



**UIT**  
TRƯỜNG ĐẠI HỌC  
CÔNG NGHỆ THÔNG TIN

# **PHƯƠNG PHÁP PHÁT TRIỂN PHẦN MỀM HƯỚNG ĐỐI TƯỢNG**

## **BUỔI 1 GIỚI THIỆU MÔN HỌC**

**GVGD: ThS. Lê Thanh Trọng**

# Nội dung

1. Giới thiệu chung
2. Hình thức đánh giá
3. Công cụ thực hành
4. Các bài học
5. Tài liệu tham khảo
6. Q&A

# Giới thiệu chung

## ❖ Mục tiêu môn học

- Cung cấp kiến thức liên quan đến việc phát triển phần mềm theo tiếp cận hướng đối tượng
- Giúp người học nắm được phương pháp, kỹ thuật, quy trình, công cụ để xây dựng phần mềm theo hướng đối tượng, đặc biệt trong giai đoạn phân tích và thiết kế

## ❖ Số đơn vị học trình

- 4 tín chỉ (3 tín chỉ lý thuyết + 1 tín chỉ thực hành)

## ❖ Môn học trước

- Lập trình trực quan
- Nhập môn công nghệ phần mềm

# Giới thiệu chung

## ❖ Giảng viên lý thuyết:

Lê Thanh Trọng ([tronglt@uit.edu.vn](mailto:tronglt@uit.edu.vn))

## ❖ Giảng viên phụ trách hướng dẫn thực hành:

- Thầy Đặng Việt Dũng ([dungdv@uit.edu.vn](mailto:dungdv@uit.edu.vn))
- Cô Trần Hạnh Xuân ([xuanth@uit.edu.vn](mailto:xuanth@uit.edu.vn))

# Hình thức kiểm tra và đánh giá

- ❖ Quá trình: 50%
  - Chuyên cần tại lớp: 20%
  - Bài tập về nhà: 20%
  - Seminar: 10% (từ tuần 9)
- ❖ Cuối kì: 50% (Đồ án)
- ❖ Nhóm: 4 sinh viên
- ❖ Kiểm tra tiến độ giữa kỳ: tuần 8 - 9

# Tiêu chí đồ án (50%)

## ❖ **4Đ** (8đ)

- Đủ (3đ)
- Đúng (2đ)
- Đẹp (2đ)
- Độc (1đ): tính năng mới, công nghệ mới

## ❖ **2L** (2đ)

- Làm việc nhóm (1.5đ)
- Language (0.5đ)

# Công cụ thực hành

- ❖ Lược đồ UML: Rational Rose, Visual Paradigm, StartUML,...
- ❖ Thể loại ứng dụng: Web app, Mobile app, Desktop app
- ❖ Database: SQL Sever, MySQL, Oracle, PostgreSQL, MariaDB, MongoDB, ...

## Chương 1: Tổng quan về phát triển phần mềm

- Tổng quan về phần mềm
- Chất lượng phần mềm
- Các quy trình phần mềm và lựa chọn
  - Quy trình thác nước
  - Quy trình thác nước cải tiến
  - Quy trình Prototype
  - Quy trình xoắn ốc



# Nội dung

## Chương 2: Các khái niệm cơ bản

- Lớp đối tượng và đối tượng
  - Thuộc tính và phương thức
  - Visibility: Public, protected, private
  - Tính kế thừa và đa hình
- Quan hệ giữa các lớp đối tượng
  - Quan hệ tổng quát hóa (Generalization)
  - Quan hệ Association, Aggregation, Composition
  - Quan hệ Dependency
- Lược đồ UML
  - Lịch sử
  - Các sơ đồ

# Nội dung

## Chương 3: Yêu cầu của người sử dụng

- ❖ Mục tiêu giai đoạn khảo sát hiện trạng và xác định yêu cầu
  - Hiện trạng tổ chức
  - Hiện trạng nghiệp vụ
  - Hiện trạng Tin học (phần cứng, phần mềm, con người)
- ❖ Xác định và thu thập yêu cầu
  - Phân loại yêu cầu
    - Yêu cầu chức năng: Lưu trữ, tra cứu, tính toán, kết xuất
    - Yêu cầu phi chức năng
  - Kỹ thuật thu thập yêu cầu
    - Phỏng vấn
    - Bảng câu hỏi
    - Nghiên cứu các tài liệu
    - Quan sát thực tế
    - Phân tích thiết kế nhóm (JAD)

## Chương 4: Mô hình hóa yêu cầu

- ❖ Mô hình hóa yêu cầu
  - Lược đồ Use-case
  - Khái niệm Actor và Usecase
- ❖ Mô hình hóa các dòng dữ liệu của mỗi Use-case
- ❖ Activity Diagram
- ❖ Sequence Diagram

# Nội dung

## Chương 5: Phân tích theo hướng đối tượng

- ❖ Mô hình lớp đối tượng
- ❖ Xác định các lớp đối tượng
- ❖ Mối quan hệ giữa các lớp đối tượng
- ❖ Các thuộc tính của lớp
- ❖ Các phương thức của lớp
- ❖ Sơ đồ trạng thái

# Nội dung

## Chương 6: Thiết kế hướng đối tượng

- ❖ Thiết kế lớp
- ❖ Thiết kế kiến trúc
- ❖ Thiết kế dữ liệu
- ❖ Thiết kế giao diện
- ❖ Các nguyên tắc thiết kế

# Nội dung

## Chương 7: Mẫu thiết kế hướng đối tượng

- ❖ Giới thiệu về mẫu thiết kế
- ❖ Các mẫu thiết kế tạo lập đối tượng
- ❖ Các mẫu thiết kế cấu trúc
- ❖ Các mẫu thiết kế hành vi

# Tài liệu tham khảo

1. Stephen Schach, **Object-Oriented and Classical Software Engineering**. McGraw Hill, 2010.
2. Matt Weisfeld, **The Object-Oriented Thought Process**, Addison-Wesley, 2019
3. Grady Booch, Robert A. Maksimchuk, ... **Object-oriented analysis and design with applications**, Addison-Wesley, 2007.
4. Hassan Gomaa, **Software Modeling and Design: UML, Use Cases, Patterns, and Software Architectures**. Cambridge University Press, 2011.
5. Raul Sidnei Wazlawick, **Object-Oriented Analysis and Design for Information Systems**, Morgan Kaufmann, 2014.
6. Ian Sommerville, **Software Engineering** (10<sup>th</sup> edition), Addison Wesley, 2021.
7. Alexander Shvets, **Dive Into Design Patterns**, 2019.

# Q&A



**UIT**  
TRƯỜNG ĐẠI HỌC  
CÔNG NGHỆ THÔNG TIN

