



КУРСОВА РОБОТА З КОМПОНЕНТІВ ПРОГРАМНОЇ ІНЖЕНЕРІЇ

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Перший (бакалаврський)</i>
Галузь знань	<i>12 Інформаційні технології</i>
Спеціальність	<i>121 Інженерія програмного забезпечення</i>
Освітня програма	<i>Інженерія програмного забезпечення інформаційних систем</i>
Статус дисципліни	<i>Нормативна</i>
Форма навчання	<i>очна (денна)</i>
Рік підготовки, семестр	<i>3 курс, осінній семестр</i>
Обсяг дисципліни	<i>1 кредит</i>
Семестровий контроль/ контрольні заходи	<i>захист курсової роботи</i>
Розклад занять	
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	<i>ст. викл., PhD Головченко Максим Миколайович, k.o.o.v.g.v.n.s@gmail.com, доц., к.ф.-м.н., доц. Поперешняк Світлана Володимирівна, доц., к.пед.н., доц. Зубик Людмила Володимирівна</i>
Розміщення курсу	<i>ipi.kpi.ua/navchannya/silabusy/</i>

Програма навчальної дисципліни

1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

Метою дисципліни «Курсова робота з компонентів програмної інженерії» є закріплення у студентів навичок проходження кожного з етапів життєвого циклу програмного забезпечення для вирішення прикладних задач.

Курсова робота – це вид самостійної роботи з елементами дослідження, що виконується студентами протягом семестру з метою закріплення, поглиблення і узагальнення знань, здобутих за час вивчення дисципліни «Компоненти програмної інженерії», та їхнього застосування до комплексного вирішення конкретного фахового завдання. Її написання є обов'язковою складовою індивідуального навчального плану бакалавра спеціальності 121 «Інженерія програмного забезпечення» за освітньою програмою «Інженерія програмного забезпечення інформаційних систем».

Предмет даної навчальної дисципліни – моделі життєвого циклу програмного забезпечення, засоби а способи аналізу вимог до програмного забезпечення, засоби проектування, мови програмування, інструментальні засоби створення програмного забезпечення, засоби та способи аналізу якості та тестування програмного забезпечення, засоби розгортання та підтримки програмного забезпечення.

Написання курсової роботи сприяє формуванню у студентів наступних компетентностей.

Загальні компетентності:

ЗК2. Здатність застосовувати знання у практичних ситуаціях.

Фахові компетентності, формуванню яких сприяє даний кредитний модуль:

ФК1. Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення.

ФК2. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.

ФК3. Здатність розробляти архітектури, модулі та компоненти програмних систем.

ФК4. Здатність формулювати та забезпечувати вимоги щодо якості програмного забезпечення у відповідності з вимогами замовника, технічним завданням та стандартами.

ФК5. Здатність дотримуватися специфікацій, стандартів, правил і рекомендацій в професійній галузі при реалізації процесів життєвого циклу.

ФК7. Володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних.

ФК8. Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення.

ФК10. Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя.

ФК11. Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення.

ФК12. Здатність здійснювати процес інтеграції системи, застосовувати стандарти і процедури управління змінами для підтримки цілісності, загальної функціональності і надійності програмного забезпечення.

ФК13. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення.

Формування зазначених компетентностей забезпечує досягнення наступних програмних результатів навчання:

ПРН1. Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки.

ПРН4. Знати і застосовувати професійні стандарти і інші нормативноправові документи в галузі інженерії програмного забезпечення.

ПРН6. Уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення.

ПРН7. Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.

ПРН8. Вміти розробляти людино-машинний інтерфейс.

ПРН12. Застосовувати на практиці ефективні підходи щодо проектування програмного забезпечення.

ПРН14. Застосовувати на практиці інструментальні програмні засоби доменного аналізу, проектування, тестування, візуалізації, вимірювань та документування програмного забезпечення.

ПРН23. Вміти документувати та презентувати результати розробки програмного забезпечення.

2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)

Для успішного засвоєння дисципліни необхідно мати базові знання з

- дисципліни «Компоненти програмної інженерії» (ПО 07),
- дисципліни «Бази даних» (ПО 05),
- дисципліни «Основи програмування» (ПО 02),
- дисципліни «Алгоритми і структури даних» (ПО 01).

Успішне оволодіння знаннями з дисципліни готує студентів до вивчення наступних дисциплін, таких як:

- «Технології паралельних обчислень» (ПО 21),
- «Моделювання систем» (ПО 24),
- «Переддипломна практика» (ПО 10),
- «Дипломне проектування» (ПО 11).

3. Зміст навчальної дисципліни

Основні етапи виконання курсової роботи:

Етап	Найменування та зміст	Результат	Години
1	<i>Підготовчий етап</i> Ознайомлення з темою, узгодження ТЗ	Узгоджене ТЗ	2
2	<i>Етап виконання</i> Аналіз вимог до ПЗ, проектування ПЗ, розробка ПЗ, тестування ПЗ, розгортання ПЗ	Працездатне ПЗ	23
3	<i>Заключний етап</i> Оформлення пояснювальної записки (згідно ДСТУ «Документація. Звіти у сфері науки і техніки. Структура і правила оформлення»)	Пояснювальна записка КР	4.5
4	<i>Захист КР</i>		0.5

Структурні частини пояснювальної записки:

Курсова робота складається з пояснювальної записки, разом з супроводжувальною документацією, та не обов'язкового графічного матеріалу (креслень).

Структура курсової роботи:

Вступна частина:

- титульний аркуш;
- завдання на курсову роботу;
- календарний план-графік виконання роботи;
- реферат (анотація) українською мовою;
- реферат (анотація) англійською мовою;
- відомість роботи.

Основна частина:

- технічне завдання;
- пояснювальна записка (ПЗ);
- текст програми.

Додатки.

4. Навчальні матеріали та ресурси

Базова література

1. Software Architecture in Practice. Third Edition. Len Bass, Paul Clements, Rick Kazman. – Addison-Wesley Professional, 2012. – 624с.
2. Офіційна нотація мови UML (OMG UML) Version 2.5.1 [Електронний ресурс]. – Режим доступу: <https://www.omg.org/spec/UML/>
3. Офіційна нотація мови BPMN Version 2.0 [Електронний ресурс]. – Режим доступу: http://www.bpmn.de/images/BPMN2_0_Poster_EN.pdf
4. Сидоров М.О. Вступ до інженерії програмного забезпечення. – К.: НАУ, 2008. - 65с.
5. SWEBOOK Guide V3.06 2014, IEEE Society.
6. ISO/IEC/IEEE 12207 Systems and software engineering – Software life cycle processes.
7. ISO/IEC/IEEE 29148 Systems and software engineering – Life cycle processes Requirements engineering.
8. A Guide to the Business Analysis Body of Knowledge (BABOK Guide) Version 3.0: International Institute of Business Analysis, Toronto, Ontario, Canada, 2015.- 514 p.
9. Software Architecture in Practice (SEI Series in Software Engineering) 3rd Edition. Len Bass, Paul Clements, Rick Kazman. Publisher: Addison-Wesley, 2012. 624 pages.
10. Designing Software Architectures: A Practical Approach. Humberto Cervantes, Rick Kazman, May 2016. Publisher: Addison-Wesley.
11. Documenting Software Architectures. Views and Beyond. 2nd Edition Paul Clements, Felix Bachmann, Len Bass, David Garlan, James Ivers. Addison Wesley, 2011. 592 pages.

Допоміжна література

1. C++. Основи програмування. Теорія та практика: підручник / [О. Г. Трофименко, Ю. В. Прокоп, І. Г. Швайко, Л. М. Буката та ін.] ; за ред. О. Г. Трофименко. – Одеса : Фенікс, 2010. – 544 с.
2. Головна сторінка середовища MS Visual Studio [Електронний ресурс]. – Режим доступу: <http://visualstudio.com>
3. Головна сторінка MSDN [Електронний ресурс]. – Режим доступу: <http://msdn.microsoft.com>
4. Сайт підтримки Python MSDN [Електронний ресурс]. – Режим доступу: <http://python.org>
5. Python documentation [Електронний ресурс]. – Режим доступу : <https://docs.python.org/doc/7>

Політика та контроль

5. Політика навчальної дисципліни (освітнього компонента)

Студент має виконувати КР протягом семестру, дотримуючись затвердженого календарного плану.

Усі етапи виконання КР студент має виконувати **самостійно**, дотримуючись Кодексу честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» (kpi.ua/code). У випадку виявлення факту академічної недобросовісності робота не зараховується (ставиться 0 балів).

КР вважається виконаною, якщо студент надав текст пояснювальної записки КР (у вигляді файлу(-ів) і у паперовому вигляді) та захистив її перед комісією (показав працездатність програмного забезпечення, відповів на усі питання).

6. Види контролю та рейтингова система оцінювання результатів навчання (PCO)

6.1 Поточний контроль

Поточний контроль виконання КР виконується шляхом перевірки готовності розділів КР згідно затвердженого графіка виконання роботи.

6.2 Календарний контроль

Проводиться двічі на семестр як моніторинг поточного стану виконання вимог робочої програми.

6.3 Система рейтингових балів та критерії оцінювання

Рейтингова оцінка з курсової роботи має дві складові: виконання курсової роботи та її захист.

Перша (стартова) складова характеризує роботу студента з курсового проектування та її результат - якість пояснювальної записки та розробленого програмного забезпечення; друга складова характеризує якість захисту студентом курсової роботи.

Розмір шкали першої складової ($r1$) - 60 балів, а другої складової ($r2$) - 40 балів.

Стартова складова виконання курсової роботи ($r1$):

- розробка технічного завдання – до 5 балів;
- аналіз предметної області задачі – до 5 балів;
- аналіз існуючих технологій та успішних ІТ-проектів – до 5 балів;
- аналіз вимог до ПЗ – до 5 балів;
- модельювання ПЗ – до 5 балів;
- архітектура ПЗ – до 5 балів;
- конструювання ПЗ – до 5 балів;
- тестування ПЗ – до 5 балів;
- розгортання ПЗ – до 5 балів;
- якість оформлення пояснювальної записки та супроводжувальної документації з

урахуванням виконання вимог нормативних документів – до 15 балів;

–несвоєчасність виконання курсової роботи – -1 бал за кожен день запізнення відповідно графіку (22.12.2024).

Складова захисту курсової роботи ($r2$):

- розробка програмного забезпечення – до 20 балів;
- якість доповіді –до 10 балів;
- відповіді на питання – до 10 балів.

Увага!! До захисту формально не допускаються роботи у яких виконано менше 60% завдань з ($r1$) або відсутнє програмне забезпечення. Також не допускаються роботи у яких складова виконання курсової роботи ($r1$) менше 20. Можливе отримання до 6 заохочувальних балів.

Узагальнені критерії оцінювання виконання курсової роботи та деталізовані бальні шкали наведено в Методичних вказівках до виконання КР.

Сума балів двох складових переводиться до залікової оцінки згідно з таблицею:

<i>Кількість балів</i>	<i>Оцінка</i>
100-95	Відмінно
94-85	Дуже добре
84-75	Добре
74-65	Задовільно
64-60	Достатньо
Менше 60	Незадовільно

Робочу програму навчальної дисципліни (силабус):

Складено ст. вик., PhD кафедри ІІІ М. Головченком

Ухвалено кафедрою ІІІ (протокол № 16_ від _29.05.2024)

Погоджено Методичною комісією факультету (протокол № 10 від 21.06.2024)