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

Кафедра програмного забезпечення комп'ютерних систем

МЕТОДОЛОГІЯ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	***Другий (магістр)***
Галузь знань	*12 Інформаційні технології*
Спеціальність	*121 Інженерія програмного забезпечення*
Освітня програма	*Інженерія програмного забезпечення мультимедійних та інформаційно-пошукових систем*
Статус дисципліни	*Нормативна*
Форма навчання	*Очна (денна)*
Рік підготовки, семестр	*1 рік підготовки, 2 семестр*
Обсяг дисципліни	*Лекції: 36 год., комп'ютерний практикум: 18 год., самостійна робота: 66 год.*
Семестровий контроль/ контрольні заходи	*Залік, модульна контрольна робота, календарний контроль*
Розклад занять	*Згідно розкладу на весняний семестр поточного навчального року (rozklad.kpi.ua)*
Мова викладання	*Українська*
Інформація про керівника курсу / викладачів	Лектор: *к.т.н., старший викладач, Хіцко Яна Володимирівна, khitsko@pzks.fpm.kpi.ua* Комп'ютерний практикум: к.*т.н., старший викладач, Хіцко Яна Володимирівна, khitsko@pzks.fpm.kpi.ua*
Розміщення курсу	Google classroom: https://classroom.google.com/c/NTU0ODU5MjM1NzY2?cjc=e7ug4pz

Програма навчальної дисципліни

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

Метою *вивчення дисципліни «Методологія інженерії програмного забезпечення» є формування у студентів здатностей:*

- *аналізувати вимоги до програмних системи та умов їх проектування;*
- *обирати методологію розробки програмних систем відповідно до визначених вимог та середовища проектування та конструювання програмного забезпечення;*
- *визначати та аналізувати метрики якості програмного забезпечення;*
- *забезпечувати якісну інспекцію артефактів розробки програмного забезпечення;*
- *забезпечувати модульне та інтеграційне тестування програмного забезпечення;*
- *визначати та аналізувати метрики якості програмного забезпечення;*
- *забезпечувати якісний рефакторинг існуючого програмного коду.*

Предметом *дисципліни «Методологія інженерії програмного забезпечення» є математичне та алгоритмічне забезпечення процесів аналізу, проектування, конструювання та рефакторингу вихідного коду.*

Вивчення дисципліни «Методологія інженерії програмного забезпечення» сприяє сформувати у здобувачів освіти загальних (ЗК) та фахових (ФК) ***компетентності****, необхідні для*

розв'язання практичних задач професійної діяльності, пов'язаної з розробленням, вдосконаленням та експлуатацією програмних систем різноманітного призначення:
***ЗК01 -** Здатність до абстрактного мислення, аналізу та синтезу;*
***ФК01 -** здатність аналізувати предметні області, формувати, класифікувати вимоги до програмного забезпечення;*
***ФК03 -** здатність проєктувати архітектуру програмного забезпечення, моделювати процеси функціонування окремих підсистем і модулів;*
***ФК05** - здатність розробляти, аналізувати та застосовувати специфікації, стандарти, правила і рекомендації в сфері інженерії програмного забезпечення;*
***ФК06** - здатність ефективно керувати фінансовими, людськими, технічними та іншими проєктними ресурсами у сфері інженерії програмного забезпечення;*
***ФК07** - здатність критично осмислювати проблеми у галузі інформаційних технологій та на межі галузей знань, інтегрувати відповідні знання та розв'язувати складні задачі у широких або мультидисциплінарних контекста;*
***ФК08** - здатність розробляти і координувати процеси, етапи та ітерації життєвого циклу програмного забезпечення на основі застосування сучасних моделей, методів та технологій розроблення програмного забезпечення;*
***ФК09** - здатність забезпечувати якість програмного забезпечення;*
***ФК16** - здатність застосовувати на практиці методології інженерії програмного забезпечення.*

*Вивчення дисципліни «Методологія інженерії програмного забезпечення» сприяє формуванню у студентів наступних **програмних результатів навчання** (ПРН) за освітньою програмою:*
***ПРН01** - знати і застосовувати сучасні професійні стандарти та нормативно-правові документи з інженерії програмного забезпечення;*
***ПРН02** - оцінювати і вибирати ефективні методи і моделі розроблення, впровадження, супроводу програмного забезпечення та управління відповідними процесами на всіх етапах життєвого циклу;*
***ПРН03** - будувати і досліджувати моделі інформаційних процесів у прикладній області;*
***ПРН04** - виявляти інформаційні потреби і класифікувати дані для проєктування програмного забезпечення;*
***ПРН05** - розробляти, аналізувати, обґрунтовувати та систематизувати вимоги до програмного забезпечення;*
***ПРН06** - розробляти і оцінювати стратегії проєктування програмних засобів; обґрунтовувати, аналізувати і оцінювати варіанти проектних рішень з точки зору якості кінцевого програмного продукту, ресурсних обмежень та інших факторі;*
***ПРН07** - аналізувати, оцінювати і застосовувати на системному рівні сучасні програмні та апаратні платформи для розв'язання складних задач інженерії програмного забезпечення;*
***ПРН08** - розробляти і модифікувати архітектуру програмного забезпечення для реалізації вимог замовника;*
***ПРН09** - обґрунтовано вибирати парадигми і мови програмування для розроблення програмного забезпечення; застосовувати на практиці сучасні засоби розроблення програмного забезпечення;*
***ПРН10** - модифікувати існуючі та розробляти нові алгоритмічні рішення детального проєктування програмного забезпечення;*
***ПРН11** - 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*
***ПРН13** - конфігурувати програмне забезпечення, керувати його змінами та розробленням програмної документації на всіх етапах життєвого циклу;*
***ПРН14** - прогнозувати розвиток програмних систем та інформаційних технологій;*

ПРН15 *- здійснювати реінжиніринг програмного забезпечення відповідно до вимог замовника;*
ПРН16 *- планувати, організовувати та здійснювати тестування, верифікацію та валідацію програмного забезпечення;*
ПРН17 *- збирати, аналізувати, оцінювати необхідну для розв'язання наукових і прикладних задач інформацію, використовуючи науково-технічну літературу, бази даних та інші джерела.*

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

Успішному вивченню дисципліни «Методологія інженерії програмного забезпечення» передує вивчення дисциплін «Програмування», «Об'єктно-орієнтоване програмування», «Якість програмного забезпечення», «Вимоги до програмного забезпечення» навчального плану підготовки бакалаврів за спеціальністю 121 Інженерія програмного забезпечення.

Отримані при засвоєнні дисципліни «Методологія інженерії програмного забезпечення» теоретичні знання та практичні уміння забезпечують успішне виконання курсових проєктів та магістерських дисертацій за спеціальністю 121 Інженерія програмного забезпечення.

3. Зміст навчальної дисципліни

Дисципліна «Методологія інженерії програмного забезпечення» передбачає вивчення таких тем:

Тема 1. Методології інженерії програмного забезпечення

Тема 2. Метрики програмного забезпечення

Модульна контрольна робота 1

Тема 3. Верифікація програмного забезпечення

Тема 4. Рефакторинг

Модульна контрольна робота 2

Залік

4. Навчальні матеріали та ресурси

Базова література:

1. *Електронний кампус НТУУ «КПІ ім. Ігоря Сікорського». Матеріали з дисципліни «Розроблення програмного забезпечення за методологією Agile». – Режим доступу : http://login.kpi.ua*
2. *Web-портал ФПМ. Архів матеріалів. Тека «Хіцко». – Режим доступу : http://fpm.kpi.ua/archive/dir.do?sys_id=obj_2*

Додаткова література:

3. *Project Management Institute, Inc. / A Guide to the Project Management Body of Knowledge: PMBOK Guide. Fifth Edition. — PMI, 2013. — 589 pp. — ISBN-13: 860-1200917796.*
4. *Booch, G., Rumbaugh, J., Jacobcon, I. The Unified Modeling Language User Guide [Text] / G. Booch, J. Rumbaugh, I. Jacobcon — Addison-Wesley, 1998. — 512 p. — ISBN 0-201-57168-4.*
5. *Kent Beck. "Manifesto for Agile Software Development" [електронний ресурс] / Kent Beck, James Grenning, Robert C. Martin, Mike Beedle, Jim Highsmith, Steve Mellor, Arie van Bennekum, Andrew Hunt, Ken Schwaber, Alistair Cockburn, Ron Jeffries, Jeff Sutherland, Ward Cunningham, Jon Kern, Dave Thomas, Martin Fowler, Brian Marick // Agile Alliance, 2001. – режим доступу - http://agilemanifesto.org/ .*
6. *Steve McConnell (1996). Rapid Development: Taming Wild Software Schedules, Microsoft Press Books, ISBN 978-1-55615-900-8*

7. *DSDM Consortium. DSDM Atern: the Handbook [електронний ресурс] / DSDM Consortium, 2008. – режим доступу - https://www.agilebusiness.org/resources/dsdm-handbooks/dsdm-atern-handbook-2008 .*
8. *Schwaber, Ken (2004). "SCRUM Development Process"(PDF). Advanced Development Methods - режим доступу - http://www.jeffsutherland.org/oopsla/schwapub.pdf*
9. *"Extreme Programming Rules". extremeprogramming.org.*
10. *Palmer, S.R., & Felsing, J.M. (2002). A Practical Guide to Feature-Driven Development. Prentice Hall. (ISBN 0-13-067615-2)*
11. *Cockburn, Alistair. Crystal Clear, A Human-Powered Methodology for Small Teams [Text] / Alistair Cockburn // Addison-Wesley Professional, 2004. – pp.336. - ISBN 0-201-69947-8.*
12. *Kanban: Successful Evolutionary Change for Your Technology Business, David J. Anderson. (United States, Blue Hole Press, 2010. ISBN 978-0984521401*
13. *Scrumban: Essays on Kanban Systems for Lean Software Development, Corey Ladas. (United States, Modus Cooperandi Press, 2009. ISBN 9780578002149*
14. *Mary Poppendieck; Tom Poppendieck (2003). Lean Software Development: An Agile Toolkit. Addison-Wesley Professional. ISBN 978-0-321-15078-3.*
15. *Cohn, Mike. "Planning Poker Cards: Effective Agile Planning and Estimation". Mountain Goat Software, 30 March 2016. - режим доступу - https://www.mountaingoatsoftware.com/tools/planning-poker*
16. *Eric Evans, 2015 Domain Driven Design, Definitions and Pattern Summaries. Режим доступу - https://domainlanguage.com/wp-content/uploads/2016/05/DDD_Reference_2015-03.pdf*
17. *Крачтен Филипп. Введение в Rational Unified Process / Крачтен Филипп // 2-е изд.: Пер. с англ. — М.: Издательский дом "Вильямс", 2002. — 240 с.*
18. *Halstead, M. H. Elements of Software Science [Text] / M. H. Halstead. — New York, Elsevier North-Hollan, 1977. — p. 128. — ISBN 0444002057.*
19. *Chidamber, S. R. A Metrics Suite for Object Oriented Design [Text] / S.R. Chidamber, C. F. Kemerer // IEEE Transactions on Software Engineering. — June 1994. — vol. 20, No. 6. — pp. 476-493.*
20. *Graham, I. Object-Oriented Methods. Principles & Practice [Text] / I. Graham. 3rd Edition. — Addison-Wesley, 2000. — 864 pp. — ISBN 978-0201619133.*
21. *Hitz, M. Measuring Coupling in Object-Oriented Systems. [Text] / M. Hitz, B. Montazeri // Object Currents. — Apr 1996. — vol. 2, No. 4. — 17 pp.*
22. *Lorenz, M. Object-Oriented Software Metrics [Text] / M. Lorenz, J. Kidd. — Prentice Hall, 1994. — 146 p. — ISBN 978-0131792920.*
23. *Humphrey, Watts. The Personal Unified Process [Text] /* Humphrey, W. S. // Mass.: Addison-Wesley, 1983. - ISBN 0-201-18095-2.
24. *Рефакторинг. Поліпшення існуючого коду / М. Фаулер, К. Бек, Дж. Брант, В. Опдайк, Д. Робертс. – «Діалектика», 2003. – 448 с.*

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

№ з/п	*Тип навчального заняття*	*Опис навчального заняття*
Тема 1. Методології інженерії програмного забезпечення		
1	*Лекція 1. Життєвий цикл розробки програмного забезпечення*	*Фази проекту згідно методології PMI та процесу розробки програмного забезпечення. Моделі життєвого циклу. Водоспадна модель. Модель прототипування. Модель великого вибуху.V-подібна. Ітерація. Інкрементальні та ітеративні моделі. Спіральна модель.*

2	*Лекція 2. Методології розробки програмного забезпечення*	*Основні методології розробки програмного забезпечення. RAD. Основні принципи Agile. Основні поняття та практики XP. Цикл розробки по XP.*
3	*Лекція 3. Scrum*	*Scrum: основні поняття та цикл розробки. Журнал побажань проекту, історія користувача та зустрічі в Scrum.*
4	*Комп'ютерний практикум 1. SCRUM*	*Завдання: запланувати проект та розробити три його ітерації, використовуючи процес SCRUM*
5	*Лекція 4. Методології DSDM, FDD та Crystal, KANBAN*	*Основні аспекти методологій DSDM, FDD, KANBAN та Crystal methodologies.*
6	*Лекція 5. Загальні практики методологій Agile*	*Ітеративна та інкрементна розробка. Історії користувача. Сюжетне моделювання. Неперервна інтеграція. Парне програмування. Крос-функціональна команда. Покер планування. Предметно-орієнтоване проектування. Тестування та моделювання Agile.*
7	*Лекція 6. Уніфікований процес розробки програмного забезпечення.*	*Rational Unified Process: динамічний та статичний аспекти, дисципліни, фази, ітерації та артефакти.*
8	*Лекція 7. Процеси управління конфігурацією*	*Configuration Management Process. Установка сервера підтримки версій продукту.*
Тема 2. Метрики програмного забезпечення		
9	*Лекція 8. Метрики програмного забезпечення. Кількісні метрики*	*Метрики програмного забезпечення. Типи метрик. Метрики розміру програми. LOC-оцінки. Метрика рівня коментування програмного коду. Метрики Холстеда. Функціонально-орієнтовані метрики. Кількість функціональних точок.*
10	*Лекція 9. Метрики складності*	*Метрики складності потоку керування програм. Метрики Цикломатичної складності МакКейба. Метрики складності потоку керування програм. Метрики складності потоку даних. Метрика 'модуль - глобальна змінна'. Метрика Чепіно. Метрики стилістики та зрозумілості програм. Метрика рівня коментуванні програмного коду. Метрики Холстеда.*
11	*Компьютерний практикум 2. Кількісні метрики ПЗ*	*Завдання: розробити програму, що розраховує кількісні метрики довільно обраного програмного коду на будь-якій мові програмування*
12	*Лекція 10. Особливості структурування ПЗ*	*Особливості структурування системи. Зв'язність модуля - cohesion. Типи зв'язності. Визначення зв'язності. Зчеплення модулів – coupling.*

13	Лекція 11. Об'єктно-орієнтовані метрики	Об'єктно-орієнтовані метрики. Зв'язність об'єктів. Метрики зв'язності за методами. Залежність зміни між класами - class level coupling.
14	Лекція 12. Об'єктно-орієнтовані метрики	Метрики Чідамбера і Кемерера - WMC, DIT, NOC, CBO, RFC-LCOM. Метрики Лоренца і Кідда. Метрики підтримки коду.
15	Компьютерний практикум 3. Об'єктно-орієнтовані метрики ПЗ	Завдання: розробити програму, що розраховує об’єктно-орієнтовані метрики довільно обраного програмного коду на будь-якій мові програмування
Модульна контрольна робота 1		
Тема 3. Верифікація програмного забезпечення		
16	Лекція 13. Верифікація і валідація програмного забезпечення	Відмінності верифікації та валідації програмного забезпечення. Артефакти проекту створення програмного забезпечення. Верифікація проектного рішення, плану, вимог, тестових планів. Аудит програмного забезпечення. Експертиза. Методи статичного аналізу. Динамічні методи
17	Лекція 14. Огляди та інспекції коду	Основні поняття технічного огляду програмного забезпечення. Основні аспекти аналізу метрик технічних оглядів. Формальна інспекція коду – процес, артефакти процесу та його учасники. Протягова перевірка, мета та ролі. Конгітивна протягова перевірка
18	Комп’ютерний практикум 4. Інспекції коду	Завдання: розробити програмну утиліту, викласти її код у вільний доступ для надання можливості інспекції коду та подальшого його вдосконалення
Тема 4. Рефакторинг		
19	Лекція 15. Вступ до рефакторингу. Запахи коду.	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. Забруднювачі вихідного коду, заплутані зв’язки між об’єктами, неповнота бібліотеки.
20	Лекція 16. Рефакторинг на рівні даних, операторів. Складання методів.	Методи рефакторингу на рівні організації даних та умовних операторів, складання методів.
21	Лекція 17. Спрощення викликів методів. Переміщення функцій між об’єктами.	Методи рефакторингу: спрощення викліків методів та переміщення функцій між об’єктами.
22	Лекція 18. 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.

Модульна контрольна робота 2

6. Самостійна робота студента/аспіранта

Дисципліна «Методологія інженерії програмного забезпечення» 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, стор. 9-28; 3, стор. 8-10; 4, стор. 8-18*
2	*Підготовка до лекції 2*	*1*	*3, стор. 54-63; 4, стор. 42-43, 47-49; 5, стор. 32-41;*
3	*Підготовка до лекції 3*	*1*	*5, 7*
4	*Підготовка до комп'ютерного практикуму 1*	*2*	*8*
5	*Підготовка до лекції 4*	*1*	*11, стор. 44-47; 12-13*
6	*Підготовка до лекції 5*	*1*	*5-6, 14-16*
7	*Підготовка до лекції 6*	*1*	*17, стор. 15-30*
8	*Підготовка до лекції 7*	*1*	*18*
9	*Підготовка до лекції 8*	*1*	*19*
10	*Підготовка до лекції 9*	*1*	*18-19*
11	*Підготовка до комп'ютерного практикуму 2*	*2*	*18-19*
12	*Підготовка до лекції 10*	*1*	*20, стор. 20-24; 21, стор. 41-48*
13	*Підготовка до лекції 11*	*1*	*20, стор. 20-24; 21, стор. 41-48*
14	*Підготовка до лекції 12*	*1*	*22, стор. 11-18*
15	*Підготовка до комп'ютерного практикуму 3*	*2*	*20, стор. 20-24; 21, стор. 41-48; 22, стор. 11-18*
16	*Підготовка до модульної контрольної роботи 1*	*6*	*5-8; 11, стор. 44-47; 12-13; 17, стор. 15-30; 18-19; 20, стор. 20-24; 21, стор. 41-48; 22, стор. 11-18*
17	*Підготовка до лекції 13*	*1*	*23, стор. 226-234*
18	*Підготовка до лекції 14*	*1*	*23, стор. 226-234*
19	*Підготовка до комп'ютерного практикуму 4*	*2*	*23, стор. 226-234*

20	*Підготовка до лекції 15*	*1*	*24, стор. 15-34*
21	*Підготовка до лекції 16*	*1*	*24, стор. 35-48*
22	*Підготовка до лекції 17*	*1*	*24, стор. 50-64*
23	*Підготовка до лекції 18*	*1*	*24, стор. 72-86*
24	*Підготовка до модульної контрольної роботи 2*	*6*	*23, стор. 226-234; 24, стор. 15-86*
25	*Підготовка до заліку*	*8*	*5-8; 11, стор. 44-47; 12-13; 17, стор. 15-30; 18-19; 20, стор. 20-24; 21, стор. 41-48; 22, стор. 11-18; 23, стор. 226-234; 24, стор. 15-86*
26	*Основні прийоми практики Test driven design.*	*4*	*4r, стор. 6-8*
27	*Процеси управління інтеграцією. Налаштування серверу для ранньої інтеграції та вміння керувати релізами.*	*4*	*6-7*
28	*Розробка через тестування. Задачу для лабораторної роботи 5 реалізувати за допомогою TDD.*	*4*	*4, стор. 16-18*
29	*Метрики зчеплення та зв'язності. Аналіз довільної програмної бібліотеки щодо метрик зв'язності та зчеплення застосовуючи середу розробки Visual Studio.*	*4*	*22, стор. 11-18*
30	*Рефакторинг: для заданого програмного модуля визначити метод рефакторингу.*	*4*	*20, стор. 20-24*

Політика та контроль

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 (РСО)

Протягом семестру студенти виконують 4 комп'ютерних практикуми. Максимальна кількість балів за кожний комп'ютерний практикум: 15 балів.

Бали нараховуються за:
- якість виконання лабораторної роботи (комп'ютерного практикуму): 0-7 бали;
- відповідь під час захисту лабораторної роботи (комп'ютерного практикуму): 0-4 бали;
- своєчасне представлення роботи до захисту: 0-4 бали.

Критерії оцінювання якості виконання:
6-7 балів – робота виконана якісно, в повному обсязі;
4-5 бали – робота виконана якісно, в повному обсязі, але має недоліки;
1-3 бали – робота виконана в повному обсязі, але містить незначні помилки;
0 балів – робота виконана не в повному обсязі, або містить суттєві помилки.

Критерії оцінювання відповіді:
4 бали – відповідь повна, добре аргументована;
3 бали – відповідь повна, але недостатньо 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:
4 бали – робота представлена до захисту не пізніше вказаного терміну;
3 бали – робота представлена до захисту пізніше вказаного терміну (до одного тижня);
2 бали – робота представлена до захисту пізніше вказаного терміну (від одного до двох тижнів);
1 бал – робота представлена до захисту пізніше вказаного терміну (від двох до трьох тижнів);
0 балів – робота представлена до захисту пізніше вказаного терміну (від трьох тижнів).

Максимальна кількість балів за виконання та захист комп'ютерних практикумів:
15 балів × 4 лаб. робіт (комп. практ.) = 60 балів.

Протягом семестру студенти виконують дві модульні контрольні роботи. Завдання на ***модульну контрольну роботу*** *складається з 5 питань – 4 теоретичних та 1 практичного. Відповідь на кожне запитання оцінюється 4 балами.*

Критерії оцінювання кожного теоретичного запитання контрольної роботи:
4 бали – відповідь вірна, повна, добре аргументована;
3 бали – в цілому відповідь вірна, але має недоліки;
2 бали – у відповіді є незначні помилки;
1 бал – у відповіді є суттєві помилки;
0 балів – немає відповіді або відповідь невірна.

Критерії оцінювання практичного запитання контрольної роботи:
4 бали – відповідь вірна, розрахунки виконані у повному обсязі;
3 бали – в цілому відповідь вірна, але має недоліки;
2 бали – у відповіді є незначні помилки;
1 бал – у відповіді є суттєві помилки;
0 балів – немає відповіді або відповідь невірна.

Максимальна кількість балів за модульну контрольну роботу:
4 бали × 4 теоретичні запитання + 4 бали × 1 практичне запитання = 20 балів.

Максимальна кількість балів за модульні контрольні роботи:
20 балів × 2 контр. роботи = 40 балів.

Рейтингова шкала з дисципліни дорівнює:
$R = R_C$ = 60 балів + 40 балів = 100 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 12 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 20 балів (50 % від максимальної кількості балів, яку може отримати студент до другої атестації).

Семестровий контроль: залік
Умови допуску до семестрового контролю:
При семестровому рейтингу (r_C) не менше 60 балів та зарахуванні усіх робіт комп'ютерного практикуму, студент отримує залік «автоматом» відповідно до таблиці (Таблиця відповідності рейтингових балів оцінкам за університетською шкалою). В іншому разі він має виконувати залікову контрольну роботу.

Необхідною умовою допуску до залікової контрольної роботи є виконання і захист комп'ютерного практикуму.

Якщо студент не погоджується з оцінкою «автоматом», то може спробувати підвищити свою оцінку шляхом написання залікової контрольної роботи, при цьому його бали, отримані за семестр, зберігаються, а з двох отриманих студентом оцінок виставляється краща («м'яка» система оцінювання).

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

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 № 12 від 26.04.2023 р.)

Погоджено Методичною комісією факультету прикладної математики (протокол № 10 від 26.05.2023 р.)

Додаток 1. Перелік питань, які виносяться на семестровий контроль

1. *Назвіть основні практики в Feature Driven Design (мінімум 3).*
2. *Як проходить щоденна нарада в Scrum і які питання обговорюються?*
3. *Чи розповсюджуються проміжні версії продукту в каскадній та інкрементній моделях розробки програмного забезпечення?*
4. *Проєкти якого масштабу можуть використовувати Crystal Orange?*
5. *Якими є умови застосування методології RAD?*
6. *Чи є закінчення спринту в Scrumban?*
7. *Чи є закінчення спринту в Kanban?*
8. *Чим відрізняється user story від use case?*
9. *Як чином оцінюється продуктивність команди в Scrum?*
10. *Що таке анти-патерн 'watch the master' в парному програмуванні?*
11. *Як можуть змінюватися вимоги в каскадній та інкрементній моделях розробки ПЗ?*
12. *Чи може архітектор або менеджер проєкту заздалегідь визначати оцінки задач в процесі покеру планування?*
13. *Для чого використовуються story points?*
14. *Що таке принцип MoSCoW?*
15. *Що таке Mock- і fake- об'єкти?*
16. *Чим відрізняється технічний огляд ПЗ від формальної інспекції ?*
17. *На які частини проєкту та як вплине зменшення бюджету проєкту?*
18. *Чи є необхідність переписувати драйвери на кожній стадії низхідного інтеграційного тестування?*
19. *На якій фазі RUP проходить процес оцінювання системи?*