



Методологія інженерії програмного забезпечення. Курсовий проект. Робоча програма навчальної дисципліни (силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Другий (магістерський)</i>
Галузь знань	<i>12 Інформаційні технології</i>
Спеціальність	<i>121 Інженерія програмного забезпечення</i>
Освітня програма	<i>Інженерія програмного забезпечення інтелектуальних кібер-фізичних систем в енергетиці</i>
Статус дисципліни	<i>нормативна</i>
Форма навчання	<i>очна(денна)</i>
Рік підготовки, семестр	<i>1 курс, весняний</i>
Об'єм дисципліни	<i>1.5 кредити, 45 годин</i>
Семестровий контроль/ контрольні заходи	<i>залік,</i>
Розклад занять	<i>http://roz.kpi.ua/</i>
Мова викладання	<i>українська</i>
Інформація про керівника курсу / викладачів	<i>Лектор: к.т.н., доцент, Ходаковський Олексій Володимирович, dzgeron@ukr.net Лабораторні заняття: к.т.н., доцент, Ходаковський Олексій Володимирович, dzgeron@ukr.net</i>
Розміщення курсу	<i>https://campus.kpi.ua/</i>

Програма навчальної дисципліни

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

Метою вивчення кредитного модуля «Методологія інженерії програмного забезпечення. Курсовий проект» 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)
- Здатність розробляти і координувати процеси, етапи та ітерації життєвого циклу програмного забезпечення на основі застосування сучасних моделей, методів та технологій розроблення програмного забезпечення (ФК8);
- Здатність створювати та використовувати програмне забезпечення високопродуктивних комп'ютерних систем (ФК12)
- 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:

ПРН 1 Знати і застосовувати сучасні професійні стандарти і інші нормативно- правові документи з інженерії програмного забезпечення

ПРН 2 Оцінювати і вибирати ефективні методи і моделі розроблення, впровадження, супроводу програмного забезпечення та управління відповідними процесами на всіх етапах життєвого циклу.

ПРН 3 Будувати і досліджувати моделі інформаційних процесів у прикладній області.

ПРН 4 Виявляти інформаційні потреби і класифікувати дані для проектування програмного забезпечення.

ПРН 5 Розробляти, аналізувати, обґрунтовувати та систематизувати вимоги до програмного забезпечення.

ПРН 6 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.

ПРН 7 Аналізувати, оцінювати і застосовувати на системному рівні сучасні програмні та апаратні платформи для розв'язання складних задач інженерії програмного забезпечення.

ПРН 8 Розробляти і модифікувати архітектуру програмного забезпечення для реалізації вимог замовника.

ПРН 9 Обґрунтовано вибирати парадигми і мови програмування для розроблення програмного забезпечення; застосовувати на практиці сучасні засоби розроблення програмного забезпечення.

ПРН 10 Модифікувати існуючі та розробляти нові алгоритмічні рішення детального проектування програмного забезпечення.

ПРН 11 Забезпечувати якість на всіх стадіях життєвого циклу програмного забезпечення, у тому числі з використанням релевантних моделей та методів оцінювання, а також засобів автоматизованого тестування і верифікації програмного забезпечення.

ПРН 13 Конфігурувати програмне забезпечення, керувати його змінами та розробленням програмної документації на всіх етапах життєвого циклу.

ПРН 16 Планувати, організовувати та здійснювати тестування, верифікацію та валідацію програмного забезпечення.

ПРН 17 Збирати, аналізувати, оцінювати необхідну для розв'язання наукових і прикладних задач інформацію, використовуючи науково-технічну літературу, бази даних та інші джерела.

ПРН 20 Вміти проектувати та розробляти розподілені програмні системи на основі сучасної методології інженерії програмного забезпечення.

1. 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, Виконання магістерської дисертації.

2. 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. Деревя 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)

2. Етапи виконання курсового проєкту

Основні етапи виконання курсового проєкту:

- Отримання теми та завдання
- Підбір 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)

3. Навчальні матеріали та ресурси

Базова:

1. Шемсєдинов Т.Г., Нечай Д.О., Кухар В.В., Орленко О.А., Голиков 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., Чорнєнький 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

4. Методика опанування навчальної дисципліни (освітнього компонента)

Виконання курсового проєкту складається з дев'яти етапів, які наведені в таблиці 4.1.

Таблиця 4.1

Тиждень семестру	Назва етапу роботи
3	<i>Отримання теми та завдання</i>
4-5	<i>Підбір 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>

5. 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

Політика та контроль

6. 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. Структура і правила оформлення»).

Увесь ілюстративний матеріал у курсового проєкту повинен бути виконаний за допомогою комп'ютерних засобів. Зміст ілюстративного матеріалу має з достатньою повнотою відобразити 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 проекту;
 - у випадку виявлення факту академічної не добросовісності та плагіату курсовий проект повертається на докорінну переробку з можливою зміною теми;
 - невчасне виконання етапу курсового проекту 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 проекту беруться до уваги наступні чинники:
- повнота виконання індивідуального завдання на курсовий проект;
 - коректність розроблених прецедентів;
 - своєчасність виконання курсового проекту згідно графіку;
 - самостійність виконання курсового проекту та відсутність ознак плагіату;
 - відповіді на питання щодо змісту курсового проекту під час її захисту.

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

1. Рейтинг студента з кредитного модуля складається з балів, що він отримує за:

- захист курсового проекту;
- доповідь та відповіді на захисті курсового проекту.

2. Критерії нараховування балів:

2.1. Захист програмного забезпечення оцінюється в 60 балів і зараховується за наявності коректних рішень. Бали нараховуються таким чином:

- оптимальність алгоритму виведення (40 балів);
- оптимальність використаних структур подання інформації (20 балів);

Штрафні бали призначаються за:

- несвоєчасний захист програмного забезпечення – 6 балів;
- ненадану або невірну відповідь на запитання – 6 балів.

2.2. Доповідь на захисті курсового проекту оцінюється в 30 балів і зараховується за наявності коректних рішень. Бали нараховуються таким чином

- якість оформлення пояснювальної записки (15 балів);
- якість оформлення презентації (15 балів);

2.3. Відповідь на захисті курсового проекту оцінюється в 10 балів за такими критеріями:

- «відмінно», повна відповідь (не менше 90% потрібної інформації) – 10-9 балів;
- «добре», достатньо повна відповідь (не менше 75% потрібної інформації, або незначні неточності) – 8-7 балів;
- «задовільно», неповна відповідь (не менше 60% потрібної інформації та деякі помилки) – 6 балів;
- «незадовільно», відповідь не відповідає умовам на «задовільно» – 0 балів.

Сума балів, отриманих студентом, переводиться до підсумкової оцінки згідно з таблицею.

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

Кількість балів	Оцінка
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

Процедура оскарження результатів контрольних заходів

Студенти мають право і можливість підняти будь-яке питання, яке стосується процедури контрольних заходів та очікувати, що воно буде розглянуто згідно із наперед визначеними процедурами (детальніше: https://osvita.kpi.ua/2020_7-170, https://document.kpi.ua/files/2020_7-170.pdf).

Студенти мають право оскаржити результати контрольних заходів, але обов'язково аргументовано, пояснивши з яким критерієм не погоджуються відповідно до оціночного листа та/або зауважень.

Норми етичної поведінки студентів і працівників визначені у розділі 2 Кодексу честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» (детальніше: [https:// kpi.ua/code](https://kpi.ua/code)).

Робочу програму навчальної дисципліни (силабус): **Методологія інженерії програмного Забезпечення. Курсовий проєкт.**

Складено к.т.н., доцентом кафедри ІПЗЕ, Ходаковським Олексієм Володимировичем

Ухвалено кафедрою ІПЗЕ (протокол № 28 від 15.05.2023)

Погоджено Методичною комісією ННІАТЕ КПІ ім. Ігоря Сікорського ¹ (протокол № 9 від 26.05.2023 р.)