



## Силабус освітнього компонента

Програма навчальної дисципліни



# Проектування програмних систем

**Шифр та назва спеціальності**  
122 – Комп'ютерні науки

**Інститут**  
ННІ Комп'ютерного моделювання, прикладної фізики та математики

**Освітня програма**  
Комп'ютерні науки. Моделювання, проектування та комп'ютерна графіка

**Кафедра**  
Комп'ютерне моделювання процесів та систем (162)

**Рівень освіти**  
Бакалавр

**Тип дисципліни**  
Профільований пакет 1, Вибіркова

**Семестр**  
5

**Мова викладання**  
Українська

## Викладачі, розробники



**Іванченко Ксенія Вікторівна**

[kseniia.ivanchenko@khp.edu.ua](mailto:kseniia.ivanchenko@khp.edu.ua)

Кандидат технічних наук, доцент, доцент кафедри комп'ютерного моделювання процесів та систем

Автор та співавтор понад 30 наукових та методичних публікацій. Курси: "Проектування програмних систем", "Кросплатформене програмування", "Аналіз та тестування програмних систем", "Програмування мобільних пристроїв".

[Детальніше про викладача на сайті кафедри](#)

## Загальна інформація

### Анотація

Дисципліна "Проектування програмних систем" розглядає ключові аспекти розробки програмного забезпечення та ІТ-проектів. Курс охоплює широкий спектр тем, включаючи стан та перспективи ІТ-галузі в Україні, методології розробки, Agile підходи, інструменти для управління проектами, а також концепції проектування архітектури, управління конфігурацією, тестування та безпеку програмного забезпечення. Студенти отримають поглиблені знання щодо важливих аспектів програмного проектування, включаючи практичні навички використання інструментів, таких як Jira та UML-діаграми. Курс дозволяє студентам зрозуміти сучасні тенденції в галузі, вибирати оптимальні методології розробки та ефективно керувати проектами.

### Мета та цілі дисципліни

Сформувані у студентів знання, необхідні для усвідомлення і раціонального використання понять у галузі проектування програмних систем, вказати шляхи використання методів проектування на практиці; виробити навички розв'язання основних задач проектування та аналізу складних програмних систем; засвоєння майбутніми фахівцями знань, умінь та навичок, необхідних для самостійного проектування програмних систем, а також їх самостійного оновлення.

## **Формат занять**

Лекції, лабораторні роботи, самостійна робота, консультації. Підсумковий контроль – залік.

## **Компетентності**

ЗК2: Здатність застосовувати знання у практичних ситуаціях

ЗК3: Знання та розуміння предметної області та розуміння професійної діяльності

ЗК12: Здатність оцінювати та забезпечувати якість виконуваних робіт

СК8: 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

СК9: Здатність реалізувати багаторівневу обчислювальну модель на основі архітектури клієнт-сервер, включаючи бази даних, знань і сховища даних, виконувати розподілену обробку великих наборів даних на кластерах стандартних серверів для забезпечення обчислювальних потреб користувачів, у тому числі на хмарних сервісах

СК10: 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

СК15: Здатність до аналізу та функціонального моделювання бізнес-процесів, побудови та практичного застосування функціональних моделей організаційно-економічних і виробничо-технічних систем, методів оцінювання ризиків їх проектування

## **Результати навчання**

ПР5: Проектувати, розробляти та аналізувати алгоритми розв'язання обчислювальних та логічних задач, оцінювати ефективність та складність алгоритмів на основі застосування формальних моделей алгоритмів та обчислюваних функцій

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

ПР10: Використовувати інструментальні засоби розробки клієнт-серверних застосувань, проектувати концептуальні, логічні та фізичні моделі баз даних, розробляти та оптимізувати запити до них, створювати розподілені бази даних, сховища та вітрини даних, бази знань, у тому числі на хмарних сервісах, із застосуванням мов веб-програмування

ПР11: 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: 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

## **Обсяг дисципліни**

Загальний обсяг дисципліни 120 год. (4 кредитів ECTS): лекції – 32 год., лабораторні роботи – 16 год., самостійна робота – 72 год.

## **Передумови вивчення дисципліни (пререквізити)**

Дисципліна базується на знаннях та компетенціях, що набуває здобувач вищої освіти під час вивчення дисциплін: Алгоритмізація та програмування, Об'єктно-орієнтоване програмування та проектування, Технології програмування.

## Особливості дисципліни, методи та технології навчання

Лекції проводяться інтерактивно з використанням мультимедійних технологій. На заняттях використовується проєктний підхід до навчання. Навчальні матеріали доступні студентам через OneNote Class Notebook.

## Програма навчальної дисципліни

### Теми лекційних занять

#### Тема 1. Вступ

Програмування та програмне забезпечення. Причини невдач ІТ-проєктів.

#### Тема 2. Стан та перспективи розвитку ІТ-галузі в Україні: виклики та можливості

Сучасний стан ІТ галузі в Україні: розмір ринку, тенденції зайнятості, інноваційні рішення та перспективи розвитку. Ролі українських ІТ-компаній на світовому ринку. Внутрішні фактори, які впливають на розвиток галузі. Важливі сфери розвитку. Основні проблеми та шляхи їх вирішення.

#### Тема 3. Концептуальне, логічне, фізичне проєктування. Інструменти моделювання в проєктуванні програмних систем

Використання UML-діаграм для моделювання системи. Класифікація та використання діаграм (класів, взаємодій, станів тощо).

#### Тема 4. Традиційні методології розробки програмного забезпечення.

Порівняння методологій розробки, які використовують лінійний підхід до процесу розробки: Waterfall, V-Model та Spiral методології. Вибір оптимальної методології для конкретного проєкту. Можливості інтеграції Agile елементів у традиційні процеси. Використання гібридних підходів для оптимізації розробки ПЗ.

#### Тема 5. Сучасні Agile методології в розробці програмного забезпечення: Scrum, Kanban, та Lean

Сучасні підходи до розробки програмного забезпечення, які базуються на принципах Agile: Scrum, Kanban та Lean. Визначення основних складових Scrum. Ролі в Scrum. Принципи роботи з ітераціями (Sprints) та планування релізів. Відстеження та оптимізація ефективності робочого процесу за допомогою метрик Kanban. Концепція постійного вдосконалення (Kaizen) в Lean розробці.

#### Тема 6. Інструменти для реалізації Agile методологій

Використання онлайн-інструментів для управління Agile проєктами (наприклад, Jira, Trello, Asana). Створення та відстеження задач, складання звітів та аналіз результатів.

#### Тема 7. Вимоги та аналіз в проєктуванні програмних систем

Збір та документування вимог до системи. Методи аналізу вимог та їх перевірка. Визначення функціональних та нефункціональних вимог.

#### Тема 8. Проєктування архітектури програмних систем

Основні принципи проєктування архітектури. Визначення компонентів та їх взаємодія. Моделювання архітектурних рішень.

#### Тема 9. Управління конфігурацією та версіонування програмного забезпечення

Засади управління конфігурацією. Використання систем контролю версій. Керування змінами та релізами.

#### Тема 10. Оптимізація та швидкість розробки

Засоби оптимізації програмного коду. Принципи швидкої розробки та ітераційного підходу. Застосування best practices в програмуванні.

#### Тема 11. Інтеграція та впровадження програмних систем

Принципи інтеграції різних компонентів системи. Способи тестування та валідації під час інтеграції. Підготовка та впровадження продукту.

#### Тема 12. Патерни проєктування в програмуванні

Знайомство з основними патернами проєктування. Використання патернів для вирішення типових задач. Архітектурні та структурні патерни.

#### Тема 13. Тестування програмного забезпечення: стратегії та методи

Визначення стратегій тестування. Основні методи тестування: чорний ящик, білий ящик, сірий ящик. Автоматизація тестування та використання тестових фреймворків.

#### Тема 14. Управління якістю в проєктуванні програмних систем

Визначення критеріїв якості програмного продукту. Автоматизація процесів контролю якості.

Практичні аспекти тестування та валідації якості.

### Тема 15. Технологічні тренди в ІТ-галузі

Аналіз новітніх технологій та їх вплив на проектування програмних систем. Штучний інтелект, блокчейн, Інтернет речей (IoT), тощо. Можливості та виклики використання нових технологій.

### Тема 16. Завдання та висновки: реальний проект програмного забезпечення

Розгляд реального проекту в галузі програмного забезпечення. Вивчення проблем, які виникли під час розробки та їх вирішення. Підсумки та рекомендації щодо подальшого вдосконалення навичок у проектуванні програмних систем.

## Теми практичних занять

Не передбачено навчальним планом.

## Теми лабораторних робіт

### Лабораторна робота 1.

Причини невдач ІТ-проектів: аналіз та висновки.

### Лабораторна робота 2.

Створення технічного завдання.

### Лабораторна робота 3.

Створення ескізного проекту.

### Лабораторна робота 4.

Створення технічного проекту.

### Лабораторна робота 5.

Управління проектами Scrum та використання Workflow в Jira.

### Лабораторна робота 6.

Управління проектами Kanban в Jira.

### Лабораторна робота 7.

Презентація та захист індивідуальних розрахункових завдань.

### Лабораторна робота 8.

Підсумкове тестування.

## Самостійна робота

Курс передбачає виконання індивідуального розрахункового завдання у вигляді створення повнофункціонального проекту в Jira або аналогічному сервісі для конкретного програмного продукту. Студент повинен врахувати основні етапи життєвого циклу розробки, встановити задачі, спроектувати схему взаємодії між завданнями, планувати та відстежувати виконання робіт, використовуючи різні функціональності Jira.

Результат розробки оформлюється у презентацію, з обов'язковим захистом в кінці курсу. Оцінка виставляється за результатом відкритого захисту розрахункового завдання перед комісією, яка складається з викладача та інших студентів групи.

Студентам також рекомендуються додаткові матеріали (відео, статті) для самостійного вивчення та аналізу.

## Література та навчальні матеріали

1 Martin R.C. Clean Code: A Handbook of Agile Software Craftsmanship, 2008.

2 Cohn M. Agile Estimating and Planning, 2005.

3 Швець О. Занурення в рефакторинг, Refactoring.Guru, 2019.

4 Kim G., Humble J., Debois P., Willis J. The DevOps Handbook: How to Create World-Class Agility, Reliability, & Security in Technology Organizations, 2016

5 Forsgren N., Humble J., Kim G. Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations, 2018.

6 Aniche M. Effective software testing: A developer's guide, 2022.

7 Авраменко А.С., Авраменко В.С., Косенюк Г.В. Тестування програмного забезпечення.

Навчальний посібник. – Черкаси: ЧНУ імені Богдана Хмельницького, 2017. – 284 с

8 Марченко А.В. Проектування інформаційних систем, 2015р.

## Система оцінювання

### Критерії оцінювання успішності студента та розподіл балів

100% підсумкової оцінки складаються з результатів оцінювання у вигляді підсумкового тестування (20%) та поточного оцінювання (80%).

*Підсумкове тестування:* 40 тестових питань, наявні питання з декількома відповідями та відкриті питання.

*Поточне оцінювання:* виконання лабораторних робіт (60%) та розрахункове завдання (20%). У разі непогодження з отриманою оцінкою 100% підсумкової оцінки складаються з результатів оцінювання у вигляді екзамену. Обов'язковою умовою допуску до екзамену є виконання всіх лабораторних робіт.

*Екзамен:* письмове завдання (2 запитання з теорії та практичне завдання) та усна доповідь.

### Шкала оцінювання

Сума балів	Національна оцінка	ECTS
90–100	Відмінно	A
82–89	Добре	B
75–81	Добре	C
64–74	Задовільно	D
60–63	Задовільно	E
35–59	Незадовільно (потрібне додаткове вивчення)	FX
1–34	Незадовільно (потрібне повторне вивчення)	F

## Норми академічної етики і політика курсу

Студент повинен дотримуватися «Кодексу етики академічних взаємовідносин та доброчесності НТУ «ХПІ»: виявляти дисциплінованість, вихованість, доброзичливість, чесність, відповідальність. Конфліктні ситуації повинні відкрито обговорюватися в навчальних групах з викладачем, а при неможливості вирішення конфлікту – доводитися до відома співробітників дирекції інституту. Нормативно-правове забезпечення впровадження принципів академічної доброчесності НТУ «ХПІ» розміщено на сайті: <http://blogs.kpi.kharkov.ua/v2/nv/akademichna-dobrochesnist/>

## Погодження

Силабус погоджено

28.08.2023

Завідувач кафедри  
Дмитро БРЕСЛАВСЬКИЙ

28.08.2023

Гарант ОП  
Оксана ТАТАРІНОВА