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

Кафедра обчислювальної техніки

Методологія інженерії програмного забезпечення

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти ***Другий (магістерський)***

Галузь знань	*12 Інформаційні технології*
Спеціальність	*121 Інженерія програмного забезпечення*
Освітня програма	*Інженерія програмного забезпечення комп'ютерних систем*
Статус дисципліни	*Нормативна*
Форма навчання	*Очна (денна)*
Рік підготовки, семестр	*1 курс, весняний семестр*
Обсяг дисципліни	*4 кредитів*
Семестровий контроль/ контрольні заходи	*Залік*
Розклад занять	*Лекцій 36 годин, Лабораторних 18 годин, СРС 66 годин*
Мова викладання	*Українська*
Інформація про керівника курсу / викладачів	Керівник: професор, д.т.н., Стіренко С.Г. sergii.stirenko@gmail.com *Старший викладач кафедри обчислювальної техніки* *Шемсединов Т.Г. timur.shemsedinov@gmail.com* *Асистент кафедри обчислювальної техніки* *Кухар В.В. kukhar.vitalii@gmail.com*
Розміщення курсу	https://github.com/HowProgrammingWorks

Програма навчальної дисципліни

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

Метою навчальної дисципліни є формування у студентів здатностей (компетентностей) програмування, вільного володіння синтаксисом та методологією програмування, розуміння базових структур даних та парадигм. За результатами вивчення дисципліни здобувач має бути здатними вирішувати професійні завдання та володіти такими компетентностями:

- Здатність до абстрактного мислення, аналізу та синтезу (ЗК01);
- Здатність генерувати нові ідеї (креативність) (ЗК05);
- Здатність аналізувати предметні області, формувати, класифікувати вимоги до програмного забезпечення (ФК01);

- Здатність проєктувати архітектуру програмного забезпечення, моделювати процеси функціонування окремих підсистем і модулів (ФК03);
- Здатність розробляти, аналізувати та застосовувати специфікації, стандарти, правила і рекомендації в сфері інженерії програмного забезпечення (ФК05);
- Здатність розробляти і координувати процеси, етапи та ітерації життєвого циклу програмного забезпечення на основі застосування сучасних моделей, методів та технологій розроблення програмного забезпечення (ФК08);
- Здатність забезпечувати якість програмного забезпечення (ФК09);
- Здатність створювати та використовувати програмне забезпечення високопродуктивних комп'ютерних систем (ФК10);

Після засвоєння навчальної дисципліни студенти мають продемонструвати такі програмні результати навчання:

- Оцінювати і вибирати ефективні методи і моделі розроблення, впровадження, супроводу програмного забезпечення та управління відповідними процесами на всіх етапах життєвого циклу (ПРН02);
- Виявляти інформаційні потреби і класифікувати дані для проєктування програмного забезпечення (ПРН04);
- Розробляти, аналізувати, обґрунтовувати та систематизувати вимоги до програмного забезпечення (ПРН05);
- 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 факторів (ПРН06).
- обґрунтовано вибирати парадигми і мови програмування для розроблення програмного забезпечення; застосовувати на практиці сучасні засоби розроблення програмного забезпечення (ПРН09);
- Модифікувати існуючі та розробляти нові алгоритмічні рішення детального проєктування програмного забезпечення (ПРН10);
- збирати, аналізувати, оцінювати необхідну для розв'язання наукових і прикладних задач інформацію, використовуючи науково-технічну літературу, бази даних та інші джерела (ПРН17);

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

Дисципліни, що передують: Основи програмування, Компоненти програмної інженерії, Основи розробки програмного забезпечення на платформі Node.Js, Системне програмування, Паралельні та розподілені обчислення, Гнучкі методи програмування, Моделювання програмного забезпечення.

Дисципліни, до яких готує цей курс: Методологія інженерії програмного забезпечення. Курсовий проєкт, Наукова робота за темою магістерської дисертації. Частина 2, Практика, Робота над магістерською дисертацією.

3. Зміст навчальної дисципліни

Тема 1. Стан додатків, структури даних та колекції

Тема 2. Підходи до роботи зі станом: stateful and stateless

Тема 3. Структури та записи

Тема 4. Стек, черга, дек

Тема 5. Дерева та графи
Тема 6. Проекції та відображення наборів даних
Тема 7. Оцінка обчислювальної складності
Тема 8. Структура додатку: файли, модулі, компоненти
Тема 9. Об'єкт, прототип та клас
Тема 10. Залежності та бібліотеки
Тема 11. Регулярні вирази
Тема 12. Фабрики та пули
Тема 13. I/O(введення-виведення) та файли
Тема 14. Мономорфний та поліморфний код, інлайн-кеш, приховані класи
Тема 15. Вимірювання продуктивності коду та оптимізація
Тема 16. Асинхронне програмування на callback`ах
Тема 17. Асинхронне програмування на promise`ах
Тема 18. Асинхронні функції, async/await, thenable, обробка помилок
Тема 19. Незмінні структури даних (immutable)
Тема 20. Автоматне програмування: кінцеві автомати (машини станів)
Тема 21. Шаблон Singleton (синглтон) JavaScript
Тема 22. Функціональні об'єкти, функтори та монади
Тема 23. Асинхронні генератори та асинхронні ітератори
Тема 24. Перерахований тип (enum)

4. Навчальні матеріали та ресурси

Базова:

1. Шемсединов Т.Г., Нечай Д.О., Кухар В.В., Орленко О.А., Голіков О.Г., Білочуб М.М., Духін В., Іванова Л.А., Чорненький А.Ю. та інші. Приклади коду та приклади проектів [Електронний ресурс] розміщено за адресою: https://github.com/HowProgrammingWorks/
2. Refactoring: Improving the Design of Existing Code // Martin Fowler
3. Clean Code: A Handbook of Agile Software Craftsmanship // Robert C. Martin
4. Introduction to Algorithms, 3rd Edition // Thomas H. Cormen
5. Gang of Four Design Patterns // Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides

Додаткова:

1. Шемсединов Т.Г., Нечай Д.О., Кухар В.В., Орленко О.А., Голіков О.Г., Білочуб М.М., Духін В., Іванова Л.А., Чорненький А.Ю. та інші. Технологічний стек Metarhia [Електронний ресурс] розміщено за адресою: https://github.com/metarhia/
2. Algorithms Unlocked // Thomas H. Cormen
3. The Art of Computer Programming // Donald Knuth
4. Code Complete // Steve McConnell
5. Designing Object Oriented C++ Applications Using The Booch Method // Robert C. Martin
6. Extreme Programming Explained // Kent Beck
7. Analysis Patterns: Reusable Object Models // Martin Fowler

5. Методика опанування навчальної дисципліни (освітнього компонента)

Основні завдання циклу лабораторних занять (комп'ютерного практикуму) – надбання практичних навичок використання структур даних та колекцій, написання коду, створення багатомодульних бібліотек та програм, оптимізація коду, відлагодження та декомпозиція коду, групова робота та використання систем контролю версій.

№ з/п	Назва лабораторної роботи (комп'ютерного практикуму)	Кількість ауд. годин
1	Реалізація зв'язаних списків	3
2	Побудова модулів та залежностей, структура додатку	3
3	Робота з файлами: дескриптори та файлові потоки	3
4	Обхід дерев та графів	3
5	Оптимізація коду під віртуальну машину V8	3
6	Використання рефлексії та інтроспекції	3
	Разом:	18

6. Самостійна робота студента

У процесі виконання індивідуальних завдань студенти повинні опрацьовувати знання, отримані під час лекцій та самостійної роботи, самостійно вивчати визначені теми, поглиблювати свої знання для подальшого навчання. Самостійна робота студентів полягає в наступному:

- підготовці до лекційних занять по вивченню попереднього лекційного матеріалу;
- розглядання та модифікація прикладів коду та проектів з наданих викладачем git репозиторіїв;
- виконання лабораторних робіт з вивченням теорії та реалізацією наведеної теми у програмному коді;
- підготовка та участь у обговоренні тем на семінарах;
- пір-рев'ю роду колег студентів;
- підготовка та захищення коду на код-рев'ю викладача.

№ з/п	Назва теми, що виноситься на самостійне опрацювання	Кількість годин СРС
1	Контрибьютшен у Open-Source проєкти	26
2	Налаштування системи тестування та середовища розробки IDE	20
3	Створення та налаштування профілю у Github	20

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

Під час занять з навчальної дисципліни студенти повинні дотримуватись певних дисциплінарних правил:
1) так використовувати чат-групи, репозиторії та середовище виконання, щоб не створювати проблем з зайвими нотифікаціями іншим учасникам учбового процесу та викладачеві;
2) вчасно виконувати роботу та коммітити все кожного дня у систему контролю версій, не створювати багато коммітів у одному пул-реквесту, не створювати великих коммітів, розділяти все на окремі тематичні комміти, чітко та зрозуміло описувати комміти та пул-реквести, додавати теги для лінкування issues, PR, та коммітів у репозиторіях;
3) не допускаються використання піратських копій середовищ розробки, операційних систем чи інших засобів розробки та розгортання як на своїх обчислювальних засобах, так і на хмарних;
4) спамити, постити забагато мемів та стікерів у групах та репозиторіях;
5) якщо виникає питання, потрібно спочатку шукати у матеріалах курсу, потім у інтернеті, потім запитати у асистентів та інших студентів, а потім вже, тільки коли рішення не знайдено чи воно незрозуміле, турбувати викладача;

Лабораторні роботи здаються тільки через Github у відкритому коді (до коду студенти додають ліцензію MIT), а репозиторій повинен мати історію розробки, щоб викладач міг простежити послідовність написання коду та авторство. Розробка відбуваєтья у feature-гілках git, після чого перевірка коду проходить через PR та включає рев'ю, яке викладач робить у пул-реквесті. Історія рев'ю залишається у Github аккаунті студента.

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

Види контролю з навчальної дисципліни «Основи програмування 2. Методології програмування» включають:

Лабораторні роботи

Заплановано самостійне виконання 6 лабораторних робіт.
Теми лабораторних робіт узгоджені у часі та за змістом з темами лекцій

Поточний контроль:

Передбачено 2 попередні рев'ю кода за курс, які повністю охоплюють розглянуту на лекціях тематику навчальної дисципліни. До рев'ю коду викладач може додавати теоретичні питання, які розкривають розуміння теми студентом.

Семестровий контроль

Остаточний рев'ю коду робиться за декілька підходів, але по дедлайну пул-реквест зачиняється, та фіксуються усі зауваження, що виправлені чи не виправлені.

Залік

проводиться у вигляді співбесіди зі студентом для об'єктивного визначення рівня знань, умінь та практичних навичок, отриманих за семестр чи парного лайвкодінгу за викладачем чи асистентом, парного кодінгу з іншим студентом чи повернення до написаного за семестр коду та усунення його недоліків чи обговорення його особливостей.

Оскільки кредитний модуль має семестрову атестацію у вигляді заліку, рейтингова система оцінювання побудована за типом РСО – 1. Рейтинг студента з кредитного модуля складається з балів, які він отримує за види робіт відповідно до таблиці 4.

Таблиця 4

Оцінювання окремих видів навчальної роботи студента

Розділ 1,2		Розділ 3,4	
Вид навчальної роботи	Максимальна кількість балів	Вид навчальної роботи	Максимальна кількість балів
Виконання та захист лабораторної роботи № 1	10	Виконання та захист лабораторної роботи № 4	10
Виконання та захист лабораторної роботи № 2	10	Виконання та захист лабораторної роботи № 5	10
Виконання та захист лабораторної роботи № 3	10	Виконання та захист лабораторної роботи № 6	10
Поточний код-рев'ю №1	10	Поточний код-рев'ю №2	10
		Фінальний код-рев'ю	20
Усього за семестр			**100**

Разом за лабораторні роботи (максимальна кількість балів) – 60

Індивідуальний семестровий рейтинг студента (підсумкова семестрова рейтингова оцінка **RD**) є сумою балів, отриманих студентом протягом семестру за участі у семінарах, рев'ю коду, обговореннях.

Усі студенти, незалежно від того, чи вони виконали всі умови допуску до семестрової атестації з кредитного модуля та мають рейтингову оцінку не менше 60 балів, проходять співбесіду з викладачем.

Необхідною умовою допуску студента до заліку є його індивідуальний семестровий рейтинг (**RD**) не менший, ніж 30% від максимуму балів, тобто 30 балів, здані 4 лабораторні роботи та наявність однієї позитивної атестації в семестрі. При невиконанні хоча б однієї зі згаданих умов студент до заліку не допускається.

Сума підсумкової семестрової (**RD**) та залікової рейтингових оцінок у балах становить підсумкову семестрову рейтингову оцінку, яка перераховується в оцінки за національною шкалою та шкалою ECTS (табл. 5).

Таблиця 5

Відповідність рейтингових балів оцінкам за університетською шкалою

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

Робочу програму навчальної дисципліни (силабус):

Склав професор кафедри обчислювальної техніки Стіренко С.Г.

Ухвалено кафедрою обчислювальної техніки (протокол № 10 від 25.05.2022)

Погоджено Методичною комісією факультету (протокол № 10 від 09.06.2022)