



Силабус освітнього компонента Програма навчальної дисципліни



Аналіз та тестування програмних систем

Шифр та назва спеціальності
122 – Комп'ютерні науки

Інститут
ННІ Комп'ютерного моделювання, прикладної фізики та математики

Освітня програма
Комп'ютерні науки. Моделювання, проектування та комп'ютерна графіка

Кафедра
Комп'ютерне моделювання процесів та систем (162)

Рівень освіти
Бакалавр

Тип дисципліни
Профільований пакет 1, Вибіркова

Семестр
7

Мова викладання
Українська

Викладачі, розробники



Іванченко Ксенія Вікторівна

kseniia.ivanchenko@khpі.edu.ua

Кандидат технічних наук, доцент, доцент кафедри комп'ютерного моделювання процесів та систем

Автор та співавтор понад 30 наукових та методичних публікацій. Курси: "Проектування програмних систем", "Кросплатформене програмування", "Аналіз та тестування програмних систем", "Програмування мобільних пристроїв".

[Детальніше про викладача на сайті кафедри](#)

Загальна інформація

Анотація

Курс присвячений основам тестування програмних продуктів з використанням міжнародних стандартів. Розглянуто базові поняття та визначення у сфері забезпечення якості тестування програмного забезпечення, критерії відбору тестів, огляд типів тестування, аналіз процесу тестування, артефакти та документація, інформаційні технології для аналізу та тестування програмних систем.

Мета та цілі дисципліни

Сформувати у студентів сучасного рівня інформаційної та програмістської культури з основ теорії якості програмного забезпечення та тестування, оволодіння базовими методами побудови тестів та автоматизованими засобами тестування, процесами управління якістю програмного забезпечення, набуття практичних навичок самостійного тестування за стандартами якості програмного забезпечення і створення якісного програмного забезпечення для розв'язання різноманітних задач в практичній діяльності.

Формат занять

Лекції, лабораторні роботи, самостійна робота, консультації. Підсумковий контроль – іспит.

Компетентності

ЗК1: Здатність до абстрактного мислення, аналізу та синтезу.

ЗК2: Здатність застосовувати знання у практичних ситуаціях.

ЗК3: Знання та розуміння предметної області та розуміння професійної діяльності.

ЗК12: Здатність оцінювати та забезпечувати якість виконуваних робіт.

СК7: Здатність застосовувати теоретичні та практичні основи методології та технології моделювання для дослідження характеристик і поведінки складних об'єктів і систем, проводити обчислювальні експерименти з обробкою й аналізом результатів.

СК8: Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.

СК10: Здатність застосовувати методології, технології та інструментальні засоби для управління процесами життєвого циклу інформаційних і програмних систем, продуктів і сервісів інформаційних технологій відповідно до вимог замовника.

Результати навчання

ПР5: Проектувати, розробляти та аналізувати алгоритми розв'язання обчислювальних та логічних задач, оцінювати ефективність та складність алгоритмів на основі застосування формальних моделей алгоритмів та обчислюваних функцій.

ПР9: Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук.

ПР10: Використовувати інструментальні засоби розробки клієнт-серверних застосувань, проектувати концептуальні, логічні та фізичні моделі баз даних, розробляти та оптимізувати запити до них, створювати розподілені бази даних, сховища та вітрини даних, бази знань, у тому числі на хмарних сервісах, із застосуванням мов веб-програмування.

ПР11: Володіти навичками управління життєвим циклом програмного забезпечення, продуктів і сервісів інформаційних технологій відповідно до вимог і обмежень замовника, вміти розробляти проектну документацію (техніко-економічне обґрунтування, технічне завдання, бізнес-план, угоду, договір, контракт).

Обсяг дисципліни

Загальний обсяг дисципліни 150 год. (5 кредитів ECTS): лекції – 32 год., лабораторні роботи – 32 год., самостійна робота – 86 год.

Передумови вивчення дисципліни (пререквізити)

Дисципліна базується на знаннях та компетенціях, що набуває здобувач вищої освіти під час вивчення дисциплін: Об'єктно-орієнтоване програмування та проектування, Технології програмування, Проектування програмних систем, Кросплатформене програмування.

Особливості дисципліни, методи та технології навчання

Лекції проводяться інтерактивно з використанням мультимедійних технологій. Для виконання лабораторних занять використовуються безкоштовне середовище розробки IntelliJ IDEA Community Edition та засіб для проведення навантажувального тестування Apache JMeter. На заняттях використовується проектний підхід до навчання. Навчальні матеріали доступні студентам через OneDrive.

Програма навчальної дисципліни

Теми лекційних занять

Тема 1. Введення в тестування

Тестування, його важливість. Історія розвитку тестування. QA та QC. Цілі та задачі тестування. Принципи тестування.

Тема 2. Класифікації видів та методів тестування програмного забезпечення

Тестування прототипу. Підходи до тестування програмного забезпечення. Рівні тестування. Поділ за часом проведення тестування. Поділ за виконанням коду. Поділ за ступенем автоматизації процесу тестування. Види тестування.

Тема 3. Тестування програмних систем у різних методологіях розробки: від Waterfall до Agile.

Особливості тестування програмних систем у різних методологіях розробки. Вплив тестування на процес розробки. Адаптація методів тестування до різних умов.

Тема 4. Тестування вимог

Види вимог. Класифікація вимог. Види документів з вимогами до ПЗ. Параметри тестування вимог. Основні засади тестування вимог. Характеристика якісних вимог.

Тема 5. Тестова документація

Що таке тестова документація. Зовнішня та внутрішня документація. Test plan. Use case. Test case. Check list. Джерела інформації для тестувальника.

Тема 6. Поняття дефекту

Поняття дефекту. Основні види дефектів. Життєвий цикл дефекту. Атрибути дефекту. Закриття дефектів. Системи відстеження дефектів.

Тема 7. Особливості тестування десктоп додатків

Термінологія програмного забезпечення. Операційна система, розрядність. Файлова система. Облікові записи. Тестування інсталяції. Робота з віртуальними машинами.

Тема 8. Особливості тестування на мобільних пристроях.

Порівняння сучасних мобільних платформ. Роздільна здатність екрану. Метрики мобільних додатків. Види мобільних додатків. Тестування на основі емуляції. Тестування з урахуванням пристроїв. Тестування в хмарі. Види тестування мобільних додатків.

Тема 9. Нефункціональні види тестування

Види тестування за цілями. Види нефункціонального тестування. Тестування установки. Тестування зручності користування. Конфігураційне тестування.

Тема 10. Системи зберігання та керування тестовими випадками.

Переваги Test management system. Види популярних TMS. Основні властивості та можливості TestRail.

Тема 11. Тестування продуктивності

Основні види тестування продуктивності. Тестування навантаження. Стрес-тестування. Тестування стабільності. Тестування на відмову та відновлення. Об'ємне тестування. Створення сценарію в залежності від виду тестування. Інструмент для проведення навантажувального тестування Apache JMeter. Основні можливості та компоненти.

Тема 12. Автоматизація тестування: цілі та сфери застосування

Чому потрібна автоматизація тестування. Які тестування можна автоматизувати. Як впроваджувати автоматизацію тестування. Рівні тестування.

Тема 13. Перші кроки налаштування тестувального фреймворку.

Selenium, огляд основних можливостей. Selenium WebDriver та його методи. Методи Selenium WebElement. Локатори та пошук елементів. Selenium implicit&explicit waiters.

Тема 14. Тестування API. Основи Postman

Піраміда тестування. Monolith vs Microservices. Що таке API. Типи API. Переваги тестування API. POSTMAN.

Тема 15. Підходи до автоматизації тестування

Test Driven Development (TDD). Data-Driven Testing (DDT). Keyword Driven Testing (KDT). Behavior-Driven Development(BDD). Налаштування Cucumber.

Тема 16. Підведення підсумків

Узагальнююча лекція. Приклади тестування реального ПЗ.

Теми практичних занять

Не передбачено навчальним планом.

Теми лабораторних робіт

Лабораторна робота 1-2.

Види тестування. Аналіз класів еквівалентності та граничних значень.

Лабораторна робота 3.

Тестування вимог.

Лабораторна робота 5-6.

Тестування програмного забезпечення: розробка тестів.

Лабораторна робота 7-8.

Пошук та документування дефектів.

Лабораторна робота 9.

Особливості тестування мобільних пристроїв

Лабораторна робота 10.

Особливості тестування десктопних застосунків

Лабораторна робота 11.

Використання системи зберігання та керування тестовими кейсами Test Rail

Лабораторна робота 12.

Тестування продуктивності. Використання Apache JMeter.

Лабораторна робота 13.

Створення тестів з Selenium WebDriver.

Лабораторна робота 14.

Автоматизація тестування API з Postman.

Лабораторна робота 15.

Презентація та захист індивідуальних розрахункових робіт.

Лабораторна робота 16.

Проведення підсумкового тестування.

Самостійна робота

Курс передбачає виконання індивідуального розрахункового завдання з проведення тестування програмного продукту за власним вибором. Результат розробки оформлюється у презентацію, з обов'язковим захистом в кінці курсу. Оцінка виставляється за результатом відкритого захисту розрахункового завдання перед комісією, яка складається з викладача та інших студентів групи. Студентам також рекомендуються додаткові матеріали (відео, статті) для самостійного вивчення та аналізу.

Література та навчальні матеріали

- 1 Авраменко А.С., Авраменко В.С., Косенюк Г.В. Тестування програмного забезпечення. Навчальний посібник. Черкаси: ЧНУ імені Богдана Хмельницького, 2017. 284 с.
- 2 Li K. Effective Software Test Automation, 2021.
- 3 Graham D., Veenendaal E. Foundations of Software Testing, 2018.
- 4 Liang Y., Collins A. Selenium WebDriver: From Foundations to Framework, 2021.
- 5 Myers G.J. The Art of Software Testing, 2011.
- 6 Crispin L., Gregory J. Agile Testing: A Practical Guide for Testers and Agile Teams, 2009.
- 7 Martin R.C. Clean Code: A Handbook of Agile Software Craftsmanship, 2008.

Система оцінювання

Критерії оцінювання успішності студента та розподіл балів

100% підсумкової оцінки складаються з результатів оцінювання у вигляді підсумкового тестування (20%) та поточного оцінювання (80%).

Підсумкове тестування: 40 тестових питань, наявні питання з декількома відповідями та відкриті питання.

Поточне оцінювання: виконання лабораторних робіт (60%) та розрахункове завдання (20%). У разі непогодження з отриманою оцінкою 100% підсумкової оцінки складаються з результатів оцінювання у вигляді екзамену. Обов'язковою умовою допуску до екзамену є виконання всіх лабораторних робіт.

Екзамен: письмове завдання (2 запитання з теорії та практичне завдання) та усна доповідь.

Шкала оцінювання

Сума балів	Національна оцінка	ECTS
90–100	Відмінно	A
82–89	Добре	B
75–81	Добре	C
64–74	Задовільно	D
60–63	Задовільно	E
35–59	Незадовільно (потрібне додаткове вивчення)	FX
1–34	Незадовільно (потрібне повторне вивчення)	F

Норми академічної етики і політика курсу

Студент повинен дотримуватися «Кодексу етики академічних взаємовідносин та доброчесності НТУ «ХПІ»: виявляти дисциплінованість, вихованість, доброзичливість, чесність, відповідальність. Конфліктні ситуації повинні відкрито обговорюватися в навчальних групах з викладачем, а при неможливості вирішення конфлікту – доводитися до відома співробітників дирекції інституту. Нормативно-правове забезпечення впровадження принципів академічної доброчесності НТУ «ХПІ» розміщено на сайті: <http://blogs.kpi.kharkov.ua/v2/nv/akademichna-dobrochesnist/>

Погодження

Силабус погоджено

28.08.2023

Завідувач кафедри
Дмитро БРЕСЛАВСЬКИЙ

28.08.2023

Гарант ОП
Оксана ТАТАРІНОВА