

Міністерство освіти і науки України

Харківський національний університет імені В.Н. Каразіна

Кафедра теоретичної та прикладної системотехніки

“ЗАТВЕРДЖУЮ”

Проректор з науково-
педагогічної роботи



“ _____ 2019 р.

Робоча програма навчальної дисципліни

Обрані методи інженерії ПЗ

рівень вищої освіти третій (освітньо-науковий)

галузь знань 12 Інформаційні технології

спеціальність 122 Комп'ютерні науки

освітньо-наукова програма підготовки докторів філософії 122 Комп'ютерні науки

спеціалізація _____

вид дисципліни обов'язкова

факультет комп'ютерних наук

2019 / 2020 навчальний рік

Програму обговорено та рекомендовано до затвердження вченою радою факультету комп'ютерних наук

“28” серпня 2019 року, протокол № 3

РОЗРОБНИК ПРОГРАМИ:

доктор технічних наук, старший науковий співробітник, професор кафедри теоретичної та прикладної системотехніки **Толстолюзька Олена Геннадіївна**

Програму схвалено на засіданні кафедри теоретичної та прикладної системотехніки
Протокол від “19” червня 2019 року № 14

Завідувач кафедри теоретичної та прикладної системотехніки


Шматков С. І.

Програму погоджено методичною комісією факультету комп'ютерних наук
Протокол від “ 20 ” червня 2019 року № 9

Голова методичної комісії факультету комп'ютерних наук


Бердніков А.Г.

ВСТУП

Програма навчальної дисципліни «Обрані методи інженерії програмного забезпечення» складена відповідно до освітньо-наукової програми підготовки третього (освітньо-наукового) рівня (докторів філософії) зі спеціальності 122 Комп'ютерні науки.

1. Опис навчальної дисципліни

1.1. Мета викладання навчальної дисципліни

Метою викладання навчальної дисципліни є формування знань, умінь і навичок в області інженерії програмного забезпечення.

1.2. Основні завдання вивчення дисципліни

Основними завданнями вивчення дисципліни є:

- оволодіння сучасними методами і технологіями інженерії програмного забезпечення;
- ознайомлення з ядром знань SWEBOOK;
- ознайомлення зі складовими життєвого циклу програмного забезпечення;
- ознайомлення та свідоме використання моделей життєвого циклу програмного забезпечення;
- надання знань з програмування CASE-засобів при об'єктно-орієнтованому проектуванні ПС.

1.3. Кількість кредитів – 6

1.4. Загальна кількість годин - 180

1.5. Характеристика навчальної дисципліни	
Нормативна	
Денна форма навчання	Заочна (дистанційна) форма навчання
Рік підготовки	
1-й	1-й
Семестр	
2-й	2-й
Лекції	
24 год.	24 год.
Семінарські заняття	
12 год.	12 год.
Лабораторні заняття	
0 год.	0 год.
Самостійна робота	
144 год.	144 год.
Індивідуальні завдання	
0 год.	

1.6. Заплановані результати навчання

Згідно з вимогами освітньо-професійної програми, студенти мають досягти таких результатів навчання:

знати:

- науково-методичні знання в галузі інженерії програмного забезпечення;
- сучасні методи і технології інженерії програмного забезпечення;
- культуру інженерії програмного забезпечення і моделі зрілості;

- складові життєвого циклу програмного забезпечення;
- складові інженерії програмного забезпечення; їх застосування і зв'язок.

уміти:

- застосовувати знання та уміння із наукової та професійної підготовки при вирішенні спеціалізованих завдань в галузі інформаційних технологій (зокрема, формувати підходи до свідомого застосування моделей життєвого циклу програмного забезпечення);
- аналізувати результати наукових досліджень в області ІСТ, власні припущення і зроблені іншими припущення, які вважаються доведеними;
- порівнювати, кількісно та якісно оцінювати, корегувати очікувані/отримані результати;
- створювати концепцію власних оригінальних наукових досліджень, організовувати наукові дослідження та експерименти в області ІСТ;
- вдосконалювати концепції, підходи, моделі, методи, методики, технології, створення ІСТ та уміння проведення наукових досліджень в даній галузі, передавати особистий науковий досвід;
- координувати дослідження, модифікувати методи дослідження, комбінувати різні методи дослідження для досягнення поставленої мети в області ІСТ;
- розробляти інформаційні системи та технології для використання в різних галузях науки, техніки, економіки, освіти, оборонної промисловості, транспорту, медицини, адміністративного управління та ін. на базі сучасних методів і технологій інженерії ПЗ;
- отримувати оцінки вартості програмного забезпечення.

2. Тематичний план навчальної дисципліни

Тема 1. Введення в програмну інженерію та процеси життєвого циклу.

Визначення програмної інженерії, її місце в інженерній діяльності фахівців при створенні комп'ютерних систем та загальний опис десяти областей знань професіонального ядра знань SWEBOOK.

Тема 2. Моделі життєвого циклу для розробки програмних систем.

Опис основних моделей ЖЦ, які використовуються в практиці проектування програмних систем. Характеристика фундаментальних моделей ЖЦ (водоспадної, спіральної, інкрементної, еволюційної) та стандартної моделі.

Тема 3. Методи визначення вимог в програмній інженерії.

Методи та інженерія вимог до системи. Процес збору, накопичення та специфікації вимог. Класифікація вимог та характеристика функціональних і не функціональних вимог.

Тема 4. Методи аналізу і побудови моделей предметної області.

Методи аналізу предметної області та побудови моделей. Об'єктно-орієнтовані та стандартизовані, традиційні методи проектування архітектури системи.

Тема 5. Методи проектування програмних систем.

Опис базових основ методів систематичного (структурного, компонентного, аспектно-орієнтованого та ін.) і теоретичного (алгебраїчного, композиційного і алгеброалгоритмічного) програмування.

Тема 6. Інженерія додатків та інженерія предметної області.

Сучасні тенденції і напрямки розвитку інженерії додатків в сенсі побудови окремих ПС з ПІК та інженерії Про з багаторазовим застосуванням використаних рішень для родини ПС.

Тема 7. Методи верифікації і тестування програм та систем.

Опис методів перевірки правильності програм: формальним методам доказу, основаних на аксіомах та ствердженнях, верифікації і тестування ПС на етапах ЖЦ.

Тема 8. Методи інтеграції, перетворення та зміни компонентів і даних.

Основи інтеграції та перетворення програм і даних, а також методи зміни (реінженерія, реверсна інженерія і рефакторінга) компонентів і систем.

Тема 9. Моделі якості та надійності в програмній інженерії.

Моделі якості ПС, метрики та методи досягнення та визначення якості ПС.

Тема 10. Методи управління проектом, ризиком та конфігурацією.

Аналіз и опис інженерії програмування, принципів та методів планування і управління програмним проектом, ризиками та формуванням версій ПС.

Тема 11. Засоби та інструменти програмної інженерії.

Огляд сучасних засобів програмування та характеристика широко використаних CASE-засобів (Project Management, Rational Rose, MSF, RUP, CORBA, DCOM та ін.), при об'єктно-орієнтованому проектуванні ПС.

3. Структура навчальної дисципліни

Назви розділів і тем	Кількість годин											
	Денна форма						Заочна форма					
	Усього	у тому числі					Усього	у тому числі				
л		п	лаб	інд	ср	л		п	лаб	інд	ср	
1	2	3	4	5	6	7	8	9	10	11	12	13
Тема 1. Вступ. Введення в програмну інженерію та процеси життєвого циклу.	6	2				4	6	2				4
Тема 2. Моделі життєвого циклу для розробки програмних систем.	18	2	2			14	18	2	2			14
Тема 3. Методи визначення вимог в програмній інженерії.	18	2	2			14	18	2	2			14
Тема 4. Методи аналізу і побудови моделей предметної області.	18	2	2			14	18	2	2			14
Тема 5. Методи проектування програмних систем.	16	2				14	16	2				14
Тема 6. Інженерія додатків та інженерія предметної області.	16	2				14	16	2				14
Тема 7. Методи верифікації і тестування програм та систем.	18	2	2			14	18	2	2			14

Тема 8. Методи інтеграції, перетворення та зміни компонентів і даних.	16	2				14	16	2				14
Тема 9. Моделі якості та надійності в програмній інженерії.	18	2	2			14	18	2	2			14
Тема 10. Методи управління проектом, ризиком та конфігурацією.	16	2				14	16	2				14
Тема 11. Засоби та інструменти програмної інженерії.	20	4	2			14	20	4	2			14
Усього годин	180	24	12			144	180	24	12			144

4. Теми практичних занять

№ з/п	Назва теми	Кількість годин
1	Моделі життєвого циклу для розробки програмних систем.	2
2	Методи визначення вимог в програмній інженерії.	2
3	Методи аналізу і побудови моделей предметної області.	2
4	Методи верифікації і тестування програм та систем.	2
5	Моделі якості та надійності в програмній інженерії.	2
6	Засоби та інструменти програмної інженерії.	2
	Разом	12

5. Завдання для самостійної роботи

№ з/п	Види, зміст самостійної роботи	Кількість годин
1	Аналіз умов виникнення кризи в програмному забезпеченні.	4
2	Метод інженерії вимог А. Джекобсона	14
3	Метод аналізу і побудови моделей С.Шлаер та С.Меллора	14
4	Моделі культур організацій Константіноса та Де Грака.	14
5	Моделі зрілості можливостей.	14
6	Стандарти, які визначають поняття продукт інженерії програмного забезпечення.	14
7	Технічне проектування.	14
8	Метод функціональної декомпозиції.	14
9	Метод моделювання UML	14
10	Основні етапи перетворення даних в БД	14
11	Засоби оцінювання ПЗ, засновані на моделях SLIM і COCOMO.	14
	Разом	144

6. Індивідуальні завдання

(не має)

7. Методи контролю

Контроль засвоєння навчального матеріалу здійснюється шляхом:

- проведення тестування за результатами відпрацювання основних положень навчальної програми;
- проведення письмового підсумкового контролю знань.

Максимальна кількість балів за результатами контролю поточної успішності складає 60 балів.

Згідно рішення методичної комісії кафедри теоретичної та прикладної системотехніки факультету комп'ютерних наук до екзаменів не допускаються студенти, які мають заборгованість по лабораторним або контрольним роботам.

Підсумковий контроль здійснюється шляхом проведення екзамену.

Екзаменаційний білет включає два теоретичних і одне практичне питання. Теоретичні питання оцінюються в 10 балів кожен, практичний - в 20.

Максимальна кількість балів за результатами екзамену складає 40 балів.

8. Схема нарахування балів

Поточний контроль, самостійна робота, індивідуальні завдання											Контрольні роботи, передбачені навчальним планом	Курсова робота	Разом	Екзамен	Сума
Теми															
Т 1	Т 2	Т 3	Т 4	Т 5	Т 6	Т 7	Т 8	Т 9	Т 10	Т 11					
5	5	5	5	5	5	5	5	5	5	10			60	40	100

T1, T2 ... – теми розділів.

За темою Т 1 студент отримує 5 балів за виконання практичного заняття 1.

За темою Т 2 студент отримує по 5 балів за виконання практичного заняття 1.

За темою Т 3 студент отримує по 5 балів за виконання практичного заняття 2.

За темою Т 4 студент отримує 5 балів за виконання практичного заняття 2.

За темою Т 5 студент отримує по 5 балів за виконання практичного заняття 3.

За темою Т 6 студент отримує 5 балів за виконання практичного заняття 3.

За темою Т 7 студент отримує 5 балів за виконання практичного заняття 4.

За темою Т 8 студент отримує 5 балів за виконання практичного заняття 4.

За темою Т 9 студент отримує 5 балів за виконання практичного заняття 5.

За темою Т 10 студент отримує 5 балів за виконання практичного заняття 5.

За темою Т 11 студент отримує 10 балів за виконання практичного заняття 6.

Критерії оцінювання знань студентів за практичні роботи

Вимоги	Кількість балів
<ul style="list-style-type: none"> ▪ Завдання відзначається повнотою виконання без допомоги викладача. ▪ Визначає рівень поінформованості, потрібний для прийняття рішень. Вибирає інформаційні джерела,. ▪ Робить висновки і приймає рішення у ситуації невизначеності. Володіє вміннями творчо-пошукової діяльності. 	5

<ul style="list-style-type: none"> ▪ Завдання – повні, з деякими огріхами, виконані без допомоги викладача. ▪ Планує інформаційний пошук; володіє способами систематизації інформації. ▪ Студент може зіставити, узагальнити, систематизувати інформацію під керівництвом викладача; вільно застосовує вивчений матеріал у стандартних ситуаціях. ▪ Робить висновки і приймає рішення у ситуації невизначеності. Володіє уміннями творчо-пошукової діяльності. 	3
<ul style="list-style-type: none"> ▪ Завдання відзначається неповнотою виконання за консультацією викладача. ▪ Застосовує запропонований вчителем спосіб отримання інформації, має фрагментарні навички в роботі з підручником, науковими джерелами; ▪ Вибирає відомі способи дій для виконання фахових методичних завдань. 	1

Критерії оцінювання екзаменаційних робіт студентів

Вимоги	Кількість балів
Показані всебічні систематичні знання та розуміння навчального матеріалу; безпомилково виконані завдання.	35-40
Показані повні знання навчального матеріалу; помилки, якщо вони є, не носять принципового характеру.	30-35
Показано повне знання необхідного навчального матеріалу, але допущені помилки.	20-30
Показано повне знання необхідного навчального матеріалу, але допущені суттєві помилки	10-20
Показано недосконале знання навчального матеріалу, допущені суттєві помилки.	5-10
Показано недосконале знання навчального матеріалу, допущені суттєві помилки, які носять принциповий характер; обсяг знань не дозволяє засвоїти предмет.	1-5

Шкала оцінювання

Сума балів за всі види навчальної діяльності протягом семестру	Оцінка	
	для чотирирівневої шкали оцінювання	для дворівневої шкали оцінювання
90 – 100	відмінно	зараховано
70-89	добре	
50-69	задовільно	
1-49	незадовільно	не зараховано

9. Рекомендована література

Основна література

1. Кане С. Тестирование программного обеспечения / С. Кане. - М: DiaSoft, 2001. - 542с.
2. Саммервил И. Инженерия программного обеспечения / И. Саммервил. - М.: Вильяме, 2002. - 720 с.

3. Сидоров Н.А. Повторное использование, переработка и восстановление программного обеспечения / Н.А. Сидоров // Управляющие системы и машины. - 2000. - № 3, 4. - С. 27 - 37.
4. Van Veendabl E. Standard glossary of term used in Software testing / E. Van Veendabl. - ISTQB. - 2007. Vol. 1,2. - June. - 30 p.

Допоміжна література

5. Bosch J. Design and use of software architectures / J. Bosch. -Addison Wesley, 2000. - 325 p.
6. Black R. Critical Testing Processes / R. Black. - Addison-Wesley, 2003.
7. Blum B.A. Taxonomy of Software Development Methods / B.A. Blum // Coinmunication of the ACM. - 1994. - Vol. 37, n. 11,-P. 82 - 94.
8. Budgen D. Software design: Reading / D. Budgen. - Addison-Wesley, 1994.-320 p.
9. Glass R.L. Extrime programming the good, the bad. and the bottom line / R.L. Glass // IEEE. Software. - 2001, - Vol.18, n.7. - P.11 -P. 112,- 111 - 112.
10. Georgiadou E. Software Process and Product Improvement: A Historical Perspective / E. Georgiadou // Кибернетика и системный анализ. - 2003. № 1, - P. 147 - 177,
11. Jacobson I. The Unified Software Development Process / I. Jacobson, G. Booch, J. Rumbaugh. - Addison-Wesley, 1999. - 310 p.
12. Jonsson P. Software Reuse. Architecture. Process and Organization for Business Success. Person Education Asia / P. Jonsson. - 2002. — 497 p.
13. Martin J. Rapid Application Development / J. Martin. -Macmillan, 1991. -250 p.
14. Perry D. Models of Software Development Environments / D. Perry, G. Kaiser // IEEE Trans. On Soft. Engin. - 1991. - Vol. 17, n. 3.-P. 283-295.
15. Railich V. Software cultures and evolution / V. Railich, N. Wilde // Computer. 2001. - Sept. - P.25 - 28.
16. Rombach H.D, Software specifications: a framework / H. D. Rombach. - Carnegie mullon Univ: SET, 1990. - 30 p.
17. Rajlich W. A stage Model for the Software Life Cycle / W. Rajlich, K. Bennett // Computer. - 2000. - July. - P. 77 - 70.
18. Spillner A. Software Testing Foundations / A. Spillner, T. Linz, H. Schafer. - Dpunkt: Verlag, 2007. - 277 p.
19. Sidorov N.A. Software Stylistics / N. A. Sidorov // Proceedings of NAU. - 2005. - 2(24), - P. 98 - 103.
20. Toriik Matsumoto K. An Environment for Computer - Aided Empirical Software Engineering / Matsumoto K. Gingerz Toriik // IEEE Trans, On Software Eng. - 1999. - Vol. 25, №. 4 - P. 474 - 485 .
21. Creting a software engineering culture / K. Wiegers // Dorset House Publishing. - New York, 2003. - 358 p.

10. Посилання на інформаційні ресурси в Інтернеті, відео-лекції, інше методичне забезпечення

1. TOP500 Supercomputing sites : Project ranks and details the 500 most powerful computer systems in the world [Electronic resource]. Available from : <http://www.top500.org>.

2. <http://www.cs.wisc.edu/condor/>
3. <http://setiathome.ssl.berkeley.edu/>
4. <http://www.Distributed.net/>
5. <http://mersenne.org/>
6. <http://www.globus.org/>
7. <http://www.eu-datagrid.org/>