



# МЕТОДИ РЕІНЖЕНІРИНГУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## Робоча програма навчальної дисципліни (Силабус)

### Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Доктора філософії</i>
Галузь знань	<i>12 Інформаційні технології</i>
Спеціальність	<i>121 Інженерія програмного забезпечення</i>
Освітня програма	<i>Інженерія програмного забезпечення</i>
Статус дисципліни	<i>Нормативна</i>
Форма навчання	<i>Очна (денна, вечірня), заочна</i>
Рік підготовки, семестр	<i>2 курс, весняний семестр</i>
Обсяг дисципліни	<i>4 кредити ЕКТС (120 годин)</i>
Семестровий контроль/ контрольні заходи	<i>Екзамен, модульна контрольна робота, календарний контроль</i>
Розклад занять	<i>Згідно розкладу на весняний семестр поточного навчального року (rozklad.kpi.ua)</i>
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	Викладачі: <i>к.т.н., старший викладач, Гречко Анастасія Валеріївна, grechna@gmail.com</i> <i>к.т.н., старший викладач, Хіцько Яна Володимирівна, iana.khitsko@gmail.com</i>
Розміщення курсу	<i>Google Classroom, Електронний кампус</i>

### Програма навчальної дисципліни

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

*Вивчення дисципліни «Методи реінженірингу програмного забезпечення» дозволяє сформувати у здобувачів освіти компетенції, необхідні для розв'язання практичних задач наукової діяльності, пов'язаної із роботою з існуючим програмним забезпеченням, його вдосконаленням та масштабуванням для задоволення сучасних потреб супроводження такого програмного забезпечення.*

***Предметом** дисципліни «Методи реінженірингу програмного забезпечення» є математичне та алгоритмічне забезпечення процесів переробки вихідного коду.*

***Метою** навчальної дисципліни є формування у здобувачів вищої освіти таких компетентностей:*

- Здатність розробляти якісне та надійне програмне забезпечення складних програмних комплексів та систем на основі новітніх технологій та стандартів розроблення програмного забезпечення.*
- Здатність проводити експериментальні дослідження з оцінювання ефективності та безпечності програмного забезпечення.*
- Здатність самостійно виконувати науково-дослідну діяльність в інженерії програмного забезпечення із застосуванням сучасних концепцій, методів, та технологій.*
- Здатність критично переосмислювати наявні технології інженерії програмного забезпечення та відстежувати тенденції їх розвитку.*

- Здатність креативно (творчо) мислити, генерувати нові прогресивні ідеї в інженерії програмного забезпечення.
- Здатність розширювати межі знань використовуючи результати оригінальних досліджень.
- Здатність забезпечувати безперервний саморозвиток і самовдосконалення, відповідальність за розвиток інших у професійній галузі, дотримуючись педагогічної етики, правил академічної доброчесності у науково-педагогічній діяльності.
- Здатність використовувати адекватні методи ефективної взаємодії з представниками різних груп (соціальних, культурних і професійних).
- Здатність працювати в команді, формувати позитивні відносини з колегами, спілкуватися з широкою науковою спільнотою та громадськістю в сфері інженерії програмного забезпечення.

В результаті вивчення навчальної дисципліни здобувачі вищої освіти набудуть таких програмних результатів навчання:

- Знати методи реінжинірингу програмного забезпечення.
- Уміти застосовувати інструментальні засоби проведення рефакторингу програмного забезпечення.
- Уміти застосовувати методи поетапного модифікування програмного забезпечення.
- Вміти формулювати та вирішувати задачі оптимізації, адаптації, прогнозування, керування та прийняття рішень щодо процесів, засобів та ресурсів розроблення, впровадження, супроводу та експлуатації програмного забезпечення.
- Мати передові концептуальні та методологічні знання з інженерії програмного забезпечення і на межі предметних галузей, а також дослідницькі навички, достатні для проведення наукових і прикладних досліджень на рівні сучасних світових досягнень з відповідного напрямку, отримання нових знань та/або здійснення інновацій.
- Глибоко розуміти загальні принципи та методи інженерії програмного забезпечення, а також методологію наукових досліджень, застосувати їх у власних дослідженнях у сфері інженерії програмного забезпечення та у викладацькій практиці.
- Вміти досліджувати робочі параметри процесів життєвого циклу програмного забезпечення, а також здійснювати аналіз вибраних методів та засобів підтримки цих процесів та бути спроможним обґрунтувати свій вибір.

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

Успішному вивченню дисципліни «Методи реінженірингу програмного забезпечення» передують вивчення дисциплін «Модельно-орієнтоване проектування програмних систем» та «Модельно-орієнтоване проектування програмних систем», а також дисциплін «Програмування», «Об'єктно-орієнтоване програмування», «Якість програмного забезпечення», «Вимоги до програмного забезпечення» магістерської та бакалаврської підготовки.

Отримані в результаті засвоєння дисципліни «Методи реінженірингу програмного забезпечення» теоретичні знання та практичні уміння є корисними для проведення наукових досліджень за темою дисертації.

## **3. Зміст навчальної дисципліни**

Дисципліна «Методи реінженірингу програмного забезпечення» передбачає вивчення таких тем:

Тема 1. Вступ по реінженірингу програмного забезпечення

Тема 2. Зворотня розробка програмного забезпечення

Тема 3. Рефакторинг

Тема 4. Особливості рефакторингу для різних мов програмування

## Модульна контрольна робота

### Екзамен

#### 4. Навчальні матеріали та ресурси

##### **Базова література:**

1. Чистий код. Створення і рефакторинг за допомогою Agile / Роберт Сесіл Мартін. – «Фабула», 2019. – 448 с.
2. Рефакторинг. Поліпшення існуючого коду / М. Фаулер, К. Бек, Дж. Брант, В. Опдайк, Д. Робертс. – «Діалектика», 2003. – 448 с.

##### **Додаткова література:**

3. Michael Feathers / Working Effectively with Legacy Code. - Person, 2004. – 464с.
4. 10 Idiomatic Ways to Refactor Your Python Code / Yong Cui. – [онлайн ресурс] - <https://towardsdatascience.com/10-idiotic-ways-to-refactor-your-python-code-cbb05bb0c820> - 3.07.2000р.

#### Навчальний контент

#### 5. Методика опанування навчальної дисципліни (освітнього компонента)

№ з/п	Тип навчального заняття	Опис навчального заняття
<i>Тема 1. Вступ по реінженірингу програмного забезпечення</i>		
1	<i>Лекція 1. Вступ по реінженірингу програмного забезпечення</i>	<i>Що таке реінженіринг програмного забезпечення. Коли проводити реінженіринг. Основні проблеми при плануванні задач. Програмна археологія. Завдання на СРС: п. 4, № 1.</i>
2	<i>Лекція 2. Етапи та процеси реінженірингу програмного забезпечення</i>	<i>Етапи та процеси реінженірингу програмного забезпечення.</i>
<i>Тема 2. Зворотня розробка програмного забезпечення</i>		
3	<i>Лекція 3. Зворотня розробка програмного забезпечення</i>	<i>Основні задачі та етапи зворотньої розробки програмного забезпечення. Завдання на СРС: п. 4, № 1.</i>
4	<i>Комп'ютерний практикум 1. Формулювання функціональних вимог до застарілого програмного забезпечення.</i>	<i>Завдання: За допомогою методів зворотньої розробки створити список вимог до застарілого програмного забезпечення. Завдання на СРС: п. 6, № 8.</i>
<i>Тема 3. Рефакторинг</i>		

5	Лекція 4. Вступ до рефакторингу. Запахи коду: порушувачі ООД, великі об'єми, ускладнювачі змін.	Що таке рефакторинг і коли його проводити. Що таке запахи коду. Класифікація проблем вихідного коду. Порушення об'єктно-орієнтованого дизайну. Порушення норм об'ємів вихідного коду. Ускладнювачі змін коду. Завдання на СРС: п. 2, № 2.
6	Лекція 5. Запахи коду: забруднювачі, заплутані зв'язки між об'єктами, інші.	Запахи коду: забруднювачі вихідного коду, заплутані зв'язки між об'єктами, неповнота бібліотеки. Завдання на СРС: п. 2, № 2.
7	Комп'ютерний практикум 2. Визначити запахи коду для вихідного коду.	Завдання: Визначити запахи коду для даного вихідного коду. Завдання на СРС: п. 6, № 8.
8	Лекція 6. Рефакторинг на рівні даних, операторів. Складання методів.	Методи рефакторингу на рівні організації даних та умовних операторів, складання методів. Завдання на СРС: п. 2, № 3.
9	Лекція 7. Спрощення викликів методів. Переміщення функцій між об'єктами.	Методи рефакторингу: спрощення викликів методів та переміщення функцій між об'єктами. Завдання на СРС: п. 2, № 3.
10	Лекція 8. Рішення задач узагальнення. Рефакторинг на рівні системи.	Методи рефакторингу: рішення задач узагальнення. Розділення наслідування. Перетворення процедурного коду в об'єкти. Відділення предметної області від презентації. Виділення ієрархії. Завдання на СРС: п. 2, № 6.
11	Комп'ютерний практикум 3. Рефакторинг вихідного коду на с-подібній мові програмування.	Завдання: Ідентифікувати можливі проблеми в кодї, усунути їх за допомогою методів рефакторингу. Завдання на СРС: п. 6, № 14.
<b>Тема 4. Особливості рефакторингу для різних мов програмування</b>		
12	Лекція 9. Особливості рефакторингу для різних мов програмування	Додаткові методи рефакторингу на C++/C#. Додаткові методи рефакторингу на Java Script та Python Завдання на СРС: п. 3.
13	Комп'ютерний практикум 4. Ідентифікація та усунення можливих проблем вихідного коду на одній з мов програмування	Ідентифікація та усунення можливих проблем вихідного коду на одній з мов програмування Завдання на СРС: п. 6, № 21.
<b>Модульна контрольна робота</b>		

## 6. Самостійна робота студента/аспіранта

Дисципліна «Математичні та алгоритмічні основи комп'ютерної графіки» ґрунтується на самостійних підготовках до аудиторних занять на теоретичні та практичні теми.

№ з/п	Назва теми, що виноситься на самостійне опрацювання	Кількість годин	Література
1	Підготовка до лекції 1	1	1-3
2	Підготовка до лекції 2	1	1-3
3	Підготовка до лекції 3	1	1-3
4	Підготовка до комп'ютерного практикуму 1	1,5	1
5	Підготовка до лекції 4	1	1, 2
6	Підготовка до лекції 5	1	1-3
7	Підготовка до комп'ютерного практикуму 2	1,5	1
8	Підготовка до лекції 6	1	1, 2, 4
9	Підготовка до лекції 7	1	1
10	Підготовка до лекції 8	1	1-3
11	Підготовка до комп'ютерного практикуму 3	1,5	1,
12	Підготовка до лекції 9	1	1
13	Підготовка до комп'ютерного практикуму 4	1,5	1
14	Підготовка до модульної контрольної роботи	5	1, 2
15	Підготовка до екзамену	16	1-3
16	Початок роботи з застарілим програмним забезпеченням та визначення вимог до його роботи	12	1
17	Перегляд вихідного коду та ідентифікація запахів коду	12	1
18	Рефакторинг на рівні організації даних	6	1, 2
19	Рефакторинг на рівні умовних операторів	6	1, 2
20	Спрощення викликів методів	6	2
21	Переміщення функцій між об'єктами.	6	2, 4
22	Вирішення задач узагальнення.	6	1
23	Розділення наслідування.	6	1, 2
24	Перетворення процедурного коду в об'єкти	6	1
25	Відділення предметної області від презентації та виділення ієрархії	6	1, 2

## 7. Політика навчальної дисципліни (освітнього компонента)

- Відвідування лекційних занять є обов'язковим.
- Відвідування занять комп'ютерного практикуму може бути епізодичним та за потреби захисту робіт комп'ютерного практикуму.
- Правила поведінки на заняттях: активність, повага до присутніх, відключення телефонів.
- Дотримання політики академічної доброчесності.
- Правила захисту робіт комп'ютерного практикуму: роботи повинні бути зроблені згідно варіанту здобувача освіти, що визначається його номером у списку групи.
- Правила призначення заохочувальних та штрафних балів є наступними.

Штрафні бали нараховуються за:

- плагіат (код програми не відповідає варіанту завдання, ідентичність коду програми серед різних робіт) у роботах комп'ютерного практикуму: -5 балів за кожну спробу.

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

Протягом семестру студенти виконують 4 комп'ютерних практикуми. Максимальна кількість балів за кожний комп'ютерний практикум: 10 балів.

Бали нараховуються за:

- якість виконання лабораторної роботи (комп'ютерного практикуму): 0-4 бали;
- відповідь під час захисту лабораторної роботи (комп'ютерного практикуму): 0-3 бали;
- своєчасне представлення роботи до захисту: 0-3 бали.

Критерії оцінювання якості виконання:

- 4 бали – робота виконана якісно, в повному обсязі;
- 2-3 бали – робота виконана якісно, в повному обсязі, але має недоліки;
- 1 бал – робота виконана в повному обсязі, але містить незначні помилки;
- 0 балів – робота виконана не в повному обсязі, або містить суттєві помилки.

Критерії оцінювання відповіді:

- 3 бали – відповідь повна, добре аргументована;
- 2 бали – в цілому відповідь вірна, але має недоліки або незначні помилки;
- 1 бал – у відповіді є суттєві помилки;
- 0 балів – немає відповіді або відповідь невірна.

Критерії оцінювання своєчасності представлення роботи до захисту:

- 3 бали – робота представлена до захисту не пізніше вказаного терміну;
- 2 бали – робота представлена до захисту пізніше вказаного терміну (до одного тижня);
- 1 бали – робота представлена до захисту пізніше вказаного терміну (від одного до двох тижнів);
- 0 балів – робота представлена до захисту пізніше вказаного терміну (від двох тижнів).

Максимальна кількість балів за виконання та захист комп'ютерних практикумів:

10 балів × 4 лаб. робіт (комп. практ.) = 40 балів.

Завдання на **модульну контрольну роботу** складається з 4 питань – 2 теоретичних та 2 практичних. Відповідь на кожне теоретичне запитання оцінюється 10 балами, а відповідь на практичне запитання оцінюється 10 балами.

Критерії оцінювання кожного теоретичного запитання контрольної роботи:

- 9-10 балів – відповідь вірна, повна, добре аргументована;
- 7-8 балів – відповідь вірна, розгорнута, але не дуже добре аргументована;
- 5-6 балів – в цілому відповідь вірна, але має недоліки;

3-4 балів – у відповіді є незначні помилки;  
1-2 бали – у відповіді є суттєві помилки;  
0 балів – немає відповіді або відповідь невірна.

*Критерії оцінювання практичного запитання контрольної роботи:*

9-10 балів – відповідь вірна, розрахунки виконані у повному обсязі;  
7-8 балів – відповідь вірна, але не дуже добре підкріплена розрахунками;  
5-6 балів – в цілому відповідь вірна, але має недоліки;  
3-4 балів – у відповіді є незначні помилки;  
1-2 бали – у відповіді є суттєві помилки;  
0 балів – немає відповіді або відповідь невірна.

*Максимальна кількість балів за модульну контрольну роботу:*

10 балів × 2 теоретичні запитання + 10 балів × 2 практичні запитання = 20 балів.

Завдання на **екзаміні** складається з 3 питань – 2 теоретичних та 1 практичного. Відповідь на кожне теоретичне запитання оцінюється 15 балами, а відповідь на практичне запитання оцінюється 10 балами.

*Критерії оцінювання кожного теоретичного запитання екзаменаційної роботи:*

14-15 балів – відповідь вірна, повна, добре аргументована;  
11-13 балів – відповідь вірна, розгорнута, але не дуже добре аргументована;  
8-10 балів – в цілому відповідь вірна, але має недоліки;  
5-7 балів – у відповіді є незначні помилки;  
1-4 бали – у відповіді є суттєві помилки;  
0 балів – немає відповіді або відповідь невірна.

*Критерії оцінювання практичного запитання екзаменаційної роботи:*

9-10 балів – відповідь вірна, розрахунки виконані у повному обсязі;  
7-8 балів – відповідь вірна, але не дуже добре підкріплена розрахунками;  
5-6 балів – в цілому відповідь вірна, але має недоліки;  
3-4 балів – у відповіді є незначні помилки;  
1-2 бали – у відповіді є суттєві помилки;  
0 балів – немає відповіді або відповідь невірна.

**Максимальна кількість балів за відповідь на іспиті:**

$R_E = 15 \text{ балів} \times 2 \text{ теоретичні запитання} + 20 \text{ балів} \times 1 \text{ практичне запитання} = 40 \text{ балів.}$

Семестрова складова рейтингової шкали  $R_C = 60 \text{ балів}$ , вона визначається як сума балів, отриманих за виконання та захист лабораторних робіт та контрольних робіт.

Екзаменаційна складова рейтингової шкали  $R_E = 40 \text{ балів.}$

Рейтингова шкала з дисципліни дорівнює:  $R = R_C + R_E = 100 \text{ балів.}$

Календарний контроль: провадиться двічі на семестр як моніторинг поточного стану виконання вимог силабусу.

На першій атестації (8-й тиждень) студент отримує «зараховано», якщо його поточний рейтинг не менше 10 балів (50 % від максимальної кількості балів, яку може отримати студент до першої атестації).

На другій атестації (14-й тиждень) студент отримує «зараховано», якщо його поточний рейтинг не менше 15 балів (50 % від максимальної кількості балів, яку може отримати студент до другої атестації).

*Семестровий контроль: іспит.*

*Умови допуску до семестрового контролю:*

*Необхідною умовою допуску студента до іспиту є виконання і захист всіх лабораторних робіт та семестровий рейтинг студента ( $r_c$ ) не менше 50 % від  $R_c$ , тобто не менше 30 балів. В іншому разі студент повинен виконати додаткову роботу та підвищити свій рейтинг.*

*Сумарний рейтинг студента  $RD$  визначається як сума семестрового рейтингу студента  $r_c$  та балів  $r_E$ , отриманих на іспиті. Оцінка (ECTS та традиційна) виставляється відповідно до значення  $RD$  згідно з таблицею (Таблиця відповідності рейтингових балів оцінкам за університетською шкалою).*

*Склад та критерії оцінювання залікової контрольної роботи:*

*Таблиця відповідності 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

## **9. Додаткова інформація з дисципліни (освітнього компонента)**

*Перелік питань, які виносяться на семестровий контроль, наведено у Додатку 1.*

**Робочу програму навчальної дисципліни (силабус):**

**Складено** к.т.н., ст. викл., Хіцко Я.В.

**Ухвалено** кафедрою ПЗКС (протокол № 1 від 27.08.2020 р.)

**Погоджено** Методичною комісією (протокол № 1 від 02.09.2020 р.)