Силабус освітнього компонента
Програма навчальної дисципліни

Об'єктно-орієнтоване програмування ч2

Шифр та назва спеціальності
123 – Комп'ютерна інженерія

Інститут
ННІ комп'ютерних наук та інформаційних технологій

Освітня програма
Сучасне програмування, мобільні пристрої та комп'ютерні ігри (інноваційний кампус/ Прикладна комп'ютерна інженерія)

Кафедра
Комп'ютерна інженерія та програмування (326)

Рівень освіти
Бакалавр

Тип дисципліни
Профільна підготовка

Семестр
4

Мова викладання
Українська

Викладачі, розробники

Молчанов Георгій Ігорович,
heorhii.molchanov@khpi.edu.ua;
старший викладач кафедри комп'ютерної інженерії та програмування
Основні курси: «Об'єктно-орієнтоване програмування ч1», «Об'єктно-орієнтоване програмування ч2», «Стандартна платформа Java».
Детальніше про викладача на сайті кафедри

Загальна інформація

Анотація

«Об'єктно-орієнтоване програмування» – один з курсів профільної підготовки, що продовжує фундаментальну підготовку бакалаврів за освітньою програмою – «Сучасне програмування, мобільні пристрої та комп'ютерні ігри (інноваційний кампус)». Вона формує фахівця за освітньою кваліфікацією бакалавра з комп'ютерної інженерії. Предметом дисципліни є вивчення методів об'єктно-орієнтованого підходу до побудови програмних продуктів. Вивчаються основні питання інженерії програмного забезпечення у контексті ООП, розглядаються різні способи вирішення типових задач об'єктно-орієнтованого проектування на основі шаблонів (патернів) проектування. Дисципліна забезпечує теоретичну та практичну підготовку для впровадження методів ООП при вирішенні програмних задач, необхідних для підвищення майстерності майбутнього спеціаліста у розробці програмного забезпечення з використанням сучасних технологій.

Мета та цілі дисципліни

Метою викладання дисципліни є: вивчення базових концепцій об'єктно-орієнтованого програмування (ООП); одержання студентами теоретичних знань про реалізацію загальних методів ООП; одержання теоретичних знань про методи вирішення типових задач об'єктно-орієнтованого проектування на основі шаблонів проектування; знайомство студентів зі спеціалізованими середовищами розробки об'єктно-орієнтованих програм; одержання студентами практичних навичок рішення прикладних задач методами ООП.

Формат занять

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

Компетентності

ФК 3. Здатність створювати системне та прикладне програмне забезпечення комп'ютерних систем та мереж.
ФК 15. Здатність аргументувати вибір методів розв'язування спеціалізованих задач, критично оцінювати отримані результати та захищати прийняті рішення..

Результати навчання

ПРН 6. Вміти застосовувати знання для ідентифікації, формулювання і розв'язування технічних задач спеціальності, використовуючи методи, що є найбільш придатними для досягнення поставлених цілей.
ПРН 8. Вміти системно мислити та застосовувати творчі здібності до формування принципово нових ідей.
ПРН 13. Вміти ідентифікувати, класифікувати та описувати роботу комп'ютерних систем та їх компонентів.

Обсяг дисципліни

Загальний обсяг дисципліни 120 год. (4 кредити ECTS): лекції – 32 год., лабораторні роботи – 16 год., самостійна робота – 72 год.

Передумови вивчення дисципліни (пререквізити)

Для вивчення курсу студенти потребують базових знань з дисциплін: «Програмування» , достатніх для:
- вивчення методів розробки програмних комплексів з використанням засобів ООП;
- вирішення задач програмування з використанням різних мов, що підтримують ООП.
- оволодіння спеціалізованими середовищами розробки об'єктно-орієнтованих програм;
- застосовування різних шаблонів проектування при рішенні типових задач з використанням ООП;
- ознайомлення з напрямками розвитку технологій проектування сучасних програмних комплексів та сучасними методами розробки складних систем з використанням ООП;
- отримання навичок об'єктно-орієнтованого аналізу, проектування, розробки, розгортання, налаштування та експлуатації програмних засобів.

Крім того курс є базовим для вивчення наступних дисциплін згідно навчального плану: «Проектування мобільних застосунків», «Проектування серверних застосунків», «Паралельні та розподілені системи».

Особливості дисципліни, методи та технології навчання

Презентація, лекція-бесіда, лекція-візуалізація, навчальна дискусія, мозкова атака, кейс-метод, демонстрування, самостійна робота, метод порівняння, метод узагальнення, метод конкретизації, метод виокремлення основного, обговорення, робота над помилками.
Вивчення курсу потребує використання програмного забезпечення Java Development Kit (JDK), Eclipse IDE for Java Developers, крім загально вживаних програм і операційних систем.
Мультимедійна дошка, проектор.

Програма навчальної дисципліни

Теми лекційних занять

Тема 1. Колекції в Java.
Java Collections Framework. Базові інтерфейси. Інтерфейс Collection. Реалізації інтерфейсу List. Реалізації інтерфейсу Set. Реалізації інтерфейсу Queue. Реалізації інтерфейсу Map. Застарілі колекції. Синхронізовані колекції. Масиви або колекції.
Тема 2. Анотації в Java.
Анотації. Супер-інтерфейс для всіх анотацій java.lang.annotation.Annotation. Застосування анотації. Політика утримання анотації. Вбудовані анотації. Приклад анотацій.
Тема 3. Параметризація в Java. Інформація про типи.
Generic-класи. Кортежі (tuple). Параметризовані методи. Підхід C++. Підхід Java.

Інформація про типи. Java Reflection API. Объект Class. RTTI. Ініціалізація класу. Рефлексія і RTTI. Динамічний код. Бібліотека java.lang.reflect.

Тема 4. Основні принципи об'єктно-орієнтованого дизайну.

Чотири правила простого дизайну: виконуються всі тести; немає логіки, що дублюється; наміри втілюються; найменша кількість класів і методів. Підхід "MIT approach". Ознаки поганого проекту. Жорсткість. Крихкість. Нерухомість. В'язкість. Невиправдана складність. Невизначеність. SOLID. Принцип єдиної відповідальності (The Single Responsibility Principle). Принцип відкритості / закритості (The Open Closed Principle). Принцип підстановки Барбари Лісков (The Liskov Substitution Principle). Принцип розділення інтерфейсу (The Interface Segregation Principle). Принцип інверсії залежностей (The Dependency Inversion Principle).

Тема 5. Обробка виключних ситуацій.

Java Exception Handling. Ієрархія виключень. Виключення, що перевіряються (Checked Exception). Виключення, що не перевіряються (Unchecked Exceptions). Виключення типу Error. Виключення типу RuntimeException. Catching and Handling Exceptions. finally блок. try-with-resources.

Тема 6. Регулярні вирази в Java.

Java Regular Expressions. Визначення. Приклад. Символьні класи. Заздалегідь визначені класи. Межі. Обмежувачі рядків. Квантіфікатори. Понаджадібні квантіфікатори. Логічні операції. Групи. Екранування символів. Попереджуючий перегляд вперед. Перегляд назад. Режими. Пакет java.util.regex. Клас Pattern. Скомпільоване уявлення регулярного виразу. Константи і еквівалентні їм прапорці. Методи matches і split. Клас Matcher. Методи для заміни тексту в зазначеному рядку. Клас PatternSyntaxException. Клас java.lang.String.

Тема 7. Паралельне виконання в Java.

Multi-threading. Потік виконання. Клас java.lang.Thread. Головний потік. Імена потоків. Interface Runnable. Створення потоку. Запуск потоку. Метод start. Завершення виконання потоку. Метод Thread.sleep. МетодThread.isAlive. Синхронізація. Інструкція synchronized. Метод Thread.join. Метод Object.wait. Методи Object.notifyі Object.notifyAll. Взаємне блокування. Стану потоків.

Тема 8. Уніфікована мова моделювання UML.

Що таке UML. Чим не є UML. Історія розвитку UML. Призначення UML. Простота і наочність. Способи використання UML. Діаграма використання. Елементи моделі. Структурні сутності. Приклад структурних сутностей. Поведінкові сутності. Сутності що групує і анотаційна. Відносини. Класифікація діаграм. Ієрархія типів діаграм для UML 1. Діаграми, додані в UML 2. Ієрархія типів діаграм для UML 2. Загальний шаблон уявлення діаграми. Загальні діаграми. Дескриптори і Класифікатори. Діаграма класів. Стандартні стереотипи класів. Атрибути і операції. Відносини на діаграмі класів. Стосунки узагальнення. Асоціації. Агрегація і композиція.

Теми практичних занять

Практичні заняття в рамках дисципліни не передбачені.

Теми лабораторних робіт

Тема 1. Розробка власних контейнерів.

Набуття навичок розробки власних об'єктів-контейнерів.

Тема 2. Поведінковий патерн проектування Ітератор.

Розробка об'єктів, що ітеруються для збереження даних користувача, застосування ітераторів.

Тема 3. Серіалізація/десеріалізація об'єктів.

Тривале зберігання та відновлення стану об'єктів. Ознайомлення з принципами серіалізації та десеріалізації об'єктів.

Тема 4. Бібліотека класів користувача.

Розробка та використання бібліотек класів користувача.

Тема 5. Основи введення/виведення Java SE.

Оволодіння навичками управління введенням та виведенням даних з використанням класів платформи Java SE.

Тема 6. Параметризація в Java.

Вивчення принципів параметризації в Java. Розробка параметризованих класів та методів.

Тема 7. Параметризовані контейнери.

Створення власних об'єктів-контейнерів, що параметризується (Generic Type), для реалізації колекції domain-об’єктів.

Тема 8. Обробка параметризованих контейнерів.

Розширення функціональності параметризованих класів. Розробка параметризованих методів (Generic Methods) для обробки колекцій об'єктів згідно прикладної задачі.

Тема 9. Основи використання регулярних виразів.

Ознайомлення з принципами використання регулярних виразів для перевірки рядка на відповідність шаблону.

Тема 10. Перевірка коректності (валідація) даних.

Використання регулярних виразів для перевірки коректності (валідації) даних, що вводяться, створених domain-об'єктів розробленого контейнера.

Тема 11. Регулярні вирази. Обробка тексту.

Ознайомлення з принципами використання регулярних виразів для обробки тексту.

Тема 12. Паралельне виконання. Multithreading.

Ознайомлення з моделлю обчислювальних потоків Java.

Тема 13. Перетворення програми для забезпечення багатопоточності.

Організація паралельного виконання декількох частин програми.

Тема 14. Multithreading. Ефективність використання.

Вимірювання часу паралельних та послідовних обчислень. Демонстрація ефективності паралельної обробки.

Тема 15. Колекції в Java.

Ознайомлення з бібліотекою колекцій Java SE. Використання колекцій для розміщення об'єктів розроблених класів.

Тема 16. Модульне тестування.

Розробка модульних тестів з використанням JUnit 5.

Самостійна робота

Опрацьовування лекційного матеріалу.
Підготовка до лабораторних робіт та модульних контролів.
Самостійне вивчення тем та питань, які не викладаються на лекційних заняттях: шаблони об'єктно-орієнтованого проектування: Proxy, Adapter, Decorator, Bridge, Chain of Responsibilit, Strategy.

Література та навчальні матеріали

ОСНОВНА ЛІТЕРАТУРА

1. The Java Programming Language. Ken Arnold, James Gosling, David Holmes
ISBN: 9780132761680
2. Core Java Volume I - Fundamentals. Cay Horstmann
ISBN: 9780135166307
3. Java Examples in a Nutshell. David Flanagan
ISBN: 9780596006204
4. Core Java Volume II - Advanced Features. Cay Horstmann
ISBN: 9780135166314
5. Thinking in Java. Bruce Eckel
ISBN: 9780131872486
6. Effective Java. Joshua Bloch
ISBN: 9780134685991
7. Design Patterns: Elements of Reusable Object-Oriented Software. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides
ISBN: 9780201633610, 9780201634983

ДОДАТКОВА ЛІТЕРАТУРА

8. Applied Java Patterns. Stephen Stelting, Olav Maassen
ISBN: 9780130935380
9. Bug Patterns In Java. Eric Allen
ISBN: 9781590590614
10. Bitter Java. Bruce A. Tate
ISBN: 9781930110434

ІНФОРМАЦІЙНІ РЕСУРСИ В ІНТЕРНЕТІ

1. Java Language Specification (Java SE 21 Edition)
https://docs.oracle.com/javase/specs/jls/se21/jls21.pdf
2. JDK 21 Documentation
https://docs.oracle.com/en/java/javase/21/
3. The Java Tutorials Bundle
https://www.oracle.com/java/technologies/javase/java-tutorial-downloads.html
4. Java SE Client Technologies: JavaFX
https://docs.oracle.com/javase/8/javase-clienttechnologies.htm
5. Introduction to Java programming - Tutorial
https://www.vogella.com/tutorials/JavaIntroduction/article.html
6. Using the Eclipse IDE for Java programming - Tutorial
https://www.vogella.com/tutorials/Eclipse/article.html
7. Eclipse Shortcuts - Tutorial
https://www.vogella.com/tutorials/EclipseShortcuts/article.html
8. Java Debugging with Eclipse - Tutorial
https://www.vogella.com/tutorials/EclipseDebugging/article.html
9. Introduction to Checkstyle for checking Java code quality - Tutorial
https://www.vogella.com/tutorials/Checkstyle/article.html
10. Java Tutorials
https://docs.oracle.com/javase/tutorial/index.html
11. How to Write Doc Comments
http://www.oracle.com/technetwork/java/javase/documentation/index-137868.html
12. Google Java Style Guide
https://google.github.io/styleguide/javaguide.html

Система оцінювання

Критерії оцінювання успішності студента та розподіл балів

Оцінювання проводиться за 100-бальною шкалою. Бали нараховуються за наступним співвідношенням:
модульні контролі - 40 балів; практичні заняття - 30 балів; залік - 30 балів.

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	Дата погодження, підпис 22.04.2024	Завідувач кафедри Олександр ЗАКОВОРОТНИЙ
	Дата погодження, підпис 22.04.2024	Гарант ОП Олександр ЗАКОВОРОТНИЙ