

Харківський національний університет імені В.Н. Каразіна
Факультет комп'ютерних наук
Кафедра теоретичної та прикладної системотехніки

УХВАЛЕНО
Вченою радою факультету
комп'ютерних наук, протокол № 4
від «03» грудня 2019 р.
Голова Вченої ради _____



Назва курсу	Обрані методи інженерії програмного забезпечення
Викладач (-і)	професор Толстолузька Олена Геннадіївна
Профайл викладача (-ів)	http://www-csd.univer.kharkov.ua/about-us/sub-faculty/tase-department/personalnij-sklad/
Контактний тел.	Кафедральний 707-50-22
E-mail:	elena.tolstoluzka@karazin.ua
Сторінка курсу в системі дистанційного навчання	
Консультації	<i>Очні консультації:</i> розклад в університеті (на кафедрі) <i>Он лайн- консультації:</i> через e-mail

ЗМІСТ

1.	Коротка анотація до курсу	3
2.	Мета та цілі курсу	3
3.	Формат курсу	3
4.	Результати навчання	3
5.	Обсяг курсу	4
6.	Ознаки курсу	4
7.	Пререквізити	4
8.	Технічне та програмне забезпечення /обладнання	4
9.	Політики курсу	4
10.	Схема курсу	5
11.	Система оцінювання та вимоги	11
12.	Рекомендована література	12

1. Коротка анотація до курсу–курс «Обрані методи інженерії програмного забезпечення» спрямований на поглиблення знань в області сучасних інформаційних технологій для розробки програмного забезпечення, розвиненню здатності розв'язувати комплексні задачі та проблеми в області ІСТ, формулювання вимог, проектування, верифікації та тестування програмного забезпечення задля проведення власного наукового дослідження, результати якого мають наукову новизну, теоретичне та практичне значення.

2. Мета та цілі курсу – розвинення здатності розробки та використання нового програмного забезпечення для аналізу та обробки великих масивів даних, розробки та експлуатації комп'ютерних систем, створення ефективних систем верифікації комп'ютерних моделей, організації проектів розробки програмних систем і управління процесом розробки.

3. Формат курсу – очний, заочний.

4. Результати 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.

5. Обсяг курсу

Вид заняття	Загальна кількість годин
лекції	24
семінарські	12
самостійна робота	144

6. Ознаки курсу:

Рік викладання	семестр	спеціальність	Курс (рік навчання)	Нормативний\ вибірковий
2020	2	Комп'ютерні науки	1	нормативний (Н)

7. Пререквізити – перед вивченням курсу здобувач повинен знати вищу математику, алгоритмізацію та програмування, прикладне програмне забезпечення, об'єктно–орієнтоване програмування, системне програмування.

8. Технічне та програмне забезпечення /обладнання – семінарські заняття проводяться у спеціалізованому комп'ютерному класі на 15 робочих місць, який оснащений сучасними комп'ютерами з ліцензованою операційною системою Windows та програмним забезпеченням (мова програмування C++).

9. Політики курсу

Академічна доброчесність - це сукупність етичних принципів та визначених законом правил, якими мають керуватися учасники освітнього процесу під час навчання, викладання та провадження наукової (творчої) діяльності з метою забезпечення довіри до результатів навчання та/або наукових (творчих) досягнень. Дотримання академічної доброчесності здобувачами освіти передбачає самостійне виконання навчальних завдань, завдань поточного та підсумкового контролю результатів навчання, а саме написання та обговорення рефератів, виконання контрольних робіт за своїм варіантом. Контрольні роботи виконуються кожним здобувачем і захищаються індивідуально.

Тиж. / акад. год.	Тема, план, короткі тези	Форма діяльності (заняття)* *лекція, СЗ, СРС) / Формат** **аудиторна, СРС	Матеріали	Завдання, год
Тиж. 1 6 год.	<p>Тема 1. Введення в програмну інженерію та процеси життєвого циклу.</p> <p>Лекція 1. Визначення програмної інженерії, її місце в інженерній діяльності фахівців при створенні комп'ютерних систем та загальний опис десяти областей знань професіонального ядра знань SWEBOOK.</p>	Лекція / аудиторна	Презентація лекції	2 год
	СРС. Аналіз умов виникнення кризи в програмному забезпеченні в сімдесяті роки минулого століття.	СРС / СРС	Конспект лекції	опрацювати лекцію, ознайомитись з літературою, 4 год
Тиж. 2 18 год.	<p>Тема 2. Моделі життєвого циклу для розробки програмних систем.</p> <p>Лекція 2. Опис основних моделей ЖЦ, які використовуються в практиці проектування програмних систем. Характеристика фундаментальних</p>	Лекція / аудиторна	Презентація лекції	2 год

	моделей ЖЦ (водоспадної, спіральної, інкрементної, еволюційної) та стандартної моделі.			
	СРС. Метод інженерії вимог А. Джекобсона.	<i>СРС</i>	Конспект лекції, Striuk A. M. Softwar eengineering: first 50 year sofformation and development // ComputerScience&SoftwareEngineer ing : Proceedingsofthe 1st StudentWorkshop (CS&SE@SW 2018), KryvyiRih, Ukraine, November 30, 2018 / Editedby : Arnold E. Kiv, Serhiy O. Semerikov, Vladimir N. Solo-viev, Andrii M. Striuk. – P. 11-36. – (CEUR WorkshopProceedings (CEUR- WS.org), Vol. 2292). – режим доступу : http://ceur-ws.org/Vol-2292/paper01.pdf	опрацювати лекцію, ознайомитись з літературою, 14 год.
	СЗ 1. Моделі життєвого циклу для розробки програмних систем.	Семинарське заняття.	Конспект лекції	опрацювати лекцію, ознайомитись з літературою, скласти доповідь, 2год
Тиж. 3 18 год	<i>Тема 3.</i> Методи визначення вимог в програмній інженерії. <i>Лекція 3.</i> Методи та інженерія вимог	<i>Лекція / аудиторна</i>	Презентація лекції	2год

	до системи. Процес збору, накопичення та специфікації вимог. Класифікація вимог та характеристика функціональних і не функціональних вимог.			
	СРС. Метод аналізу і побудови моделей С.Шлаер та С.Меллора .	<i>СРС</i>	Конспект лекції, Striuk A. M. Softwareengineering: first 50 yearsofformationanddevelopment // ComputerScience&SoftwareEngineer ing : Proceedingsofthe 1st StudentWorkshop (CS&SE@SW 2018), KryvyiRih, Ukraine, November 30, 2018 / Editedby : Arnold E. Kiv, Serhiy O. Semerikov, Vladimir N. Solo-viev, Andrii M. Striuk. – P. 11-36. – (CEUR WorkshopProceedings (CEUR- WS.org), Vol. 2292). – режим доступу : http://ceur-ws.org/Vol-2292/paper01.pdf	опрацювати лекцію, ознайомитись з літературою, 14 год.
	СЗ 2. Методи визначення вимог в програмній інженерії.	Семінарське заняття.	Конспект лекції, самостійно підібрана література	опрацювати лекцію, ознайомитись з літературою, скласти доповідь 2 год
Тиж. 4	<i>Тема 4.</i> Методи аналізу і побудови	<i>Лекція / аудиторна</i>	Презентація лекції	2 год

18 год	моделей предметної області. <i>Лекція 4.</i> Методи аналізу предметної області та побудови моделей. Об'єктно-орієнтовані та стандартизовані, традиційні методи проектування архітектури системи.			
	СРС. Моделі культур організацій Константіноса та Де Грака.	<i>СРС</i>	Конспект лекції	Опрацювати лекцію, ознайомитись з літературою, 14 год..
	СЗ 3. Методи аналізу і побудови моделей предметної області.	Семінарське заняття.	Конспект лекції	Опрацювати лекцію, ознайомитись з літературою, 2 год.
Тиж. 5 16 год	<i>Тема 5.</i> Методи проектування програмних систем. <i>Лекція 5.</i> Опис базових основ методів систематичного (структурного, компонентного, аспектно-орієнтованого та ін.) і теоретичного (алгебраїчного, композиційного і алгеброалгоритмічного) програмування.	<i>Лекція / аудиторна</i>	Презентація лекції	2 год.

	СРС. Моделі зрілості можливостей.	СРС	Конспект лекції, М.О.Сидоров Вступ до інженерії програмного забезпечення.-К.: Вид-во Нац. авіац. ун-ту «НАУ-друк», 2010. - 112 с. (С. 15-20)	опрацювати лекцію, ознайомитись з літературою, 14 год..
Тиж. 6 16 год	Тема 6. Інженерія додатків та інженерія предметної області. Лекція 6. Сучасні тенденції і напрямки розвитку інженерії додатків в сенсі побудови окремих ПС з ПК та інженерії ПрО з багаторазовим застосуванням використаних рішень для родини ПС.	Лекція / аудиторна	Презентація лекції	2 год.
	СРС. Стандарти, які визначають поняття продукт інженерії програмного забезпечення.	СРС	Конспект лекції, ISO/IEC 12207:2008 Systems and software engineering — Software lifecycle processes, режим доступу: https://www.iso.org/standard/43447.html	опрацювати лекцію, ознайомитись з літературою, 14 год.
Тиж. 7 18 год	Тема 7. Методи верифікації і тестування програм та систем. Лекція 7. Опис методів перевірки правильності програм: формальним методам доказу, основаних на аксіомах та ствердженнях, верифікації і тестування ПС на етапах ЖЦ.	Лекція / аудиторна	Презентація лекції	2 год

	СРС. Технічне проектування.	<i>СРС</i>	Конспект лекції,	опрацювати лекцію, ознайомитись із літературою, 14 год.
	СЗ 4. Методи верифікації і тестування програм та систем.	Семінарське заняття	Конспект лекції	опрацювати лекцію, ознайомитись з літературою, 2 год.
Тиж. 8 16 год	<i>Тема 8.</i> Методи інтеграції, перетворення та зміни компонентів і даних. <i>Лекція 8.</i> Основи інтеграції та перетворення програм і даних, а також методи зміни (реінженерія, реверсна інженерія і рефакторінга) компонентів і систем.	<i>Лекція / аудиторна</i>	Презентація лекції	2 год.
	СРС. Метод функціональної декомпозиції.	<i>СРС</i>	Конспект лекції, ISO/IEC/IEEE 16326:2009 Systems and software engineering – Project management режим доступу: https://www.iso.org/standard/41977.html	опрацювати лекцію, ознайомитись із літературою, 14 год.
Тиж. 9	<i>Тема 9.</i> Моделі якості та надійності в	<i>Лекція / аудиторна</i>	Презентація лекції	2год

18 год	програмній інженерії. <i>Лекція 9.</i> Моделі якості ПС, метрики та методи досягнення та визначення якості ПС.			
	СРС. Метод моделювання UML	<i>СРС</i>	Конспект лекції	опрацювати лекцію, ознайомитись із літературою, 14год
	СЗ 5. Моделі якості та надійності в програмній інженерії.	Семінарське заняття	Конспект лекції	опрацювати лекцію, ознайомитись з літературою, 2 год..
Тиж. 10 16 год	<i>Тема 10.</i> Методи управління проектом, ризиком та конфігурацією. <i>Лекція 10.</i> Аналіз и опис інженерії програмування, принципів та методів планування і управління програмним проектом, ризиками та формуванням версій ПС.	<i>Лекція / аудиторна</i>	Презентація лекції	опрацювати лекцію, переглянути презентацію, 2год
	СРС. Основні етапи перетворення даних в БД	<i>СРС</i>	Конспект лекції	опрацювати лекцію, ознайомитись із літературою, 14 год.

Тиж. 11 9 год.	Тема 11. Засоби та інструменти програмної інженерії. Лекція 11. Огляд сучасних засобів програмування та характеристика широко використаних CASE-засобів	<i>Лекція / аудиторна</i>	Презентація лекції	опрацювати лекцію, переглянути презентацію, 2 год
	СРС. Засоби оцінювання ПЗ, засновані на моделі SLIM.	<i>СРС</i>	Конспект лекції	опрацювати лекцію, 7 год.
Тиж. 12 11 год.	Лекція 12. Використання Project Management, Rational Rose, MSF, RUP, CORBA, DCOM при об'єктно-орієнтованому проектуванні ПС.	<i>Лекція / аудиторна</i>	Презентація лекції	опрацювати лекцію, переглянути презентацію, 2 год
	СРС. Засоби оцінювання ПЗ, засновані на моделі COCOMO.	<i>СРС</i>	Конспект лекції	опрацювати лекцію, 7 год.
	СЗ 6. Засоби та інструменти програмної інженерії.	Семінарське заняття	Конспект лекції	опрацювати лекцію, ознайомитись з літературою, 2 год.

***якщо література подається в скороченому вигляді, то розшифрування подаєте вкінці

11. Система оцінювання та вимоги
Загальна система оцінювання курсу.

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

T1, T2 ... – теми розділів.

За темою T 1 студент отримує 5 балів за виконання семінарського заняття 1.

За темою T 2 студент отримує по 5 балів за виконання семінарського заняття 1.

За темою T 3 студент отримує по 5 балів за виконання семінарського заняття 2.

За темою T 4 студент отримує 5 балів за виконання семінарського заняття 2.

За темою T 5 студент отримує по 5 балів за виконання семінарського заняття 3.

За темою T 6 студент отримує 5 балів за виконання семінарського заняття 3.

За темою T 7 студент отримує 5 балів за виконання семінарського заняття 4.

За темою T 8 студент отримує 5 балів за виконання семінарського заняття 4.

За темою T 9 студент отримує 5 балів за виконання семінарського заняття 5.

За темою T 10 студент отримує 5 балів за виконання семінарського заняття 5.

За темою T 11 студент отримує 10 балів за виконання семінарського заняття 6.

Загальна система оцінювання курсу	<p><i>участь в роботі впродовж семестру/екзамен - 60/40</i></p> <p>Розподіл балів, що присвоюються студентам з навчальної дисципліни, є сумою балів за виконання всіх видів завдань та самостійну роботу плюс бали, отримані під час екзамену. Впродовж семестру студент за виконання всіх завдань може отримати до 60 балів і до 40 балів під час складання екзамену.</p>
Семінарські заняття	<p>На семінарських заняттях в режимі діалогу відбувається опрацювання вивченого матеріалу та розвинення здатності розв'язувати комплексні задачі та проблеми в області ІСТ. Особлива увага приділяється вмінню застосовувати моделі, методи, методики, технології інженерії ПЗ при проведенні особистих наукових досліджень.</p>
Умови допуску до підсумкового контролю	<p>Умовою допуску студента до підсумкового семестрового контролю (екзамену) є наявність прийнятих і зарахованих звітів з семінарських завдань і наявність мінімальної кількості балів (не менше ніж 20) з поточного тестування та контролю.</p> <p>Згідно рішення кафедри теоретичної та прикладної системотехніки факультету комп'ютерних наук до екзамену не допускаються студенти, які мають заборгованість по практичним або контрольним роботам.</p> <p>В кінці курсу всі бали підсумовуються і студент має можливість (в разі успішного виконання всіх завдань) отримати максимальну підсумкову оцінку 100 балів.</p>

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 екзамену	для заліку
90 – 100	відмінно	зараховано
70-89	добре	
50-69	задовільно	
1-49	не задовільно	не зараховано

12. Рекомендована література

Основна література

1. Кане С. Тестирование программного обеспечения / С. Капе. - М: DiaSoft, 2001. - 542с.

2. Саммервил И. Инженерия программного обеспечения / И. Саммервил. - М.: Вильямс, 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. VanVeendabl E. Standard glossary of termusedin Software testing / E. VanVeendabl. - ISTQB. - 2007. Vol. 1,2. - June. - 30 p.

Допоміжна література

5. Bosch J. Designanduseofsoftwarearchitectures / J. Bosch. -Addison Wesley, 2000. - 325 p.
6. Black R. CriticalTestingProcesses / R. Black. - Addison-Wesley, 2003.
7. Blum B.A. TaxonomyofSoftwareDevelopmentMethods / B.A. Blum // Coinmunicationofthe ACM. - 1994. - Vol. 37, n. 11,-P. 82 - 94.
8. Budgen D. Softwaredesign: Reading / D. Budgen. - Addison-Wesley, 1994.-320 p.
9. Glass R.L. Extrimeprogrammingthegood, thebad. andthebottomline / R.L. Glass // IEEE. Software. - 2001, - Vol.18, n.7. - P.11 -P. 112,- 111 - 112.
10. Georgiadou E. SoftwareProcessandProductImprovement: A HistoricalPerspective / E. Georgiadou // Кибернетика и системныйанализ. - 2003. № 1, - P. 147 - 177,
11. Jacobson I. TheUnifiedSoftwareDevelopmentProcess / I. Jacobson, G. Booch, J. Rumbaugh. - Addison-Wesley, 1999. - 310 p.
12. Jonsson P. SoftwareReuse. Architecture. ProcessandOrganizationforBusinessSuccess. PersonEducationAsia / P. Jonsson. - 2002. — 497 p.
13. Martin J. RapidApplicationDevelopment / J. Martin. -Macmillan, 1991. -250 p.
14. Perry D. ModelsofSoftwareDevelopmentEnvironments / D. Perry, G. Kaiser // IEEE Trans. OnSoft. Engin. - 1991. - Vol. 17, n. 3.-P. 283-295.
15. Railich V. Softwareculturesandevolution / V. Railich, N. Wilde // Computer. 2001. - Sept. - P.25 - 28.
16. Rombach H.D, Softwarespecifications: a framework / H. D. Rombach. - CarnegiemullonUniv: SET, 1990. - 30 p.
17. Rajlich W. A stageModelfortheSoftwareLifeCycle / W. Rajlich, K. Bennett // Computer. - 2000. - July. - P. 77 - 70.
18. Spillner A. SoftwareTestingFoundations / A. Spillner, T. Linz, H. Schafer. - Dpunkt: Verlag, 2007. - 277 p.
19. Sidorov N.A. SoftwareStylistics / N. A. Sidorov // Proceedingsof NAU. - 2005. - 2(24), - P. 98 - 103.
20. ToriikMatsumoto K. AnEnvironmentforComputer - AidedEmpiricalSoftwareEngineering / Matsumoto K. GingerzToriik // IEEE Trans, OnSoftwareEng. - 1999. - Vol. 25, №. 4 - P. 474 - 485 .
21. Creting a softwareengineeringculture / K. Wiegers // DorsetHousePublishing. - NewYork, 2003. - 358 p.

13. Посилання на інформаційні ресурси в Інтернеті, відео-лекції, інші методичні 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/>
8. <http://cyb.univ.kiev.ua/library/books/lavrishcheva-6.pdf>
9. <https://www.you-books.com/book/M-O-Sidorov/Vstup-do-inzhenerii-programnogo-zabezpechennya>
10. <http://ceur-ws.org/Vol-2292/paper01.pdf>