

Методологія інженерії програмного забезпечення. Курсовий проєкт

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти

Другий (магістерський)

Галузь знань	12 Інформаційні технології
Спеціальність	121 Інженерія програмного забезпечення
Освітня програма	Інженерія програмного забезпечення комп'ютерних систем
Статус дисципліни	Нормативна
Форма навчання	очна(денна)
Рік підготовки, семестр	1 курс, весняний семестр
Обсяг дисципліни	1,5 кредита (45 годин)
Семестровий контроль/ контрольні заходи	Залік
Розклад занять	
Мова викладання	Українська
Інформація про керівника курсу / викладачів	Керівник: професор, д.т.н., Стіренко С.Г. sergii.stirenko@gmail.com Старший викладач кафедри обчислювальної техніки Шемсєдинов Т.Г. timur.shemsedinov@gmail.com
Розміщення курсу	https://github.com/HowProgrammingWorks

Програма навчальної дисципліни

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

Метою вивчення кредитного модуля «Методологія інженерії програмного забезпечення. Курсовий проєкт» є закріплення, поглиблення, узагальнення теоретичних знань та практичних навичок, які отримують студенти під час вивчення методології інженерії програмного забезпечення є підготовка фахівців, здатних розв'язувати комплексні проблеми в галузі дослідницько-інноваційної та практичної діяльності у сфері побудови та використанні нових ефективних масштабованих, відмово стійких архітектур для ПЗ комп'ютерних систем з високою продуктивністю.

Формування у студентів низки компетентностей, а саме:

- Здатність аналізувати предметні області, формувати, класифікувати вимоги до програмного забезпечення (ФК01).
- Здатність розробляти і реалізовувати наукові та/або прикладні проекти у сфері інженерії програмного забезпечення (ФК02)
- Здатність проектувати архітектуру програмного забезпечення, моделювати процеси функціонування окремих підсистем і модулів (ФК03)
- Застосування принципів проектування ПЗ побудови масштабованих програмних архітектур для комп'ютерних систем;
- Застосування різноманітних методів синтезу і аналізу ПЗ при проектуванні та дослідженні високонавантажених програмних застосунків.
- Розробка та аналіз різноманітних ефективних алгоритмів.
- Імплементация надійних та відмовостійких рішень при проектуванні ПЗ для комп'ютерних систем високої продуктивності.
- Відображення задач на основі організацій і алгоритми їх вирішення.
- Оцінки отриманих рішень.

Також після засвоєння навчальної дисципліни студенти мають продемонструвати такі програмні результати навчання:

- Виявляти інформаційні потреби і класифікувати дані для проектування програмного забезпечення (ПРН04);
- Розробляти і оцінювати стратегії проектування програмних засобів обґрунтовувати, аналізувати і оцінювати варіанти проектних рішень з точки зору якості кінцевого програмного продукту, ресурсних обмежень та інших факторів (ПРН06).
- обґрунтовано вибирати парадигми і мови програмування для розроблення програмного забезпечення; застосовувати на практиці сучасні засоби розроблення програмного забезпечення (ПРН09);
- збирати, аналізувати, оцінювати необхідну для розв'язання наукових і прикладних задач інформацію, використовуючи науково-технічну літературу, бази даних та інші джерела (ПРН17);

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

Дисципліни, що передують: Основи програмування, Компоненти програмної інженерії 1 (Вступ до програмної інженерії), Компоненти програмної інженерії 2 (Моделювання програмного забезпечення), Основи розробки програмного забезпечення на платформі Node.js, Системне програмування, Паралельні та розподілені обчислення, Гнучкі методи програмування, Моделювання програмного забезпечення.

Дисципліни, до яких готує цей курс: Наукова робота за темою магістерської дисертації. Частина 2, Практика, Робота над магістерською дисертацією.

3. Зміст навчальної дисципліни

- Тема 1. Стан додатків, структури даних та колекції
- Тема 2. Підходи до роботи зі станом: stateful and stateless
- Тема 3. Структури та записи
- Тема 4. Стек, черга, дек
- Тема 5. Дерева та графи
- Тема 6. Проекції та відображення наборів даних
- Тема 7. Оцінка обчислювальної складності
- Тема 8. Структура додатку: файли, модулі, компоненти
- Тема 9. Об'єкт, прототип та клас
- Тема 10. Залежності та бібліотеки
- Тема 11. Регулярні вирази
- Тема 12. Фабрики та пули
- Тема 13. І/О (введення-виведення) та файли
- Тема 14. Мономорфний та поліморфний код, інлайн-кеш, приховані класи
- Тема 15. Вимірювання продуктивності коду та оптимізація
- Тема 16. Асинхронне програмування на callback`ax
- Тема 17. Асинхронне програмування на promise`ax
- Тема 18. Асинхронні функції, async/await, thenable, обробка помилок
- Тема 19. Незмінні структури даних (immutable)
- Тема 20. Автоматне програмування: кінцеві автомати (машини станів)
- Тема 21. Шаблон Singleton (синглтон) JavaScript
- Тема 22. Функціональні об'єкти, функтори та монади
- Тема 23. Асинхронні генератори та асинхронні ітератори
- Тема 24. Перерахований тип (enum)

Етапи виконання курсового проєкту

Основні етапи виконання курсового проєкту:

- Отримання теми та завдання
- Підбор та вивчення літератури
- Формування технічного завдання
- Розробка розділів курсового проєкту та програмного додатка
- Тестування програмного додатку
- Оформлення пояснювальної записки
- Подання курсового проєкту (роботи) на перевірку
- Захист курсового проєкту (роботи)

4. Навчальні матеріали та ресурси

Базова:

1. Шемсєдинов Т.Г., Нечай Д.О., Кухар В.В., Орленко О.А., Голіков О.Г., Білочуб М.М., Духін В., Іванова Л.А., Чорненький А.Ю. та інші. Приклади коду та приклади проєктів [Електронний ресурс] розміщено за адресою:
<https://github.com/HowProgrammingWorks/>

2. Refactoring: Improving the Design of Existing Code // Martin Fowler
3. Clean Code: A Handbook of Agile Software Craftsmanship // Robert C. Martin
4. Introduction to Algorithms, 3rd Edition // Thomas H. Cormen
5. Gang of Four Design Patterns // Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides

Додаткова:

1. Шемсєдинов Т.Г., Нечай Д.О., Кухар В.В., Орленко О.А., Голіков О.Г., Білочуб М.М., Духін В., Іванова Л.А., Чорненький А.Ю. та інші. Технологічний стек Metarhia[Електронний ресурс] розміщено за адресою: <https://github.com/metarhia/>
2. Algorithms Unlocked // Thomas H. Cormen
3. The Art of Computer Programming // Donald Knuth
4. CodeComplete // Steve McConnell
5. Designing Object Oriented C++ Applications Using The Booch Method // Robert C. Martin
6. Extreme Programming Explained // Kent Beck
7. Analysis Patterns: Reusable Object Models // Martin Fowler

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

Виконання курсового проєкту складається з дев'яти етапів, які наведені в таблиці 5.1.

Таблиця 5.1

Тиждень семестру	Назва етапу роботи
3	<i>Отримання теми та завдання</i>
4-5	<i>Підбор та вивчення літератури</i>
6-7	<i>Формування технічного завдання</i>
8-13	<i>Розробка розділів курсового проєкту та програмного додатка</i>
14	<i>Тестування програмного додатку</i>
15	<i>Оформлення пояснювальної записки</i>
16	<i>Подання курсового проєкту (роботи) на перевірку</i>
17	<i>Захист курсового проєкту (роботи)</i>

6. Самостійна робота студента

№ з/п	Назва теми, що виноситься на самостійне опрацювання	Кількість годин СРС
1	<i>Отримання теми та завдання на курсовий проєкт</i>	1
2	<i>Аналіз завдання, підбір та вивчення літератури</i>	5
3	<i>Розробка розділів курсового проєкту та самого програмного додатка</i>	34
4	<i>Оформлення пояснювальної записки до курсового проєкту</i>	5
Разом		45

7. Політика навчальної дисципліни (освітнього компонента)

Оформлення курсового проєкту має відповідати вимогам до звітів про НДР (ДСТУ 3008-2015 «Державний стандарт України. Документація. Звіти в сфері науки і техніки. Структурі правила оформлення»).

Увесь ілюстративний матеріал у курсового проєкту повинен бути виконаний за допомогою комп'ютерних засобів. Зміст ілюстративного матеріалу має з достатньою повнотою відображати основні положення, які виносяться на захист.

Як викладач, так і студент зобов'язані дотримуватись Кодексу честі Національного технічного університету України «Київський політехнічний інститут».

Основні положення політики:

- тема курсового проєкту може бути узгодженою з темою майбутньої кваліфікаційної роботи бакалавра;
- етапи курсового проєкту повинні бути виконані згідно встановленого календарного графіку робіт;
- розроблений програмний додаток повинен бути протестований, результати тестування програмного додатку наводяться у тексті основної частини курсового проєкту;
- у випадку виявлення факту академічної не доброчесності та плагіату курсовий проєкт повертається на докорінну переробку з можливою зміною теми;
- невчасне виконання етапу курсового проєкту тягне за собою зниження отриманих за нього балів на 10%, якщо запізнення не більше двох тижнів, на 20% якщо запізнення більше двох тижнів.

При оцінюванні курсового проєкту беруться до уваги наступні чинники:

- повнота виконання індивідуального завдання на курсовий проєкт;
- коректність розроблених прецедентів;
- своєчасність виконання курсового проєкту згідно графіку;
- самостійність виконання курсового проєкту та відсутність ознак плагіату;
- відповіді на питання щодо змісту курсового проєкту під час її захисту.

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

Система оцінювання успішності студентів з освітнього компонента «Методологія інженерії програмного забезпечення. Курсовий проєкт» ґрунтується на «Положенні про систему оцінювання результатів навчання в КПІ ім. Ігоря Сікорського» (https://document.kpi.ua/files/2020_1-273.pdf), а саме Рейтингової системи оцінювання (PCO) другого типу (PCO-2). PCO-2 курсового проєкту (RK) складається з двох складових:

- стартової (RC);
- складова захисту (R3).

$$RK = RC + R3$$

Перша (стартова) складова характеризує роботу студента з курсового проєкту та її результат - якість пояснювальної записки та розробленого програмного додатку. Друга складова характеризує якість захисту студентом курсового проєкту.

Розмір шкали першої складової дорівнює **80 балам**, а другої складової - **20 балам**. **Якість пояснювальної записки та ступінь дотримання календарного графіку роботи** Ваговий бал – **80 (RC)**. Критерії оцінювання складових пояснювальної записки наведені в Таблиці 8.1.

Оскільки кредитний модуль має семестрову атестацію у вигляді заліку, рейтингова система оцінювання побудована за типом PCO – 1. Рейтинг студента з кредитного модуля складається з балів, які він отримує за види робіт відповідно до таблиці 8.1.

Таблиця 8.1 – Критерії оцінювання виконання складових пояснювальної записки

№ етапу	Складові роботи	Максимальна кількість балів при вчасному виконанні	Урахування вчасності виконання
1	Оформлення титульного аркушу	2	100% від оцінки при дотриманні графіку роботи
2	Наявність технічного завдання на КР	2	
3	Наявність та зміст опису альбому	2	
4	Наявність змісту	2	
5	Наявність та зміст ступу	2	
6	Наявність та зміст розробки всіх розділів завдання на курсовий проєкт.	45	90% у разі затримки до 2 тижнів
7	Наявність та зміст програмного додатку	10	
8	Наявність та зміст результатів тестування програмного додатку	5	80% у разі затримки більше 2 тижнів
9	Наявність та зміст висновків	5	
10	Наявність та оформлення переліку джерел	5	
Усього за семестр		80	

Студент допускається до захисту курсового проєкту за умови, він має стартову складову RC не менш ніж 60% від максимального значення, що складає **$80 \times 0.6 = 48$ балів**.

Якість захисту

Ваговий бал – **20 (R3)**.

Критерії оцінювання виступу з доповіддю за матеріалами КР та відповідей на питання:

- ступінь володіння теоретичним матеріалом до 10 балів;
- ступінь володіння методами розробки програмного додатку в цілому до 10 балів.

Захист курсового проєкту вважається успішним, якщо R3 становить не менш ніж 60% від свого максимального значення, тобто **$20 \times 0.6 = 12$ балів**.

Після завершення захисту курсового проєкту визначається **RK**, яка у подальшому переводиться у оцінку за університетською шкалою згідно таблиці:

Відповідність рейтингових балів оцінкам за університетською шкалою

<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
Не виконані умови допуску	Не допущено

Робочу програму навчальної дисципліни (силабус):

Склав професор кафедри обчислювальної техніки Стіренко С.Г.

Ухвалено кафедрою обчислювальної техніки (протокол № 10 від 25.05.2022)

Погоджено Методичною комісією факультету (протокол № 10 від 09.06.2022)