



Сучасні технології програмування

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Перший (бакалаврський)</i>
Галузь знань	17 Електроніка та телекомунікації
Спеціальність	172 Телекомунікації та радіотехніка
Освітня програма	Інформаційно-комунікаційні технології
Статус дисципліни	Вибіркова
Форма навчання	очна(денна)
Рік підготовки, семестр	рік третій, осінній семестр
Обсяг дисципліни	4 кредити ЄКТС, з них лекції 36 годин, практичні заняття 18 годин
Семестровий контроль/ контрольні заходи	Модульна контрольна робота Залік
Розклад занять	Згідно з розкладом
Мова викладання	Українська
Інформація про керівника курсу / викладачів	Лектор: к.т.н. Педан С.І., 066-120-04-65, stas.pedan@gmail.com Практичні заняття: к.т.н. Педан С.І.
Розміщення курсу	https://classroom.google.com/c/NjJwNjJzMDk1MjY3

Програма навчальної дисципліни

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

Під час вивчення дисципліни студентам надаються знання щодо використання сучасних технологій для організації спільної розробки програмного забезпечення (ПЗ), проектування систем за допомогою UML діаграм, базові поняття об'єктно-орієнтованої методології розробки ПЗ та використання шаблонів проектування, а також контролю якості розробки ПЗ. Також приділяється увага сучасним технологіям програмування систем передачі даних та мобільних телекомунікаційних пристроїв. Важливим аспектом дисципліни є отримання навичок об'єктно-орієнтованого програмування та тестування в результаті виконання серії самостійних практичних робіт.

Вивчення дисципліни базується на знанні студентами матеріалів, передбачуваних навчальними планами дисциплін:

- «Прикладне програмування в інформаційно-комунікаційних системах»;
- «Інформатика-1»;
- «Інформатика-2» та інші.

Дисципліна забезпечує студентам вивчення таких дисциплін:

- «Технології побудови web-орієнтованих систем»;

- «Системне проектування телекомунікаційних мереж»;
- «Інтелектуальна обробка інформації»;
- «Моделювання в інфо-комунікаційних системах»;
- «Технології інтернет»;
- «Програмування мікрокомп'ютерних систем обміну даних»;
- «Основи програмування мовою Python».

Цілі дисципліни	<p>Метою навчальної дисципліни є формування у студентів знань з загальних питань, які використовуються для забезпечення наступних процесів телекомунікацій:</p> <p>підготовка фахівця, який має базові компетенції, що засновані на системи знань в області системного проектування у системах, обладнанні і програмному забезпеченні систем телекомунікацій;</p> <p>формування та розвиток загальних і професійних компетентностей з впровадження та застосування технологій телекомунікацій і радіотехніки, що сприяють соціальній стійкості та мобільності випускника на ринку праці;</p> <p>формування у здобувачів освіти логічного мислення, розвиток їх інтелекту та здібностей;</p> <p>формування знань, вмінь і навичок, необхідних для розуміння сучасних технології програмування телекомунікаційних систем та мереж.</p>
------------------------	--

Згідно з вимогами освітньо-професійної програми студенти після засвоєння навчальної дисципліни мають продемонструвати такі результати навчання:

знання:

Здатність використовувати базові методи, способи та засоби отримання, передавання, обробки та зберігання інформації (ФК-3);

Здатність здійснювати комп'ютерне моделювання пристроїв, систем і процесів з використанням універсальних пакетів прикладних програм (ФК-4);

Здатність здійснювати розробку, тестування, використання та супровід спеціалізованого програмного забезпечення з дотриманням атрибутів якості, програмування прикладних задач, створення WEB-сайтів, моделювання та віртуалізацію інфокомунікаційних процесів, систем, мереж (ФК-16);

уміння:

Мати навички оцінювання, інтерпретації та синтезу інформації і даних (ПРН5);

Описувати принципи та процедури, що використовуються в телекомунікаційних системах, інформаційно-телекомунікаційних мережах та радіотехніці (ПРН8);

Знаходити, оцінювати і використовувати інформацію з різних джерел, необхідну для розв'язання професійних завдань, включаючи відтворення інформації через електронний пошук (ПРН18);

Здійснювати аналіз вимог до програмного забезпечення, створювати сценарії для його тестування, програмувати прикладні задачі, створювати власні математичні моделі, програмні модулі для структурування даних, аналізу, моделювання, віртуалізації, розробки та оптимізації інфокомунікаційних засобів та процесів (ПРН23).

2. Пререквізити та постреквізити дисципліни

Перелік дисциплін або знань та умінь, володіння якими необхідні здобувачу вищої освіти для успішного засвоєння дисципліни	Перелік дисциплін, які базуються на результатах навчання з даної дисципліни
Дисципліна вивчається на основі ступеня бакалавра базуючись на знаннях таких дисциплін: - «Прикладне програмування в інформаційно-комунікаційних системах»; - «Інформатика-1»; - «Інформатика-2» та інші.	- «Технології побудови web-орієнтованих систем»; - «Системне проектування телекомунікаційних мереж»; - «Інтелектуальна обробка інформації»; - «Моделювання в інфо-комунікаційних системах»; - «Технології інтернет»; - «Програмування мікрокомп'ютерних систем обміну даних»; - «Основи програмування мовою Python».

3. Зміст навчальної дисципліни

Тема 1. Організація процесу спільної розробки програмного забезпечення (ПЗ)

- 1.1 Життєвий цикл розробки ПЗ
- 1.2 Каскадна модель життєвого циклу розробки ПЗ
- 1.3 Agile моделі життєвого циклу розробки ПЗ

Тема 2. Основи роботи з системою контролю версій Git

- 2.1 Типи систем контролю версій
- 2.2 Збереження історії змін програмного коду
- 2.3 Робота з гілками для паралелізації розробки програмного коду
- 2.4 Аналіз змін програмного коду

Тема 3. Основи роботи з платформою GitHub

- 3.1 Схема розробки ПЗ з використанням Git та GitHub
- 3.2 Ініціалізація та структура GitHub репозиторію
- 3.3 Синхронізація змін локального та віддаленого репозиторіїв
- 3.4 Використання GitHub при командній розробці ПЗ
- 3.5 Контроль якості розробки з використанням безперервної інтеграції

Тема 4. Проектування та реалізація ПЗ з використанням принципів об'єктно-орієнтованого програмування (ООП)

- 4.1 Сучасні методології програмування (процедурна, об'єктно-орієнтована, функціональна)
- 4.2 Основи об'єктно-орієнтованої методології розробки ПЗ, поняття класу та об'єкту
- 4.3 Універсальна мова моделювання UML, проектування архітектури ПЗ (структурні, поведінкові діаграми та діаграми взаємодії)
- 4.4 Базові принципи ООП (абстракція, інкапсуляція, наслідування, поліморфізм)
- 4.5 Принципи проектування SOLID

Тема 5. Шаблони проектування ООП

- 5.1 Поняття шаблону проектування та класифікація
- 5.2 Породжувальні шаблони проектування (Будівельник (Builder), Фабричний метод (Factory method), Одинак (Singleton))
- 5.3 Структурні шаблони проектування (Компонувальник (Composite), Адаптер (Adapter), Декоратор (Decorator))

5.4 Поведінкові шаблони проектування (Спостерігач (Observer), Ланцюжок обов'язків (Chain of responsibility))

Тема 6. Основи програмування мобільних пристроїв

6.1 Основи мережевої передачі даних, контроль цілісності даних, використання завадостійкого кодування

6.2 Поняття розробки мобільних додатків та сервісів, мобільні операційні системи (ОС)

6.3 Структура компонентів мобільної ОС Android, історія розвитку версій

6.4 Практичні основи програмування на ОС Android (датчики, комунікація, мережева передача даних)

Тема 7. Методи забезпечення якості розробки ПЗ

7.1 Основи контролю якості розробки ПЗ (місце в життєвому циклі розробки, поняття тестового плану та кейсу)

7.2 Класифікація методів тестування якості ПЗ (ручне/автоматизоване, чорного/білого ящиків, функціональне/нефункціональне)

7.3 Основні види функціонального тестування (модульне тестування ПЗ, реалізація на ОС Android, інтеграційні та системні тести)

7.4 Основні види нефункціонального тестування (тестування продуктивності та безпеки)

Тема 8. Перспективні технології програмування та розробки

8.1 Основи штучного інтелекту (ШІ) та машинного навчання (лінійна алгебра, згорткові та рекурентні нейронні мережі, генеративний ШІ)

8.2 Технології майбутнього (Метавсесвіт, Біометрика, 6G, доповнена та віртуальна реальність, інтернет речей, квантові обчислення)

4. Навчальні матеріали та ресурси

4.1 Базова література:

1. Lemke, Gillian, "The Software Development Life Cycle and Its Application" (2018). Senior Honors Theses. 589. <https://commons.emich.edu/honors/589>
2. Scott Chacon, Ben Straub, "Pro Git. Second edition" (2014). Apress, <https://git-scm.com/book/uk/v2>
3. Bobby Iliev, "Introduction to Git and GitHub" (2023). <https://github.com/bobbyiliev/introduction-to-git-and-github-ebook>
4. Объектно-ориентированный анализ и проектирование. С примерами приложений. Буч Гради, Максимчук Роберт А., Энгел Майкл У., Янг Бобби Дж., Коаллен Джим, Хьюстон Келли А.: Пер. с англ. – 3-е изд. – М.: ООО «И.Д.Вильямс», 2008. – 720 с.
5. Фаулер М. UML. Основы. Краткое руководство по стандартному языку объектного моделирования [Текст] / М. Фаулер; пер. с англ. А. Петухова. - М.: Символ-Плюс, 2011. - 192 с.
6. Martin, Robert, "Agile Software Development: Principles, Patterns, and Practices" (2003). Prentice Hall PTR, 720p.
7. Gamma, Erich, Helm, Richard, Johnson, Ralph and Vlissides, John M.. Design Patterns: Elements of Reusable Object-Oriented Software. 1 : Addison-Wesley Professional, 1994.
8. Bill Phillips, Chris Stewart, Kristin Marsicano, Brian Gardner. Android Programming: The Big Nerd Ranch Guide (Big Nerd Ranch Guides), 2017.
9. Andreas C. Müller, Sarah Guido. Introduction to Machine Learning with Python, O'Reilly Media, Inc., 2016.

5. Методика опанування навчальної дисципліни (освітнього компонента)

5.1 Лекційні заняття – 36 годин

Тема 1. Організація процесу спільної розробки програмного забезпечення (ПЗ)

Лекція 1. Вступ в сучасні технології програмування

Основні питання:

1. Структура дисципліни та задачі курсу
2. Життєвий цикл розробки ПЗ
3. Каскадна модель життєвого циклу розробки ПЗ
4. Agile моделі життєвого циклу розробки ПЗ

Дидактичні засоби: Слайди за темою лекції, навчально-методичні матеріали за темою лекції.

Література: [1]

Завдання на СРС: підготувати відповідь на контрольні запитання до л. 1.

Тема 2. Основи роботи з системою контролю версій Git

Лекція 2. Основи роботи с системою контролю версій Git

Основні питання:

1. Типи систем контролю версій
2. Основні принципи роботи системи контролю версій Git
3. Збереження історії змін програмного коду (git add/commit)

Дидактичні засоби: Слайди за темою лекції, навчально-методичні матеріали за темою лекції.

Література: [2].

Завдання на СРС: підготувати відповідь на контрольні запитання до л. 2.

Лекція 3. Організація паралельної розробки ПЗ в команді

Основні питання:

1. Робота з гілками для паралелізації розробки програмного коду (git branch)
2. Об'єднання змін гілок розробки, вирішення конфліктів об'єднання змін програмного коду (git merge)

Дидактичні засоби: Слайди за темою лекції, навчально-методичні матеріали за темою лекції.

Література: [2].

Завдання на СРС: підготувати відповідь на контрольні запитання до л. 3.

Лекція 4. Управління історією змін програмного коду

Основні питання:

1. Аналіз змін програмного коду (git diff)
2. Тимчасове приховування змін програмного коду (git stash)
3. Навігація історією змін програмного коду (git checkout)
4. Повернення до попередніх версій ПЗ та видалення останніх змін (git reset)

Дидактичні засоби: Комп'ютерний проектор з набором експозицій до л. 4.

Література: [2].

Завдання на СРС: підготувати відповідь на контрольні запитання до л. 4.

Тема 3. Основи роботи з платформою GitHub

Лекція 5. Основи командної роботи з використанням платформи GitHub

Основні питання:

1. Схема розробки ПЗ з використанням Git та GitHub
2. Ініціалізація та структура GitHub репозиторію (git init/clone)
3. Синхронізація змін локального та віддаленого репозиторіїв (git push/fetch/pull)

Дидактичні засоби: Комп'ютерний проектор з набором експозицій до л. 5.

Література: [3].

Завдання на СРС: підготувати відповідь на контрольні запитання до л. 5

Лекція 6. Організація командного контролю якості розробки ПЗ

Основні питання:

1. Використання GitHub при командній розробці ПЗ
2. Етапи життєвого циклу змін програмного коду
3. Контроль якості змін програмного коду через створення pull request
4. Автоматизація контролю якості коду з використанням безперервної інтеграції.

Дидактичні засоби: Комп'ютерний проектор з набором експозицій до л. 6.

Література: [3].

Завдання на СРС: підготувати відповідь на контрольні запитання до л. 6.

Тема 4. Проектування та реалізація ПЗ з використанням принципів об'єктно-орієнтованого програмування (ООП)

Лекція 7. Сучасні методології програмування

1. Поняття методології програмування
2. Процедурна методологія програмування
3. Об'єктно-орієнтоване програмування
4. Функціональні, логічні та скриптові мови програмування
5. Класифікація мов програмування (високо/низько-рівневі, компільовані/інтерпретуючі)

Дидактичні засоби: Комп'ютерний проектор з набором експозицій до л. 7.

Література: [4].

Завдання на СРС: підготувати відповідь на контрольні запитання до л. 7.

Лекція 8. Основи об'єктно-орієнтованої методології розробки ПЗ

1. Поняття ООП, класу та об'єкту
2. Структура класу (поля, методи, конструктор, деструктор)
3. Приклади реалізації класу
4. Базові принципи ООП (абстракція, інкапсуляція, наслідування, поліморфізм)

Дидактичні засоби: Комп'ютерний проектор з набором експозицій до л. 8.

Література: [4].

Завдання на СРС: підготувати відповідь на контрольні запитання до л. 8.

Лекція 9. Універсальна мова моделювання UML

1. Класифікація типів діаграм в UML
2. Структурні діаграми (діаграма класів)
3. Поведінкові діаграми (діаграма прецедентів)
4. Діаграми взаємодії (діаграма послідовностей)

Дидактичні засоби: Комп'ютерний проектор з набором експозицій до л. 9.

Література: [5].

Завдання на СРС: підготувати відповідь на контрольні запитання до л. 9.

Лекція 10. Принципи проектування SOLID

1. Реалізація базових принципів ООП на практиці
2. Класифікація принципів SOLID
3. Принцип «єдиного обов'язку»
4. Принцип «відкритості-закритості»
5. Принцип «підстановки Ліскова»
6. Принцип «розділення інтерфейсу»
7. Принцип «інверсії залежностей»

Дидактичні засоби: Комп'ютерний проектор з набором експозицій до л. 10.

Література: [6].

Завдання на СРС: підготувати відповідь на контрольні запитання до л. 10.

Тема 5. Шаблони проектування ООП

Лекція 11. Поняття та класифікація шаблонів проектування, породжувальні шаблони

1. Поняття шаблонів проектування
2. Класифікація шаблонів проектування
3. Види породжувальних шаблонів
4. Шаблон Будівельник (Builder)
5. Шаблон Фабричний метод (Factory method)
6. Шаблон Одинак (Singleton)

Дидактичні засоби: Комп'ютерний проектор з набором експозицій до л. 11.

Література: [7].

Завдання на СРС: підготувати відповідь на контрольні запитання до л. 11.

Лекція 12. Структурні шаблони проектування

1. Види структурних шаблонів
2. Шаблон Компонувальник (Composite)
3. Шаблон Адаптер (Adapter)
4. Шаблон Декоратор (Decorator)

Дидактичні засоби: Комп'ютерний проектор з набором експозицій до л. 12.

Література: [7].

Завдання на СРС: підготувати відповідь на контрольні запитання до л. 12.

Лекція 13. Поведінкові шаблони проектування

1. Види поведінкових шаблонів
2. Шаблон Спостерігач (Observer)
3. Шаблон Ланцюжок обов'язків (Chain of responsibility)

Дидактичні засоби: Комп'ютерний проектор з набором експозицій до л. 12.

Література: [7].

Завдання на СРС: підготувати відповідь на контрольні запитання до л. 12.

Тема 6. Основи програмування мобільних пристроїв

Лекція 14. Особливості розробки ПЗ для мобільних пристроїв

1. Основи мережевої передачі даних, контроль цілісності даних, використання завадостійкого кодування
2. Поняття розробки мобільних додатків та сервісів, мобільні операційні системи (ОС)
3. Структура компонентів мобільної ОС Android, історія розвитку версій

Дидактичні засоби: Комп'ютерний проектор з набором експозицій до л. 14.

Література: [8].

Завдання на СРС: підготувати відповідь на контрольні запитання до л. 14.

Лекція 15. Практичні основи програмування на ОС Android

1. Приклад створення першого додатка для ОС Android
2. Огляд датчиків, доступних на ОС Android
3. Приклади програмування датчиків лінійного та кутового прискорення, датчику світла
4. Огляд комунікаційних інтерфейсів на ОС Android
5. Приклад реалізації сканування точок доступу WiFi
6. Приклад реалізації клієнт-сервера та передачі даних по протоколу UDP

Дидактичні засоби: Комп'ютерний проектор з набором експозицій до л. 15.

Література: [8].

Завдання на СРС: підготувати відповідь на контрольні запитання до л. 15.

Тема 7. Методи забезпечення якості розробки ПЗ

Лекція 16. Основи контролю та забезпечення якості розробки ПЗ

1. Основи контролю якості розробки ПЗ (місце в життєвому циклі розробки, поняття тестового плану та кейсу)
2. Класифікація методів тестування якості ПЗ (ручне/автоматизоване, чорного/білого ящиків, функціональне/нефункціональне)
3. Основні види функціонального тестування (модульне тестування ПЗ, реалізація на ОС Android, інтеграційні та системні тести)
4. Основні види нефункціонального тестування (тестування продуктивності та безпеки)

Дидактичні засоби: Комп'ютерний проектор з набором експозицій до л. 16.

Література: [9].

Завдання на СРС: підготувати відповідь на контрольні запитання до л. 16.

Тема 8. Перспективні технології програмування та розробки

Лекція 17. Основи штучного інтелекту та машинного навчання

1. Поняття штучного інтелекту (ШІ) та машинного навчання
2. Роль лінійної алгебри в машинному навчанні
3. Згорткові та рекурентні нейронні мережі
4. Генеративний ШІ

Дидактичні засоби: Комп'ютерний проектор з набором експозицій до л. 17.

Література: [10].

Завдання на СРС: підготувати відповідь на контрольні запитання до л. 17.

Лекція 18. Огляд розвитку технологій майбутнього

1. Прогнози розвитку технологій майбутнього
2. Огляд перспективних технологій майбутнього (Метавсесвіт, Біометрика, 6G, доповнена та віртуальна реальність, інтернет речей, квантові обчислення)

Дидактичні засоби: Комп'ютерний проектор з набором експозицій до л. 18.

Завдання на СРС: підготувати відповідь на контрольні запитання до л. 18.

5.2 Практичні заняття – 18 годин

Основні завдання циклу практичних занять ознайомитися з основними поняттями курсу Сучасні технології програмування, та навчитися застосовувати ці поняття для розробки телекомунікаційних об'єктів та систем.

Тема 1-2. Організація процесу спільної розробки програмного забезпечення (ПЗ) та основи роботи з системою контролю версій Git

Практичне заняття 1. Основи роботи з системою Git.

Основні питання:

1. Організація практичної роботи з курсу для аудиторних та самостійних занять
2. Життєвий цикл розробки ПЗ.
3. Встановлення GitBash Terminal, налаштування конфігурації Git.
4. Базові Unix команди для роботи з терміналом.
5. Базові команди роботи з Git: status, init, add, commit, log.
6. Вивчення структури Git commit.

Практичне заняття 2. Командна розробка ПЗ та управління історією змін програмного коду

1. Створення гілок паралельної розробки (branch, checkout)

2. Об'єднання гілок розробки та вирішення конфліктів об'єднання (merge)
3. Аналіз змін програмного коду (diff)
4. Тимчасове приховування змін програмного коду (git stash)
5. Навігація історією змін програмного коду (git checkout)
6. Повернення до попередніх версій ПЗ та видалення останніх змін

Тема 3. Основи роботи з платформою GitHub

Практичне заняття 3. Робота с платформою GitHub

Основні питання:

1. Реєстрація на платформі GitHub.
2. Створення репозиторія та клонування його локальної копії (init, clone).
3. Копіювання локальних змін програмного коду до віддаленого репозиторію (push).
4. Синхронізація локального репозиторію з віддаленим (fetch, pull).
5. Створення заявки на додавання змін програмного коду в загальну гілку (pull request).

Тема 4. Проектування та реалізація ПЗ з використанням принципів об'єктно-орієнтованого програмування (ООП)

Практичне заняття 4. Вступ до ООП, базові принципи та принципи SOLID

Основні питання:

1. Реалізація ієрархії класів з використанням базових принципів ООП: інкапсуляції, наслідування, поліморфізму.
2. Практичне використання принципів SOLID.
3. Виправлення реалізації ієрархії класів, яка порушує принцип «єдиного обов'язку» SOLID.
4. Виправлення реалізації ієрархії класів, яка порушує принцип «розділення інтерфейсів» SOLID.
5. Побудова діаграми класів UML.

Тема 5. Шаблони проектування ООП

Практичне заняття 5. Огляд шалонів проектування, породжувальні шаблони ООП

Основні питання:

1. Огляд класифікація шаблонів проектування ООП
2. Практична реалізація шаблону Будівельник (Builder)
3. Практична реалізація шаблону Фабричний метод (Factory method)
4. Практична реалізація шаблону Шаблон Одинак (Singleton)

Практичне заняття 6. Структурні шаблони проектування ООП

Основні питання:

1. Практична реалізація шаблону Компонувальник (Composite)
2. Практична реалізація шаблону Адаптер (Adapter)
3. Практична реалізація шаблону Декоратор (Decorator)

Практичне заняття 7. Поведінкові шаблони проектування ООП

Основні питання:

1. Практична реалізація шаблону Спостерігач (Observer)
2. Практична реалізація шаблону Шаблон Ланцюжок обов'язків (Chain of responsibility)

Тема 6. Основи програмування мобільних пристроїв

Практичне заняття 8. Огляд засобів програмування мобільних пристроїв на ОС Android

Основні питання:

1. Огляд структури ОС Android
2. Встановлення Android Studio, створення першого проекту, огляд структури проекту
3. Програмування датчиків лінійного та кутового прискорення, датчику світла
4. Практична реалізація сканування точок доступу WiFi
5. Реалізація клієнт-сервера та передачі даних по протоколу UDP

Тема 7. Методи забезпечення якості розробки ПЗ

Практичне заняття 9. Покриття програмного коду модульними тестами на ОС Android

Основні питання:

1. Огляд інтерфейсу бібліотеки junit для реалізації модульних (unit) тестів на ОС Android
2. Реалізація програмного коду на мові Java для ОС Android
3. Реалізація модульних тестів для програмного коду: позитивних та негативних
4. Запуск модульних тестів на виконання
5. Вивчення основ покриття програмного коду тестами

6. Самостійна робота студента

До самостійної роботи студентів включається підготовка до аудиторних занять шляхом опанування матеріалів лекцій, вивчення базової, додаткової літератури та виконання практичних завдань.

Тема 1-2. Організація процесу спільної розробки програмного забезпечення (ПЗ) та основи роботи з системою контролю версій Git

Самостійна практична робота 1. Основи роботи с системою Git

Ціль: отримання навичок використання базових команд Git: init, status, branch, add, commit, merge

Завдання 1: створення коммітів

Завдання 2: робота з гілками

Завдання 3: злиття гілок

Результат виконання завдання: текстовий документ з консольними логами по виконанню умов завдання

Самостійна практична робота 2. Робота з історією змін Git

Ціль: отримання навичок управління історією змін коду в Git: diff, reset, checkout

Завдання 1: аналіз різниці між змінами

Завдання 2: робота з командою stash

Завдання 3: скасування змін

Результат виконання завдання: текстовий документ з консольними логами по виконанню умов завдання

Тема 3. Основи роботи з платформою GitHub

Самостійна практична робота 3. Основи роботи з GitHub

Ціль: отримання навичок роботи з GitHub

Завдання: додавання змін в GitHub репозиторій

Результат виконання завдання: комміти, додані в основну гілку через створення затвердженого pull request

Тема 4. Проектування та реалізація ПЗ з використанням принципів об'єктно-орієнтованого програмування (ООП)

Самостійна практична робота 4. Основи ООП, базові принципи, проектування дизайну та принципи SOLID

Ціль: отримання навичок використання принципів ООП та побудови діаграми класів

Завдання 1: створення ієрархії класів з використанням принципів інкапсуляції, наслідування і поліморфізму

Завдання 2: побудова діаграми класів Завдання 1.

Результат виконання завдання: код рішення завантажений на GitHub (завдання 1) та прикріплене зображення діаграми класів (завдання 2).

Тема 5. Шаблони проектування ООП

Самостійна практична робота 5. Шаблони проектування ООП. Породжувальні шаблони

Ціль: отримання навичок використання породжувальних шаблонів ООП та побудови діаграми класів

Завдання 1: реалізація програми з використанням шаблону Будівельник (Builder).

Завдання 2: побудова діаграми класів Завдання 1.

Результат виконання завдання: код рішення завантажений на GitHub (завдання 1) та прикріплене зображення діаграми класів (завдання 2).

Самостійна практична робота 6. Шабини проектування ООП. Структурні шабини

Ціль: отримання навичок використання структурних шабини ООП та побудови діаграми класів

Завдання 1: реалізація програми з використанням одного зі структурних шабини: Композит (Composite), Адаптер (Adapter) або Декоратор (Decorator).

Завдання 2: побудова діаграми класів Завдання 1.

Результат виконання завдання: код рішення завантажений на GitHub (завдання 1) та прикріплене зображення діаграми класів (завдання 2).

Самостійна практична робота 7. Шабини проектування ООП. Поведінкові шабини

Ціль: отримання навичок використання поведінкових шабини ООП та побудови діаграми послідовностей

Завдання 1: реалізація програми з використанням одного з поведінкових шабини: Ланцюг відповідальності (Chain of responsibility) або Наглядач (Observer).

Завдання 2: побудова діаграми послідовностей Завдання 1.

Результат виконання завдання: код рішення завантажений на GitHub (завдання 1) та прикріплене зображення діаграми послідовностей (завдання 2).

Тема 6-7. Основи програмування мобільних пристроїв та методи забезпечення якості розробки ПЗ

Самостійна практична робота 8. Основи тестування програмного забезпечення. Модульне тестування

Ціль: отримання навичок тестування програмного забезпечення шляхом реалізації юніт-тестів для покриття програмного коду алгоритмів

Завдання 1: реалізація класу з алгоритмічними методами: пошук мінімального елемента масиву позитивних чисел; розрахунок суми елементів масиву, який може складатися лише з від'ємних чисел, алгоритм розрахунку N-го елемента послідовності Фібоначчі, алгоритм розрахунку сили струму на ділянці кола

Завдання 2: реалізація позитивних та негативних юніт тестів для кожного з чотирьох методів.

Завдання 3: запуск виконання юніт тестів і підрахунок покриття коду. Збереження звіту по покриттю коду юніт тестами.

Результат виконання завдання: код рішення завантажений на GitHub (завдання 1) та прикріплений звіт по покриттю коду.

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

Дисципліна вивчається студентами в ході відвідування і прослуховування лекцій, розв'язання задач на практичних заняттях та виконання домашніх практичних робіт. В ході вивчення дисципліни передбачені поточні контрольні заходи результатів навчання у вигляді опитування та тестування, модульна контрольна робота і підсумковий семестровий контроль у формі заліку. Поточні контрольні заходи здійснюються в основному у формі електронного тестування та опитування на практичних заняттях і в ході перевірки виконання домашніх практичних робіт. Підсумковий семестровий контроль проводиться у формі заліку. Загальна оцінка засвоєння студентами тем початкової дисципліни виставляється на підставі загального рейтингу кожного студента відповідно до рейтингової системи оцінювання результатів навчання.

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

8.1. Рейтинг студента з кредитного модуля розраховується виходячи із 100-бальної шкали і визначається за результатами виконання набору контрольних заходів (п.8.2.).

8.2. Контрольні заходи:

Лекційні заняття	– бал за роботу на лекції та написання конспекту – 1.5; За семестр проводиться 18 лекційних занять, кожен студент має можливість роботати на кожному з занять Максимальна кількість балів: $18 \times 1.5 = 27$
Практичні заняття	– бал за роботу на практичному занятті – 1.5; За семестр проводиться 9 практичних занять, кожен студент має можливість роботати на кожному з занять Максимальна кількість балів: $1.5 \times 9 = 13$.
Самостійні практичні роботи	– бал за одну повну вірно виконану практичну роботу зі звітом в поставлені терміни – 5; – бал за виконану практичну роботу з запізненням – 2.5; Максимальна кількість балів: $5 \times 8 = 40$.
Модульна контрольна робота (МКР)	– бал за повну відповідь на всі запитання модульної контрольної роботи №1 – 10; – бал за повну відповідь на всі запитання модульної контрольної роботи №2 – 10; – бал за відсутність відповіді чи за неправильну відповідь на всі запитання – 0; Максимальна кількість балів: 20.

8.3. Підсумковий семестровий контроль проводиться формі заліку з виставленням у відомість оцінки кожному із студентів на підставі поточного загального рейтингу студента, або проведення залікової контрольної роботи (співбесіди). Залікова контрольна робота (співбесіда) оцінюється із 100 балів. Контрольне завдання цієї роботи (співбесіди) складається з п'яти запитань з переліку, що надається для підготовки до заліку.

Кожне запитання оцінюється з 20 балів за такими критеріями:

- «відмінно» – повна відповідь (не менше 90% потрібної інформації), надані відповідні обґрунтування – 18-20 балів;
- «добре» – достатньо повна відповідь (не менше 75% потрібної інформації), є незначні неточності – 13-17 бали;
- «задовільно» – неповна відповідь (не менше 60% потрібної інформації), є деякі помилки – 7-12 балів;
- «незадовільно» – незадовільна відповідь – 0 балів.

8.4. Календарна проміжна атестація студентів проводиться на підставі поточного рейтингу студента на час проведення атестації. Якщо значення цього рейтингу не менше 30% від максимально можливого на час атестації, студент вважається атестованим. Умовою позитивної першої атестації є отримання не менше 10 балів, другої атестації – отримання не менше 50 балів.

8.5. Сума рейтингових балів, отриманих студентом протягом семестру, за умови зарахування МКР та всіх практичних робіт, переводиться до підсумкової оцінки згідно з таблицею (п.8.7.). Якщо сума балів менша за 60, студент виконує залікову контрольну роботу (співбесіду). Умовою допуску до залікової контрольної роботи (співбесіди) при зарахуванні МКР та всіх лабораторних робіт, є сума рейтингових балів не менша 33.

8.6. Студент, який у семестрі отримав більше 60 балів, але бажає підвищити свій результат, може взяти участь у заліковій контрольній роботі (пройти співбесіду).

8.7. Таблиця переведення рейтингових балів до оцінок:

<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. Політика університету

9.1. Політика щодо академічної доброчесності

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>

9.2. 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>

10. Додаткова інформація з дисципліни (освітнього компонента)

Робочу програму навчальної дисципліни (силабус):

Складено к.т.н. Педан С.І.

Ухвалено кафедрою ІТТ НН ІТС (протокол №4 від 03 листопада 2023 року)