного забезпечення.

2. Попередні вимоги до опанування або вибору навчальної дисципліни:

1. Успішне опанування основ програмування.
2. Знання теоретичних основ: формальних методів розробки програмних систем.

3. Анотація навчальної дисципліни:

Навчальна дисципліна «Програмна інженерія» є складовою освітньо-наукової програми підготовки фахівців за освітнім рівнем «магістр» галузі знань 03 «Гуманітарні науки» зі спеціальності 035 «Філологія» (035.10 «Прикладна лінгвістика»), освітньо-наукової програми «Прикладна лінгвістика: редакторсько-перекладацька та експертна діяльність».

Дана дисципліна є дисципліною спеціалізації «*Автоматичне оброблення природної мови – NLP*» за освітньою програмою «*Прикладна лінгвістика: редакторсько-перекладацька та експертна діяльність*».

Викладається у 3 семестрі в обсязі – **90 годин**.

(**3 кредити ECTS**) зокрема: лекції – 9 год., практичні – 10 год., самостійна робота – 71 год. Завершується дисципліна заліком в 3 семестрі.

В результаті вивчення навчальної дисципліни студент повинен:

знати: методи аналізу прикладної області, виявлення інформаційних потреб і збору вихідних даних для проектування; базові методи і прийоми проектування інформаційних систем і технологій; сучасні моделі і методи оцінки якості та надійності на всіх стадіях життєвого циклу інформаційних систем; професійні стандарти і інші нормативно-правові документи в галузі інформаційних технологій; кодекс професійної етики і слідувати йому в житті.

вміти: оцінювати, класифікувати і обґрунтовувати вибір методів формування вимог до інформаційної системи, формулювати вимоги; оцінювати і вибирати методи і моделі створення, впровадження, експлуатації інформаційних систем та управління ними на всіх етапах життєвого циклу; обґрунтовувати, аналізувати і оцінювати прийняті проектні рішення.

Місце дисципліни. Навчальна дисципліна "Програмна інженерія" є складовою циклу професійної підготовки фахівців освітнього рівня "магістр" за спеціалізацією «Автоматичне оброблення природної мови – NLP» освітньої програми «Прикладна лінгвістика: редакторсько-перекладацька та експертна діяльність».

4. Завдання (навчальні цілі):

набуття знань, умінь та навичок (компетенцій) на рівні досягнень у прикладній лінгвістиці, відповідно до освітньої кваліфікації «Магістр з прикладної лінгвістики за спеціалізацією автоматичне оброблення природної мови – NLP». Зокрема, розвивати:

- здатність спілкуватися державною мовою як усно, так і письмово;
- здатність до пошуку, опрацювання та аналізу інформації з різних джерел;
- здатність працювати в команді та автономно;
- здатність застосовувати і розвивати фундаментальні і міждисциплінарні знання для успішного розв'язання наукових завдань інженерії програмного забезпечення у лінгвістичній предметній галузі.

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			
РН1.1	Знати методи аналізу прикладної області, виявлення інформаційних потреб і збору вихідних даних для проектування	Лекція, практичне заняття	Тест, 60% правильних відповідей, Залік	25%
РН1.2	Знати професійні стандарти і інші нормативно-правові документи в галузі інформаційних технологій; кодекс професійної етики	Лекція, практичне заняття	Тест, 60% правильних відповідей, Залік	25%
РН1.3	Знати моделі і методи оцінки якості та надійності на всіх стадіях життєвого циклу інформаційних систем	Лекція, практичне заняття	Доповідь	10%
РН2.1	Вміти оцінювати і вибирати методи і моделі створення, впровадження, експлуатації інформаційних систем та управління ними на всіх етапах життєвого циклу.	Практичне заняття, самостійна робота	Доповідь	10%
РН3.1	Обґрунтовувати власний погляд на задачу, спілкуватися з колегами з питань проектування та розробки програм, складати письмові звіти	Практичне заняття, лабораторна робота, самостійна робота	Захист лабораторної роботи	20%
РН4.1	Відповідально ставитися до виконуваних робіт, нести відповідальність за їх якість	Лабораторна робота, самостійна робота	Захист лабораторної роботи	10%

6. Співвідношення результатів навчання дисципліни із програмними результатами навчання

Результати навчання дисципліни	РН 1.1	РН 1.2	РН 1.3	РН 2.1	РН 3.1	РН 4.1
Програмні результати навчання						
(з опису освітньої програми)						
ПРН 1. Оцінювати власну навчальну та науково-професійну діяльність, будувати і втілювати ефективну стратегію саморозвитку та професійного самовдосконалення.					+	+
ПРН 25. Знати і застосовувати сучасні професійні стандарти і інші нормативно-правові документи з інженерії програмного забезпечення.	+	+	+	+	+	+

7. Схема формування оцінки.

7.1 Форми оцінювання студентів:

- семестрове оцінювання:

1. Контрольна робота (тест) 1: РН 1.1., РН 1.2 — 25 балів/25 балів.
2. Контрольна робота (тест) 2: РН1.2, РН1.3 - 25 балів/25 балів.
3. Підготовка доповіді (за темами лекційних занять та самостійної роботи): РН 2.1 – 20 балів/20 балів.
4. Лабораторна робота (проект з оформленим технічним завданням): РН 3.1, РН 4.1 – 30 балів/30 балів.

- підсумкове оцінювання у формі заліку. Виставляється за результатами роботи студентами впродовж усього семестру та не передбачає додаткових заходів оцінювання для успішних студентів.

Запитання для підготовки до заліку

1. Основні поняття програмної інженерії.
2. Предмет і мета курсу програмної інженерії. Сутність дисциплін щодо програмної інженерії.
3. Виникнення та розвиток інженерії програмування. Основи інженерного проектування систем.
4. Сутність технології програмування. Моделі, методи та засоби інженерії програмування та програмної інженерії.
5. Моделі життєвого циклу. Типи моделей (водоспадна, ітеративна, спіральна ін.).
6. Стандарт ISO/IEC 12207 – Процеси ЖЦ ПС. Задачі процесів.
7. Побудова и опис структури і змісту нового ЖЦ для конкретних цілей.
8. Аналіз стандартних процесів ЖЦ (проектування, верифікації, валидації, вимірювання, оцінювання тощо) і співставлення с різними прикладними моделями ЖЦ.
9. Основні принципи моделювання програмних систем.
10. Життєвий цикл ПС. Моделі життєвого циклу ПС. Ітеративні моделі життєвого циклу. Керування ризиками.
11. Життєвий цикл ПС. Моделі життєвого циклу ПС. Модель Гантера «Фази-Функції».
12. Діаграми класів та шаблони проектування. Структура шаблонів. Класифікація шаблонів: породжуючі, структурні, поведінки.
13. Шаблонів проектування. Антишаблони.
14. Вимоги до продукту та процесу.
15. Функціональні та нефункціональні вимоги.
16. Методології розробки програмного забезпечення.
17. Методології розробки програмного забезпечення.
18. Поняття життєвого циклу ПЗ. Послідовні та ітераційні методології розробки.
19. Методи збирання та аналізу вимог до ПЗ.
20. Розробка вимог до ПЗ.
21. Специфікація вимог до ПЗ.
22. Сценарний підхід до розробки вимог.
23. Завдання верифікації та валидації ПЗ.

24. Причини появи помилок на етапах ЖЦ.
25. Основні поняття та методи тестування програм.
26. Засоби формальної специфікації програм.

7.2 Організація оцінювання:

Терміни проведення форм оцінювання:

1. *Контрольна робота (тест): до 6 тижня семестру.*
2. *Контрольна робота (тест): до 11 тижня семестру.*
3. *Підготовка доповіді (за темами лекційних занять та самостійної роботи): до 5 тижня семестру.*
4. *Лабораторна робота (проект з оформленим технічним завданням): до 10 тижня семестру.*

Студент має право на одне перескладання контрольної роботи із можливістю отримання максимально 10 балів за кожну. Термін перескладання визначається викладачем.

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

Студент має право здавати лабораторну роботу та доповідь 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 лекцій і лабораторних занять

№ лекції	Назва лекції	Кількість годин		
		Лекції	Лабораторна робота	Самостійн. робота
	Частина 1. Інженерія програмного забезпечення			
1	Тема 1. Основні поняття програмної інженерії.	1		4
2	Тема 2. Кодекс етики	1		4
3	Тема 3. Стандарти програмної інженерії	1	1	4
4	Тема 4. Области знань програмної інженерії за SWEBOOK	1	2	5
5	Тема 5. Моделі життєвого циклу ПЗ	1	1	4
6	Тема 6. Керування життєвим циклом розробки ПЗ		1	5
7	Тема 7. Методології розробки програмного забезпечення. Прогнозуємі методології.			5
8	Тема 8. Методології розробки програмного забезпечення. Адаптивні методології.			5
	<i>Контрольна робота 1</i>			
Контроль за підсумками лабораторної роботи				
Всього по частині 1		5	5	36
	Частина 2. Теорія і практика оцінювання програмного продукту і процесів			
9	Тема 9. Програмні вимоги.	1	1	5
10	Тема 10. Проектування.	1	1	5
11	Тема 11. Якість ПЗ.		1	5
12	Тема 12. Верифікація і атестація.	1	1	5
13	Тема 13. Тестування.	1	1	5
14	Тема 14. Супроводження ПЗ.			5
15	Тема 15. Оцінка вартості			5
	<i>Контрольна робота 2</i>			
Контроль за підсумками доповіді				
Всього по частині 2		4	5	35
ВСЬОГО		9	10	71

Загальний обсяг 90 год., в тому числі:

Лекцій – **9 год.**

Лабораторні заняття - **10 год.**

Самостійна робота - **71 год.**

9. Рекомендовані джерела:

Основна

1. Abran, A. and J.W. Moore (exec. eds); P. Bourque and R. Dupuis (eds.). 2004. *Guide to the Software Engineering Body of Knowledge (SWEBOOK)*. Piscataway, NJ, USA: The Institute of Electrical and Electronic Engineers, Inc. (IEEE). P. Bourque and R.E. Fairley (eds), *Guide to the Software Engineering Body of Knowledge, Version 3.0*, IEEE Computer Society, 2014; Available at <http://www.swebok.org>
2. European e-Competence Framework, e-CF [El. resource]. — URL: www.ecompetences.eu.
3. Computer Science Curriculum 2013: Strawman Draft. The Joint Task Force on Computing Curricula, Association for Computing Machinery, IEEE-Computer Society [El. resource]. — URL: <http://ai.stanford.edu/users/sahami/CS2013/strawman-draft/cs2013-strawman.pdf>.
4. SE 2014 - Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering [Електронний ресурс]. — 2015. — Режим доступу до ресурсу: <http://www.acm.org/binaries/content/assets/education/se2014.pdf>.
5. Software Engineering Body of Knowledge. Точка доступа https://en.wikipedia.org/wiki/Software_Engineering_Body_of_Knowledge.
6. Буч Г., Якобсон А., Рамбо Дж. UML. Классика CS. 2-е изд. / Пер. с англ.; Под общей редакцией проф. С. Орлова — СПб.: Питер, 2006. — 736 с. ISBN 5-469-00599-2
7. Крэг Ларман. Применение UML 2.0 и шаблонов проектирования = Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development. — 3-е изд. — М.: Вильямс, 2006. — 736 с. — ISBN 0-13-148906-2
8. Соммервиль.И. Инженерия программного обеспечения. — М.; “Вильямс”, 2002. — 624с.
9. Боэм Б.У. Инженерное проектирование программного обеспечения. -М.: Радио и связь, 1985.—511с.
10. Кибанов А.Я. Управление персоналом.—2005.—352с.
11. Лаврищева К.М. Програмна інженерія. —Підручник.—К.: Академперіодика, 2008.—415с.

Додаткова:

1. В.В. Зубенко, Л.Л. Омельчук. Програмування : навчальний посібник (гриф МОН України) / - К. : ВПЦ "Київський університет", 2011. - 623 с.
2. <http://msdn.microsoft.com/ru-RU/>