

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ АВІАЦІЙНИЙ ФАХОВИЙ КОЛЕДЖ

Найменування центрального органу управління освітою, власника

«ЗАТВЕРДЖУЮ»

Заст. директора



Алла АДАМЕНКО

« 25 »

08

2022р.

КОНСТРУЮВАННЯ ПРОГРАМНОГО
ЗАБЕЗПЕЧЕННЯ

Назва навчальної дисципліни

ПРОГРАМА

навчальної дисципліни

підготовки

фахового молодшого бакалавра

Назва освітньо-професійного ступеня

спеціальності

121 Інженерія програмного забезпечення

Шифр і назва спеціальності

РОЗРОБЛЕНО ТА ВНЕСЕНО: Київським авіаційним фаховим коледжем

Повне найменування вищого навчального закладу

Розробники:


Підпис

М. Кіздученко – викладач

Схвалено на засіданні циклової комісії
професійної та практичної підготовки

Назва циклової комісії

Протокол № 1, від «25» 08 2022р.

Голова циклової комісії


Підпис

Сурмачук А. П.
Прізвище та ініціали

ВСТУП

Програма вивчення навчальної дисципліни «Конструювання програмного забезпечення» складена відповідно до освітньо-професійної програми підготовки фахового молодшого бакалавра спеціальності «Інженерія програмного забезпечення»

Предметом вивчення навчальної дисципліни є застосування теорії, знань і практики за умови ефективної побудови програмних систем, які задовольняють потребам замовників та користувачів; вивчення всіх фаз життєвого циклу програмної системи: аналіз потреб, розробку специфікацій, проектування, побудову, тестування, експлуатацію та супроводження.

Міждисциплінарні зв'язки: вивченню «Конструювання програмного забезпечення» передують дисципліни «Основи програмування та алгоритмічних мов», «Об'єктно-орієнтоване програмування», «Основи програмної інженерії».

Програма навчальної дисципліни складається з таких тем:

Тема 1. Предмет і зміст дисципліни

Тема 2. Планування конструювання ПЗ.

Тема 3. Проектування при конструюванні.

Тема 4-5. Проектування при конструюванні.

Тема 6-7. Ефективне використання класів.

Тема 8-9. Якість ПЗ а системна інтеграція

Тема 10. Рефакторинг ПЗ.

Тема 11. Оптимізація програмного коду.

Тема 12. Інструменти проектування програмного продукту.

Тема 13. Інструменти програмування.

Тема 14. Організація та представлення моделі проекту.

Тема 15. Рівнева організація проекту.

Тема 16-17. Проектування рівня представлення.

Тема 18. Проектування рівня бізнес-логіки.

Тема 19. Проектування рівня даних.

Тема 20. Засоби роботи з БД.

Тема 21. Сервіс-орієнтована архітектура ПЗ.

Тема 22. Web-сервіси.

Тема 23. Технологія Blockchain.

Тема 24. Дизайн рівня представлення ПЗ.

Тема 25. CASE-технології розробки ПЗ.

Тема 26. Тестування програмного забезпечення.

Тема 27. Етапи проведення тестування. Види тестів.

Тема 28. Роль тестування в життєвому циклі розробки ПЗ.

Тема 29. Керування змінами на етапі тестування.

Тема 30. Роль вимог в життєвому циклі розробки ПЗ.

Тема 31. Принципи побудови специфікації вимог.

Тема 32. Моделювання прецедентів. Діаграми прецедентів.

Тема 33. Моделювання та діаграми класів, взаємодій, станів та переходів.

1. Мета та завдання навчальної дисципліни

Метою викладання навчальної дисципліни «Конструювання програмного забезпечення» є оволодіння основними принципами конструювання програмного забезпечення (ПЗ), ознайомлення та використання сучасних технологій створення та проектування ПЗ, а також його подальшого супроводу; набуття практичних навичок самостійного написання якісного коду для розв'язання різноманітних задач у практичній діяльності.

Навчання проводиться у формі лекцій та практичних занять із застосуванням персональних комп'ютерів (ПК).

Освоєння дисципліни дозволить майбутнім фахівцям забезпечити необхідний рівень вивчення і аналізу фахових дисциплін за рахунок ефективного використання сучасних технологій конструювання ПЗ.

Основними завданнями вивчення дисципліни «Конструювання програмного забезпечення» є формування теоретичних знань та практичних навичок у відповідності з поставленою метою.

Згідно з вимогами освітньо-професійної програми студенти повинні:

знати:

- мету конструювання ПЗ;
- місце конструювання в програмній інженерії;
- необхідність попередніх умов при конструюванні ПЗ;
- основні концепції та методики проектування при конструюванні;
- ефективне використання класів;
- характеристики якості ПЗ;
- стратегій інтеграції ПЗ;
- поняття рефакторінгу;
- оцінки продуктивності коду;
- інструменти програмування;

вміти:

- ефективно планувати та проектувати конструювання якісного ПЗ;
- розробляти якісні інтерфейси та методи класів;
- створювати власні механізми тверджень і виключень;
- використовувати відлагоджувальні засоби;
- оцінювати продуктивність ПЗ.

На вивчення навчальної дисципліни відводиться **135** годин за навчальним планом.

2. Інформаційний обсяг навчальної дисципліни

Тема 1. Предмет і зміст дисципліни

Визначення конструювання програмного забезпечення. Місце конструювання ПЗ в програмній інженерії.

Тема 2. Планування конструювання ПЗ.

Визначення типу ПЗ. Формування попередніх умов для проектування та створення програмних продуктів.

Тема 3. Проектування при конструюванні.

Проблеми, пов'язані з проектуванням ПЗ. Основні концепції проектування. Компоненти проектування: евристичні принципи. Методики проектування.

Коментарі з приводу популярних методологій.

Тема 4. Ефективне використання класів

Основи класів: абстрактні типи даних. Якісні інтерфейси класів. Питання проектування та реалізації. Причини створення класів. Аспекти, специфічні для мов. Наступний рівень: пакети класів.

Тема 5. Якість ПЗ та системна інтеграція.

Характеристики якості ПЗ. Методики підвищення якості ПЗ. Важність вибору підходу до інтеграції. Частота інтеграції — поетапна чи інкрементна? Стратегії інкрементної інтеграції. Щоденна збірка та димові тести.

Тема 6. Рефакторинг.

Поняття рефакторингу, окремі види рефакторингу, безпечний рефакторинг, стратегії рефакторингу стратегії рефакторингу.

Тема 7. Стратегії та методики оптимізації коду.

Поняття оптимізації коду, оцінка продуктивності коду. Логіка, цикли, зміни типів даних, вирази та методи, переробка коду на низькому рівні.

Тема 8. Інструменти програмування.

Інструменти для проектування, інструменти для роботи з початковим кодом, інструменти для роботи з виконавчим кодом, інструменти та середовища.

Тема 9. Рівнева організація прикладу.

Пошарова організація прикладу та виділення рівнів Представлення, Бізнес-логіки та Даних. Дизайн рівневої структури. Визначення необхідних рівнів, прийняття рішення про розподіл компонентів, визначення правил взаємодії між рівнями, визначення інтерфейсу між рівнями, вибір стратегій реалізації, Вибір протоколів взаємодії.

Тема 10: Проектування рівня представлення.

Компоненти рівня представлення. Кешування, Комунікації, Композиція, Керування виключеннями, Специфічні проблеми рівню представлення. Шаблони проектування.

Тема 11. Проектування рівня бізнес-логіки.

Компоненти рівня бізнес-логіки. Специфічні проблеми рівню бізнес-логіки. Шаблони проектування.

Тема 12: Проектування рівня даних.

Рекомендації що до проектування рівня даних. Компоненти рівня даних. Специфічні проблеми рівню даних. Шаблони проектування. Засоби роботи з компонентами рівня даних.

Тема 13. Проектування рівня сервісів.

Компоненти рівню сервісів. Специфічні проблеми рівню даних. Шаблони проектування.

Тема 14. Сервіс-орієнтована архітектура ПЗ.

Визначення. Концепції. Вигоди використання. Стандарти. Роль XML. Етапи циклу життя SOA. Архітектурний шаблон SOA. Дизайн рівня сервісів. Компоненти рівню сервісів.

Тема 15. WEB-сервіси.

Концепції. Довідка з проектування та принципи проектування. Приклади. XML, JSON. Системи обміну повідомленнями. Базові поняття протоколу HTTP та стеку протоколів TCP/IP. Віддалений виклик процедур (RPC) та GraphQL.

Тема 16. Засоби дизайну рівня представлення ПЗ.

Інструменти для проектування рівня представлення ПЗ. Графічні засоби проектування інтерфейсів.

Тема 17. CASE-технології розробки ПЗ.

Порівняння життєвого циклу програмного забезпечення при традиційній розробці і розробці з використанням CASE-засобів. Концептуальні основи CASE-технології.

Тема 18. CASE-технології розробки ПЗ.

Тема 19. Тестування програмного забезпечення.

Принципи та види тестування ПЗ.

Тема 20. Етапи проведення тестування.

Види тестів. Планування перевірки відповідності (validation planning). Основи тестування, включаючи створення тестового плану та генерацію тестів.

Тема 21. Роль тестування в життєвому циклі розробки ПЗ.

Тестування методом «чорного ящика» та «білого ящика». Тестування елементів, інтеграційне, системне тестування та перевірка відповідності.

Тема 22. Керування змінами на етапі тестування.

Об'єктно-орієнтоване тестування. Перевірка рівних за рангом та інспекції (peer reviews, inspections)

Тема 23. Роль вимог в життєвому циклі розробки ПЗ. Принципи побудови специфікації вимог.

Виявлення потреб. Методи моделювання та аналізу потреб. Функціональні та нефункціональні потреби. Прототипування. Основні концепції методів формальної специфікації.

Тема 24. Принципи побудови специфікації вимог.

Виявлення потреб. Методи моделювання та аналізу потреб. Функціональні та нефункціональні потреби. Прототипування. Основні концепції методів формальної специфікації.

Тема 25. Моделювання прецедентів.

Діаграми прецедентів. Будова діаграми прецедентів. Відношення на діаграмі прецедентів.

Тема 26. Моделювання та діаграми класів, взаємодій, станів та переходів. Клас (class) в мові UML.

Відношення залежності (dependency relationship). Відношення асоціації (association relationship).

3. Рекомендована література

Основна література

1. Бабенко Л.П., Лаврищева К.М, Основи програмної інженерії.–К.: Знання, 2001.–269с.
2. Лаврищева К.М. Програмна інженерія. – Підручник.–К.: Академперіодика, 2008.–415с.
3. Стандарт ISO/IEC 9126 , ДСТУ Програмна інженерія. Якість продукту (частина 1-6). ISO 9000 (1-4). Інформаційні технології. Стандарти з керування якістю та забезпечення якості.
4. ДСТУ ISO 9000:2007. Системи управління якістю. Основні положення та словник термінів. – К.: Держспоживстандарт, 2008. – [Чинний від 2008-01-01] – 35 с.– (Державний стандарт).

Додаткова література

1. Піхлер Р. Agile продукт-менеджмент за допомогою Scrum / Роман Піхлер. – Київ: Фабула, 2019. – 128 с.
2. Арлоу Дж. UML 2 та Уніфікований процес. Практичний об'єктно-орієнтований аналіз та проектування; перев. з англ. Текст / Дж. Арлоу, А. Нейштадт - Символ-Плюс, 2007
3. Богс У., Богс М. UML u Rational Rose.- вид. Лорі, 2000.-580с.
4. Ерік Фрімен. Head First. Патерни проектування / Ерік Фрімен, Елізабет Робсон. – Київ: Фабула, 2020. – 672 с.
5. Matthew David. Learning DevSecOps. Integrating Continuous Security Across Your Organization / Matthew David., 2022. – 180 с.
6. Роб Коул. Блискучий Agile / Роб Коул, Едвард Скотчер. – Київ: Фабула, 2020. – 192 с.
7. Пол Дж. Філдінг. Як керувати проектами / Пол Дж. Філдінг. – Київ: Фабула, 2020. – 240 с.
8. Лаврищева Є.М. Становлення та розвиток модульно-компонентної інженерії програмування в Україні //Препринт 2008–1.–Ін-т кібернетики ім. В.М. Глушкова, 33с.

9. Лаврищева К.М. Генерувальне програмування програмних систем і їх сімейств // Проблеми програмування.– 2009, №1.–с.3-16.
10. Ларман К. Застосування UML 2.0 та шаблонів проектування; пров. з англ. Текст/К Ларман - М.: Вільямс, 2009.
11. Роберт С. Мартін. Чистий код / Роберт С. Мартін. – Київ: Фабула, 2019. – 448 с.
12. Felix Gessert. Fast and Scalable Cloud Data Management / Felix Gessert, Wolfram Wingerath, Norber. – Люксембург: Springer, 2021. – 208 с.
13. Роберт С. Мартін. Чиста архітектура. Мистецтво розробки програмного забезпечення / Роберт С. Мартін. – Київ: Фабула, 2019. – 368 с.
14. Фаулер М. Предметно-орієнтовані мови програмування; пер. з англ. Текст / М. Фаулер, Р. Парсонс - М: Вільямс, 2011.
15. Хасан Р. UML. Проектування систем реального часу, розподілених та паралельних додатків, 2-ге видання; пер. з англ. Текст / Г. Хасан - ДМК Прес, 2011.
16. McCabe : Complexity Measure, IEEE Transactions on Software Engineering, Volume 2, No 4, pp 308-320, December 1976
17. Шаббір Чаллавала. MySQL 8 для великих даних / Шаббір Чаллавала, Джадіп Лакхатарія, Чин., 2018. - 226 с.
18. Кай Ф. Л. Наддержави штучного інтелекту / Фу Лі Кай., 2020. – 304 с.
19. Карл Чатфілд. Microsoft Project 2019. Шаг за шагом / Карл Чатфілд, Тимоти Д. Джонсон, Синди Л., 2021. – 570 с.

4. Форма підсумкового контролю успішності навчання (Екзамен).

5. Засоби діагностики успішності навчання (Комплексний державний екзамен).