



ТЕХНОЛОГІЇ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	Перший (бакалаврський)
Галузь знань	12 Інформаційні технології
Спеціальність	122 Комп'ютерні науки
Освітня програма	Цифрові технології в енергетиці
Статус дисципліни	Нормативна
Форма навчання	Очна (денна)
Рік підготовки, семестр	3 курс осінній семестр
Обсяг дисципліни	4,5 кред./135 год. 36 лек., 18 лаб., 81 СРС
Семестровий контроль/ контрольні заходи	Екзамен, МКР
Розклад занять	https://schedule.kpi.ua/
Мова викладання	Українська
Інформація про керівника курсу / викладачів	Лектор: к.т.н., доцент каф. ЦТЕ, Тихоход Володимир Олександрович. Лабораторні роботи: к.т.н., доцент каф. ЦТЕ, Тихоход Володимир Олександрович, асистент, Пасічнюк Антон Олексійович
Розміщення курсу	https://campus.kpi.ua

Програма навчальної дисципліни

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

В результаті вивчення дисципліни у студентів повинні сформуватися наступні компетентності: загальні:

- ЗК 01 — здатність до абстрактного мислення, аналізу та синтезу.
- ЗК 02 — здатність застосовувати знання у практичних ситуаціях.
- ЗК 03 — знання та розуміння предметної області та розуміння професійної діяльності.
- ЗК 06 — здатність вчитися і оволодівати сучасними знаннями.
- ЗК 07 — здатність до пошуку, оброблення та аналізу інформації з різних джерел.
- ЗК 08 — здатність генерувати нові ідеї (креативність).
- ЗК 09 — здатність працювати в команді.
- ЗК 11 — здатність приймати обґрунтовані рішення.
- ЗК 13 — здатність діяти на основі етичних міркувань.

фахові компетентності:

- ФК 03 — здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв'язності та нерозв'язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем.

- ФК 08 — Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.

- ФК 10 — Здатність застосовувати методології, технології та інструментальні засоби для управління процесами життєвого циклу інформаційних і програмних систем, продуктів і сервісів інформаційних технологій відповідно до вимог замовника.

Очікувані програмні результати навчання:

- ПР 09 — Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук.

- ПР 11 — Володіти навичками управління життєвим циклом програмного забезпечення, продуктів і сервісів інформаційних технологій відповідно до вимог і обмежень замовника, вміти розробляти проектну документацію (техніко-економічне обґрунтування, технічне завдання, бізнес-план, угоду, договір, контракт).

- ПР 15 — Застосовувати знання методології та CASE-засобів проектування складних систем, методів структурного аналізу систем, об'єктно-орієнтованої методології проектування при розробці і дослідженні функціональних моделей організаційно-економічних і виробничо-технічних систем.

Після засвоєння навчальної дисципліни студенти мають продемонструвати такі знання та вміння:

ЗНАННЯ:

- інформаційних технологій, мов програмування, інструментарію програміста;
- сучасних технологій та інструментальних засобів розробки програмних систем;
- CASE-технологій проектування інформаційних та програмних систем;
- мов програмування, сучасних теорій організації баз даних та знань, методів і технологій їх розробки.
- методів та стандартів оформлення документації;
- міжнародних стандартів з оцінки якості програмного забезпечення (ISO 9126:2001, тощо), правил та методів забезпечення якості ІТ-систем;

ВМІННЯ:

- застосовувати мови програмування, мови опису інформаційних технологій, мови специфікацій;
- застосовувати інструментальні засоби при проектуванні та створенні інформаційних систем, продуктів і сервісів інформаційних технологій;
- володіти методами і засобами підтримки командної роботи.
- використовувати державні та міжнародні стандарти в галузі інформаційних технологій;
- застосовувати сучасні технології та інструментальні засоби розробки на всіх етапах життєвого циклу ІС;
- застосовувати CASE засоби;
- моделювати системи та процеси, стани та поведінки складних об'єктів інформатизації в процесі проектування інформаційних систем і технологій;
- володіти сучасними технологіями автоматизації проектування складних об'єктів і систем, продуктів і сервісів інформаційних технологій, сучасними парадигмами та мовами програмування;
- складати технічну документацію;
- застосовувати стандарти, методи та засоби управління процесами життєвого циклу розробки програмних систем;
- застосовувати у роботі міжнародні стандарти з оцінки якості програмного забезпечення.

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

Пререквізити дисципліни. Вивчення дисципліни спирається на знання, отримані в дисциплінах з попередніх семестрів, зокрема:

- «Алгоритмізація та програмування» (1, 2 семестри),
- «Об'єктно-орієнтоване програмування» (3 семестр),
- «Системи баз даних» (3 семестр),
- «Веб-технології та веб-дизайн» (3 семестр).

Постреквізити дисципліни. Матеріал дисципліни може бути використаний при вивченні дисциплін: «Вступ до інтелектуального аналізу даних», «Моделювання систем в енергетиці», «Технології паралельних обчислень в енергетичних комплексах», «Проектування інформаційних систем» та інших, що подаються в наступних семестрах, а також при дипломному проектуванні

3. Зміст навчальної дисципліни

1. Життєвий цикл і процеси розробки програмного забезпечення
2. Інженерія вимог до ПЗ
3. Проектування ПЗ.
4. Конфігурація ПЗ.
5. Архітектура ПЗ, стандарти опису архітектур ПЗ.
6. Тестування ПЗ.

4. Навчальні матеріали та ресурси

Основна література

1. Fowler M. UML Distilled: A Brief Guide to the Standard Object Modeling Language, Third Edition. – Addison-Wesley, 2003. – 208 pages.
2. Cockburn Alistair. Writing Effective Use Cases. Addison. — Addison-Wesley Professional, 2000. — 304 pages.
3. Fowler Martin. Patterns of Enterprise Application Architecture. — Addison-Wesley Professional, 2002. — 560 pages.
4. Лавріщева К.М. Програмна інженерія. — К. — 2008. — 319 с.
5. Мартін Роберт. Чиста архітектура: Мистецтво розроблення програмного забезпечення / пер. з англ. І. Бондар-Терещенко. — Харків : Вид-во «Ранок» : Фабула, 2019. — 368 с.

Додаткова література

1. Booch G. The Unified Modeling Language User Guide/ Booch G., Rumbaugh J., Jacobson I. — Second edition. — Addison-Wesley Professional, 2005. – 475 pages.
2. Wiegers Karl. Software Requirements (Developer Best Practices). — 3rd edition.— Microsoft Press; 2013. —672 pages.
3. Cohn Mike. User Stories Applied: For Agile Software Development. — Addison-Wesley Professional, 2004. — 304 с.
4. Pro Git. Режим доступу: <https://git-scm.com/book/uk/v2> .
5. Rubin Kenneth. Essential Scrum: A Practical Guide to the Most Popular Agile Process. — 1st edition — Addison-Wesley Professional, 2012. — 496 p.
6. McConnell Steve. Code Complete: A Practical Handbook of Software Construction.. — 2nd Edition. — Microsoft Press, 2004. — 960 pages.
7. Мартін Р. Чистий код: створення і рефакторинг за допомогою Agile / пер. з англ. І. Бондар-Терещенко. — Харків : Вид-во «Ранок» : Фабула, 2019. — 448 с.
8. Seemann Mark. Dependency Injection Principles, Practices, and Patterns. — Manning, 2019. — 552 pages.
9. Meszaros Gerard. xUnit Test Patterns: Refactoring Test Code. — Addison-Wesley Professional, 2007. — 833 pages.
10. Roy Oshero. The Art of Unit Testing: with examples in C#. — Manning, 2013. — 296 pages.

11. Beck Kent. Test Driven Development: By Example. — Addison-Wesley Professional, 2002. — 240 pages.
12. Wolff Eberhard. A Practical Guide to Continuous Delivery. — Addison-Wesley Professional, 2017. — 435 pages.
13. Gamma Erich. Design Patterns: Elements of Reusable Object-Oriented Software / Erich Gamma, Richard Helm, Ralph Johnson. — Addison-Wesley Professional, 1994. — 416 pages.
14. Braude E.J., Bernstein M.E. Software Engineering: Modern Approaches, 2 ed.: Wiley, 2011.
15. Sommerville I., Software Engineering, 10 ed.: Pearson, 2016.

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

Тема 1. Життєвий цикл і процеси розробки програмного забезпечення

Лекція 1. Моделі життєвого циклу програмного забезпечення

Технології розроблення програмного забезпечення. Програмний продукт. Життєвий цикл програмного забезпечення. Моделі життєвого циклу. Стратегії розробки ПЗ. Водоспадна, інкрементна, спіральна, компонентно-орієнтована модель ЖЦ. Макетування. Модель швидкої розробки (RAD-модель). V-модель (V-Model).

Лекція 2. Гнучкі методології розробки програмного забезпечення.

Важкі та полегшені процеси. Маніфест гнучкої розробки ПЗ. Порівняння гнучких процесів. Метод Канбан. Екстремальне програмування. Метод Скрам. Концепції розвитку Agile продукту. Макетний прототип. Підхід «ходячий скелет». Мінімально життєздатний продукт.

Лекція 3. Особливості процесу Scrum.

Загальна картина. Ролі. Артефакти процесу. Види діяльності.

Тема 2. Інженерія вимог до ПЗ

Лекція 4. Формування вимог до програмного забезпечення.

Методики формування вимог. Аналіз вимог до ПЗ. Вимоги до програмного забезпечення. Категорії вимог. Етапи інженерії вимог. Типи вимог. Формування вимог. Діаграма прецедентів. Аналіз вимог (класичний підхід). Характеристики якісних вимог. Опис вимог у вигляді текстових сценаріїв. Формування та аналіз вимог в процесі Scrum. Історія користувача. Критерії якісної історії користувача. Ієрархія історій. Критерії якісної задачі. Карта історій.

Лекція 5. Аналіз вимог до ПЗ.

Використання мови моделювання UML для аналізу вимог. Мова моделювання UML. Використання UML для аналізу вимог. Цілі використання UML. Режими використання UML. Типи діаграм UML. Діаграма діяльності. Діаграма станів.

Тема 3. Проектування ПЗ.

Лекція 6. Якість та надійність ПЗ.

Якість ПЗ. Види якості. Модель якості програмних систем. Модель Мак-Кола. Модель якості ISO/IEC 9126. Стандартні показники якості. Метрики якості.

Лекція 7. Проектування класів програмного забезпечення.

Проектування ПЗ з допомогою структурних та поведінкових діаграм UML. Діаграма класів.

Лекція 8. Принципи проектування програмного забезпечення.

Принципи проектування програмного забезпечення. Зв'язність та зв'язаність. Принципи SOLID. Принцип KISS. Принцип YAGNI. Принцип DRY. Принцип «Говори, а не питай».

Лекція 9. Використання діаграм UML при проектуванні структур та поведінки.

Діаграма об'єктів. Діаграма послідовності. Діаграма пакетів. Діаграма компонентів. Діаграма розгортання.

Лекція 10. Стандартні методи композиції об'єктів.

Інверсія залежності. Шаблони впровадження залежності: впровадження в конструктор, впровадження у властивість, впровадження в метод, навколишній контекст. Антишаблони впровадження залежності: диктатор, «Гібридне впровадження», «Обмежене конструювання», локатор сервісів.

Тема 4. Архітектура ПЗ, стандарти опису архітектур ПЗ

Лекція 11. Стандартні архітектурні шаблони та стилі.

Огляд. Поєднання архітектурних стилів. Архітектура клієнт/сервер. Компонентна архітектура. Об'єктно-орієнтована архітектура. Багатошарова архітектура. Архітектури на основі предметної області. Лукова архітектура. Гексагональна архітектура. Архітектура, основана на шині повідомлень. N-рівнева / 3-рівнева архітектура. Сервісно-орієнтована архітектура. Мікро-сервісна архітектура.

Лекція 12. Багатошарова архітектура базових типів застосунків.

Концепція розшарування. Архітектура базових типів застосунків. Особливості типових шарів. Рівень представлення. Бізнес логіка. Шар доступу до даних. Шар служб. Наскрізна функціональність. Чиста архітектура. Впорядкування коду в рамках чистої архітектури на прикладі .NET Core.

Лекція 13. Особливості реалізації багатошарової архітектури.

Особливості реалізації типових шарів. Рівень представлення. Шар служб. Шаблони проектування шару бізнес логіки. Шар доступу до даних. Наскрізна функціональність.

Тема 5. Керування конфігурацією

Лекція 14. Керування конфігурацією. Системи контролю версій

Системи контролю версій. Безперервна інтеграція. Про контроль версій. Типи СКВ. Основні поняття. Типовий порядок роботи з СКВ. Особливості GIT: принципи збереження даних, стани файлу в робочому каталозі, процеси роботи з гілками, робочий процес одного розробника, робочий процес взаємодії декількох розробників, хостинг репозиторіїв, GitHub.

Лекція 15. Технології неперервної інтеграції та неперервного розгортання.

Методика автоматизації технологічних процесів DevOps. Технології безперервної інтеграції. Технології безперервного розгортання. Життєвий цикл випуску програмного забезпечення та версії випусків. Ступені готовності випуску. Засоби безперервної інтеграції та безперервного розгортання.

Тема 6. Тестування програмного забезпечення

Лекція 16. Базові поняття тестування програмного забезпечення.

Контроль якості виробництва програмного забезпечення. Піраміда тестування. Види тестування. Тестова документація. Призначення модульного тестування. Переваги модульного тестування. Розробка через тестування.

Лекція 17. Основи модульного тестування.

Шаблон структуризації модульних тестів AAA. Стандарти іменування модульних тестів. Тестове покриття. Характеристики якісного модульного тесту. Загальні вимоги до модульних тестів. Використання тестових двійників.

6. Самостійна робота студента

Тема 1 Життєвий цикл і процеси розробки програмного забезпечення. Порівняння Agile-методологій.

Тема 2 Аналіз вимог замовника до ПЗ. Рівні деталізації сценаріїв за Алістером Коберном.

Тема 3 Проектування ПЗ. Діаграма потоків даних (data flow diagrams).

Тема 4 Архітектура ПЗ, стандарти опису архітектур ПЗ. Архітектура DDD.

Тема 5 Керування конфігурацією. Ознайомлення з основними командами командного рядка Git.

Тема 6 Тестування ПЗ. Особливості TDD.

7. Політика навчальної дисципліни (освітнього компонента)

Виконання лабораторних робіт є обов'язковою умовою. Для отримання допуску до іспиту студент повинен захистити всі лабораторні роботи. Це обов'язковим за винятком поважних причин (хвороби, форс-мажорних обставин).

В разі пропущення занять з поважних причин викладач надає можливість студенту виконати усі або деякі лабораторні завдання (винятком є виконання деяких завдань у зв'язку із закінченням навчального процесу).

В разі пропущення занять без поважних причин, а також через порушення граничного терміну виконання завдання (deadline) студент може отримати не більше 80% балів від максимальної оцінки відповідного завдання.

Протягом семестру студенти:

- виконують та захищають лабораторні роботи у відповідні терміни (на кожен роботу відводиться два тижні для здачі),
- пишуть модульну контрольну роботу,
- повинні отримати дві позитивні атестації,
- по закінченні навчального процесу складають іспит.

Політика та принципи академічної доброчесності визначені у розділі 3 Кодексу честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського». Детальніше: <https://kpi.ua/code>.

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>.

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

Система рейтингових (вагових) балів та критерії оцінювання

1) Лабораторні роботи

Кожна робота оцінюється максимально в 5 балів.

Для оцінки роботи використовується наступні критерії оцінювання:

- виконаний своєчасно (протягом двох тижнів з моменту видачі), у повному обсязі – відповідний бал згідно номеру лабораторної роботи;
- виконаний із запізненням – знімається 10 – 30% від максимальної кількості балів в залежності від терміну запізнення;
- виконаний не самостійно, із запізненням – знімається 50% від максимальної кількості балів;
- невиконаний протягом відведеного часу – 0 балів.

Максимальна кількість балів за усі виконані лабораторні роботи дорівнює $S_{max} = 45$ балів. Для підрахунку кількості набраних студентом за практиками балів протягом семестру використовується наступна формула:

$$r_{\text{прат}} = \frac{\sum_{i=1}^9 K\phi_i * v_i}{\sum_{i=1}^9 K_{max_i} * v_i} * S_{max}$$

де $K\phi_i$ — це отриманий (фактичний) бал за практикум під номером i , K_{max_i} — це максимальний бал за практикум i , v_i — ваговий коефіцієнт практикуму i .

Вагові коефіцієнти за комп'ютерні практики представлено в наступній таблиці.

№ з/п	Назва комп'ютерного практикуму	Ваговий коефіцієнт
1	Формування первинних вимог до ПЗ.	5
2	Аналіз вимог до ПЗ. Діаграма активностей та станів.	5
3	Проектування ПЗ. Діаграми класів та об'єктів.	5
4	Проектування ПЗ. Діаграми послідовностей та компонентів.	5
5	Заділ продукту.	5

6	Система контролю версій Git. GitHub.	5
7	Реалізація шару доступу до даних.	5
8	Реалізація шару бізнес-логіки.	5
9	Модульне тестування	5

Передбачено систему штрафних та заохочувальних балів:

активність на комп'ютерних практикумах	+ 1 бали
виконання комп'ютерного практикуму: - з використанням власного архітектурного рішення з дотриманням сучасних стандартів програмування; - що реалізація шаблону проектування, що не включено до основного матеріалу, але відноситься до предмету дисципліни; - виявлено всі можливі вимоги користувача до ПЗ при моделюванні вимог; - покриття модульними тестами виконано на високому рівні.	+ 0,5 бали
несвоєчасна здача комп'ютерного практикуму без поважної причини	-0,5 балів- -1 бал
обидві наскрізні атестації позитивні	+2 бал

2) Календарний контроль

Проводиться двічі на семестр як моніторинг поточного стану виконання вимог силабусу.

Для отримання «зараховано» з першої проміжної атестації студент повинен захистити 3 лабораторні роботи та отримати позитивну оцінку з тестування. Тестування проводиться в тиждень, що передує першій проміжній атестації (сьомий-восьмий тиждень навчання). Студент отримує бал, який відповідає частці правильних відповідей тесту з розрахунку, що максимальний бал за тест дорівнює $r_{\text{тест}} = 5$ балів, що становить 100%.

Для отримання «зараховано» з другої проміжної атестації студент повинен набрати не менше ніж 30 балів.

3) Модульна контрольна робота

Модульна контрольна робота (МКР) проводиться в кінці семестру з метою контролю оцінки отриманих знань.

Максимальна кількість балів за МКР дорівнює 10 балів.

Якість виконання роботи оцінюється за наступними критеріями:

- усі відповіді вірні та повні – 10 балів,
- у відповідях допущені несуттєві неточності – 8 балів,
- половина відповідей вірна – 5 балів,
- відповіді з суттєвими неточностями, але без критичних помилок – 2 бали,
- менше половини відповідей вірна – 0 балів.

Можливе також проведення модульної контрольної роботи у формі тестування, в такому випадку студент отримує бал, який відповідає частці правильних відповідей тесту з розрахунку, що 10 балів становлять 100%.

4) Складання іспиту

Умови допуску до іспиту

Необхідною умовою допуску до іспиту є зарахування всіх комп'ютерних практикумів та виконання модульної контрольної роботи, а також стартовий рейтинг (R_c) не менше 40 балів.

Максимальний ваговий бал $r_{\text{ісп}}=40$

На іспиті студент виконує письмову контрольну роботу, яка містить два теоретичних питання і одне практичне питання. Теоретичні питання оцінюються максимально по 10 балів, практичне – 20 балів.

Теоретична частина оцінюється таким чином:

- правильна чітко викладена, повна відповідь – (не менше 90% потрібної інформації) – 9-10 балів;

- достатньо повна відповідь (не менше 75% потрібної інформації) – 7.5-8.9 балів;

- неповна відповідь (не менше 60% потрібної інформації) – 6-7.4 балів;

- незадовільна відповідь – 0 балів.

Практичне завдання оцінюється таким чином:

- повне, безпомилкове розв'язування завдання – 14-15 балів;

- повне, розв'язування завдання із несуттєвими невідповідностями – 11-13 балів;

- завдання виконане з певними недоліками – 9-10 балів;

- завдання не виконано – 0 балів.

5) Розрахунок шкали (R) рейтингу:

Сума вагових балів контрольних заходів протягом семестру (шкала рейтингу) складає:

$$R = r_{\text{практ}} + r_{\text{тест}} + r_{\text{мод}} + r_{\text{ісп}} = 50 + 5 + 10 + 40 = 100 \text{ балів.}$$

Максимальний стартовий рейтинг становить $R_c = r_{\text{практ}} + r_{\text{тест}} + r_{\text{мод}} = 60$ балів.

Рейтинг іспиту дорівнює 40 балів. Мінімальний рейтинг допуску до іспиту становить 40 балів.

Таким чином, рейтингова шкала з кредитного модуля складає

$$R = 60 + 40 = 100 \text{ балів.}$$

Для отримання студентом відповідних оцінок рейтингова оцінка студента переводиться згідно таблиці:

Бали	Оцінка
95 - 100	Відмінно
85 - 94	Дуже добре
75 - 84	Добре
65 - 74	Задовільно
60 - 64	Достатньо
Менше 60	Незадовільно
$R < 40$ є незараховані роботи комп'ютерного практикуму або не виконані інші умови допуску до екзамену	Не допущено

Робочу програму навчальної дисципліни (силабус):

Складено к.т.н., доцент каф. ЦТЕ, Тихоход Володимир Олександрович

Ухвалено кафедрою ЦТЕ (протокол № 22 від 25.06.25)

Погоджено Методичною комісією НН ІАТЕ КПІ ім. Ігоря Сікорського (протокол № 9 від 27.06.25)