Національний технічний університет України
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Кафедра цифрових технологій в енергетиці

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	3,5 кред./105 год. 36 лек., 18 лаб. 51 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 Комп'ютерні практикуми: к.т.н., доцент каф. ЦТЕ, Тихоход Володимир Олександрович
Розміщення курсу	https://campus.kpi.ua

Програма навчальної дисципліни

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

В результаті вивчення дисципліни у студентів повинні сформуватися наступні компетентності:

загальні:

- ЗК1 — здатність до абстрактного мислення, аналізу та синтезу.
- ЗК2 — здатність застосовувати знання у практичних ситуаціях.
- ЗК3 — знання та розуміння предметної області та розуміння професійної діяльності.
- ЗК 6 —здатність вчитися і оволодівати сучасними знаннями.
- ЗК7 — здатність до пошуку, оброблення та аналізу інформації з різних джерел.
- ЗК8 — здатність генерувати нові ідеї (креативність).
- ЗК9 — здатність працювати в команді.
- ЗК13 — здатність діяти на основі етичних міркувань.

фахові компетентності:

- ФК 3 — 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

професійні:

- ПР1 — Застосовувати знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу, обробки та синтезу інформації в предметній області комп'ютерних наук.

- ПР9 — 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).
- ПР14 — Використовувати мережні технології, архітектури комп'ютерних мереж, мати практичні навички технології адміністрування комп'ютерних мереж та їх програмного забезпечення

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: "Основи програмування та алгоритмічні мови", «Об'єктно-орієнтоване програмування», «Системи баз даних», «Проектування та використання баз даних».

Постреквізити дисципліни. Матеріал дисципліни може бути використаний при вивченні дисциплін «Технології комп'ютерного проектування», «Програмування комп'ютерних мереж», «Геометричне моделювання та комп'ютерна графіка», "Технології розподілених систем та паралельних обчислень", «Теорія прийняття рішень», «Управління ІТ-проектами» та інших, що подаються в наступних семестрах.

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

6. Booch G. The Unified Modeling Language User Guide/ Booch G., Rumbaugh J., Jacobson I. — Second edition. — Addison-Wesley Professional, 2005. – 475 pages.
7. Wiegers Karl. Software Requirements (Developer Best Practices). — 3rd edition.— Microsoft Press; 2013. —672 pages.
8. Cohn Mike. User Stories Applied: For Agile Software Development. — Addison-Wesley Professional, 2004. — 304 с.
9. Pro Git. Режим доступу: https://git-scm.com/book/uk/v2 .
10. Rubin Kenneth. Essential Scrum: A Practical Guide to the Most Popular Agile Process. — 1st edition — Addison-Wesley Professional, 2012. — 496 p.
11. McConnell Steve. Code Complete: A Practical Handbook of Software Construction.. — 2nd Edition. — Microsoft Press, 2004. — 960 pages.
12. Мартін Р. Чистий код: створення і рефакторинг за допомогою Agile / пер. з англ. І. Бондар-Терещенко. — Харків : Вид-во «Ранок» : Фабула, 2019. — 448 с.
13. Seemann Mark. Dependency Injection Principles, Practices, and Patterns. — Manning, 2019. — 552 pages.
14. Meszaros Gerard. xUnit Test Patterns: Refactoring Test Code. — Addison-Wesley Professional, 2007. — 833 pages.
15. Roy Osherove. The Art of Unit Testing: with examples in C#. — Manning, 2013. — 296 pages.
16. Beck Kent. Test Driven Development: By Example. — Addison-Wesley Professional, 2002. — 240 pages.
17. Wolff Eberhard. A Practical Guide to Continuous Delivery. — Addison-Wesley Professional, 2017. — 435 pages.
18. Gamma Erich. Design Patterns: Elements of Reusable Object-Oriented Software / Erich Gamma, Richard Helm, Ralph Johnson. — Addison-Wesley Professional, 1994. — 416 pages.

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

Тема 1. Життєвий цикл і процеси розробки програмного забезпечення

Лекція 1. Моделі життєвого циклу ПЗ.

Технології розроблення програмного забезпечення. Програмний продукт. Життєвий цикл програмного забезпечення. Моделі життєвого циклу. Стратегії розробки ПЗ. Водоспадна модель

(Waterfall model) ЖЦ. Інкрементна (incremental) модель. Макетування. Спіральна модель. Компонентно-орієнтована модель. RAD-модель. V- модель (V-Model).

Лекція 2. Гнучкі методології.
Важкі та полегшені процеси. Agile-маніфест. Методологія Kanban. Методологія XP (Extreme Programming). Методологія Scrum. Порівняння Agile-методологій

Лекція 3. Методологія Scrum.
Ролі (дійові особи). Артефакти. Види діяльності.

Тема 2. Інженерія вимог до ПЗ
Лекція 4. Інженерія вимог до ПЗ. 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 (user story). Критерії якісної історії користувача. Ієрархія історій. Критерії якісної задачі. Карта історій.

Лекція 5. Аналіз вимог. 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. Поняття чистого коду. Стандарти створення чистого коду: змістовні імена, стандарти написання функцій, коментарів, класів. Оброблення помилок.

Лекція 7. Проектування програмного забезпечення.
Проектування ПЗ з допомогою структурних та поведінкових діаграм UML. Діаграма класів. Діаграма об'єктів.

Лекція 8. Проектування програмного забезпечення.
Діаграма послідовності (Sequence diagram). Діаграма пакетів (Package Diagram). Діаграма компонентів (Component Diagram). Діаграма розгортання (Deployment Diagram).

Лекція 9. Принципи проектування програмного забезпечення. Шаблони DI.
Написання якісного коду. Принципи проектування програмного забезпечення. Зв’язність та зв’язаність. Шаблони інверсії залежностей. Шаблони DI: впровадження в конструктор, впровадження у властивість, впровадження в метод, навколишній контекст (Ambient Context). Принципи SOLID. Принцип KISS. Принцип YAGNI. Принцип DRY. Принцип «Говори, а не питай».

Лекція 10. Знайомство з патернами проектування.
Породжуючі, структурні та поведінкові шаблони проектування. Factory Method. Абстрактна фабрика (Abstract Factory). Одинак (Singleton). Стратегія (Strategy). Спостерігач (Observer). Стан (State). Ланцюжок Обов'язків (Chain of responsibility).

Тема 4. Архітектура ПЗ, стандарти опису архітектур ПЗ

Лекція 11. Архитектурні шаблони та стилі.

Архітектура клієнт/сервер. Компонентна архітектура. Проектування на основі предметної області. Багатошарова архітектура (N-layer architecture). Лукова архітектура (Onion Architecture). Гексагональна архітектура (Hexagonal Architecture). Архітектура, основана на шині повідомлень. N-рівнева / 3-рівнева архітектура (N-tier architecture). Об’єктно-орієнтована архітектура. Сервісно-орієнтована архітектура (Service-Oriented Architecture). Мікро-сервісна архітектура.

Лекція 12. Багатошарова архітектура
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.

Тема 5. Керування конфігурацією
Лекція 13. Керування конфігурацією. 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 14. Безперервна інтеграція. Практики DevOps
Концепція безперервної інтеграції. Автоматизація складання. Гаки (hooks) в системах контролю версій. Інструменти складання. Засоби безперервної інтеграції. Концепція безперервного розгортання. Засоби безперервного розгортання. Практики DevOps.

Тема 6. Тестування ПЗ
Лекція 15. Тестування.
Значення тестування. Види тестування. Тестова документація.

Лекція 16. Модульне тестування.
Призначення. Переваги. Характеристики якісного теста. Тестові двійники: об'єкт-заглушка (Dummy Object), тестовая заглушка (Test Stub), тестовий агент (Test Spy), імітація (Fake), підставний об'єкт (Mock). TDD. Фреймворки тестування. Підтримка модульного тестування на рівні мови. Рекомендації.

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 конфігурацію
Ознайомлення з основними командами командного рядка Git.

Тема 6 Тестування ПЗ
Особливості TDD.

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

Відвідування лекційних та лабораторних занять є 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 комп'ютерні практикуми у відповідні терміни (на кожну практичну 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 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 (РСО)

Система рейтингових (вагових) балів та критерії оцінювання

1) Комп'ютерні практикуми

Кожний практикум оцінюється максимально в 5 балів.

Для оцінки практикуму використовується наступні критерії оцінювання:

- виконаний своєчасно (протягом двох тижнів з моменту видачі), у повному обсязі – відповідний бал згідно номеру комп'ютерного практикуму;
- 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 комп'ютерні практикуми дорівнює $S_{max} = 50$ балів. Для підрахунку кількості набраних студентом за практикуми балів протягом семестру використовується наступна формула:

$$r_{\text{практ}} = \frac{\sum_{i=1}^{13} K\text{ф}_i * v_i}{\sum_{i=1}^{13} Kmax_i * v_i} * S_{max}$$

де $K\text{ф}_i$ — це отриманий (фактичний) бал за практикум під номером *i*, $K\text{м}_i$ — це максимальний бал за практикум *i*, v_i — ваговий коефіцієнт практикуму *i*.

Вагові коефіцієнти за комп'ютерні практикуми 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.	4
7	Реалізація шару доступу до даних.	6
8	Реалізація шару бізнес-логіки.	8
9	Модульне тестування	7

2) Модульна контрольна робота

Максимальна кількість балів за модульну контрольну роботу дорівнює 10 балів.

Якість виконання роботи:

- 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.

Штрафні та заохочувальні бали за:	
- активність на комп'ютерних практикумах	+ 2 бали
- виконання комп'ютерного практикуму з використанням власного оптимального алгоритму	+ 1 бали
- відсутність на занятті без поважної причини	– 2 бали
- несвоєчасна здача комп'ютерного практикуму (пізніше ніж за тиждень)	– 0,5 балів;

3) Складання іспиту

Максимальний ваговий бал $r_{ісп}$=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.

Умови позитивної проміжної атестації

Для отримання „зараховано" з першої проміжної атестації студент повинен захистити 3 практикуми.

Для отримання „зараховано" з другої проміжної атестації студент повинен набрати не менше ніж 22 балів.

Умови допуску до *іспиту*

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

Розрахунок шкали (R) рейтингу:

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

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

Максимальний стартовий рейтинг становить $R_c = r_{практ} + r_{мод} =$ 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 доцент, к.т.н., доцент каф. ЦТЕ,Тихоход Володимир Олександрович

Ухвалено кафедрою ЦТЕ (протокол № 20 від 10.05.23)

Погоджено Методичною комісією НН ІАТЕ КПІ ім. Ігоря Сікорського (протокол № 9 від 26.05.23)