

HK I, 2012-2013

# Bài 3: Trừu tượng hóa dữ liệu

Giảng viên: Hoàng Thị Diệp

Khoa Công nghệ Thông tin – Đại học Công Nghệ

# Một số vấn đề cần thảo luận (1)

- **Bắt buộc:** Đọc giáo trình để chuẩn bị trước khi lên lớp lý thuyết!
- Cụ thể hóa điều kiện thi cuối kì bằng điểm điểm danh
  - đi học: 1đ (nghỉ có lý do chính đáng và báo trước)
  - nghỉ học không báo trước: 0đ
  - không trả lời được câu hỏi lý thuyết: -1đ
  - trả lời được câu hỏi lý thuyết: +1đ
  - tổng kết điểm danh >12đ thì được thi!
- Đăng bài trên diễn đàn
  - phải liên quan tới môn học
  - chưa có bài đăng tương tự
  - tiêu đề rõ ràng đúng trọng tâm
  - tiếng Việt có dấu, đúng chính tả

## Địa chỉ website bài giảng tập trung

[Options](#)

☆ 3 messages - [Collapse all](#) - [Translate all to English](#) - [Report discussion as spam](#)

**Diep** [View profile](#) [Translate to English](#)

[More options](#) Sep 15, 2:54 am

Chào INT2203 1 và 3,

Mình đã lập website bài giảng ở địa chỉ  
<https://sites.google.com/site/uetint2203/>.

Các file bài giảng và bài tập sẽ được mình cập nhật tập trung tại đó.

Hiện đã có bài giảng 1 (đã chỉnh sửa), bài giảng 2 và bài thực hành 2.  
Các bạn cập nhật nhé!

Have a nice weekend!

[Reply](#) [Reply to author](#) [Forward](#)

## Cả nhà cho mình hỏi bài giảng và bài tập thực hành cô up lên đâu vậy thế ???

[Options](#)

☆ 3 messages - [Collapse all](#) - [Translate all to English](#) - [Report discussion as spam](#)

**Trần Trọng Đông** [View profile](#) [Translate to English](#) [More options](#) Sep 19, 10:43 pm

Mình không đi học buổi đầu tiên nên không biết được bài tập và bài giảng  
cô  
up lên trang nào. bạn nào biết chỉ giúp mình nhé, thank much!!!

[Reply](#) [Reply to author](#) [Forward](#) [Report spam](#)

## bài giảng trên lớp

[Options](#)

☆ 2 messages - [Collapse all](#) - [Translate all to English](#) - [Report discussion as spam](#)

**Nguyễn Thoại IT class CC** [View profile](#) [Translate to English](#) [More options](#) Sep 19, 10:41 pm

Chào mọi người! cho mình hỏi ai có bản mềm bài giảng của cô trên lớp  
không? up lên cho mình với, thanks so much! mỗi lần cô giáo bảo về đọc  
bài  
mà mình không mượn được sách trên thư viện nên không có mà đọc!  
đến lớp  
trong đó ầu chẳng có gì hết trơn.

[Reply](#) [Reply to author](#) [Forward](#) [Report spam](#)

# Một số vấn đề cần thảo luận (2)

- Đi thực hành
  - Tranh thủ đọc trước đề và các tài liệu tham khảo cần thiết ở nhà
  - Lên phòng thực hành để trao đổi thêm và hoàn thiện
  - Nhớ mang laptop nếu có
  - Ngồi desktop thì phải tắt máy trước khi về
- Bài thực hành số 2
  - Phải đánh giá được độ phức tạp của thuật toán đề xuất!
- Địa chỉ web
  - Diễn đàn <https://groups.google.com/group/uetint2203>
  - Download bài giảng và bài thực hành <https://sites.google.com/site/uetint2203/>
- Bầu nhóm trưởng, lớp trưởng

# Mục tiêu bài học

- Biểu diễn dữ liệu trong các ngôn ngữ lập trình
- Sự trừu tượng hóa dữ liệu
- Kiểu dữ liệu trừu tượng
  - Đặc tả
  - Cài đặt

# Biểu diễn như thế nào?

- Tuổi của một người.
- Điểm của một môn học tín chỉ.
- Một phân số. Một dãy phân số.
- Một điểm ảnh (pixel) của ảnh RGB biết cường độ mỗi màu nằm trong  $[0; 255]$ . Một ảnh RGB.
- Một điểm, một đoạn thẳng, một tam giác trong hệ tọa độ 2 chiều.
- Một đa thức bậc  $n$ .
- Giá trị của  $n!$  với  $n$  nhỏ. Giá trị của  $n!$  với  $n$  lớn.

# Dữ liệu

- Dữ liệu là những thông tin mà máy tính có thể xử lý: số nguyên, số thực, xâu kí tự, và các dữ liệu phức tạp được tạo từ nhiều thành phần
- Trong bộ nhớ máy tính, dữ liệu được biểu diễn dưới dạng nhị phân (dãy 0, 1)
- Trong các ngôn ngữ lập trình bậc cao (C++, Java..), dữ liệu được biểu diễn dưới dạng trừu tượng, xuất phát từ biểu diễn toán học và dễ hiểu cho con người:
  - `int`        `age`
  - `double`   `weight`

# Kiểu dữ liệu cơ bản

Kiểu dữ liệu được xác định bởi:

1. Phạm vi giá trị
2. Các phép toán

Ví dụ trong C++

kiểu	phạm vi	phép toán thường dùng
➤ bool	true / false	and, or, not
➤ char	-128 -> 127	'<', '>', '=='
➤ short int	-32,768 -> 32,767	'<', '>', '==', '+', '-', '*', '/'
➤ float	+/- 3.4e +/- 38	'<', '>', '==', '+', '-', '*', '/'
➤ double	+/- 1.7e +/- 308	'<', '>', '==', '+', '-', '*', '/'



# Kiểu dữ liệu có cấu trúc

Ngôn ngữ lập trình cung cấp cho ta những luật để xây dựng kiểu dữ liệu mới  $T$  từ những kiểu dữ liệu đã biết  $t_1, t_2, \dots, t_n$ .

Ví dụ trong C++:

```
struct T {  
     $t_1$   $x_1$   
     $t_2$   $x_2$   
    .....  
     $t_n$   $x_n$   
}
```

# Kiểu dữ liệu có cấu trúc

- Xây dựng cấu trúc dữ liệu để biểu diễn dữ liệu của 1 điểm trên mặt phẳng

```
struct PointType {  
    double    x;  
    double    y;  
}
```

- Xây dựng cấu trúc dữ liệu để biểu diễn dữ liệu của 1 đoạn thẳng trên mặt phẳng

```
struct LineType {  
    Point Type    start;  
    PointType     end;  
}
```

# Phạm vi và các phép toán trên kiểu dữ liệu có cấu trúc

Xét kiểu dữ liệu mới  $T$  được tạo từ những kiểu dữ liệu đã biết  $t_1, t_2, \dots, t_n$ ,

Ví dụ:

```
struct ComplexType {  
    double real;  
    double image;  
}
```

**Phạm vi: Xác định bởi phạm vi của các kiểu dữ liệu thành phần**

- real: là số thực nằm trong phạm vi kiểu 'double'
- image: là số thực nằm trong phạm vi kiểu 'double'

# Phạm vi và các phép toán trên kiểu dữ liệu có cấu trúc

## Phép toán: Do người dùng định nghĩa

Ví dụ:

```
struct ComplexType {  
    double real;  
    double image;  
}
```

```
ComplexType createComplex (double real, double image) {  
    ComplexType c;  
    c.real = real;  
    c.image = image;  
    return c;  
}
```

# Phạm vi và các phép toán trên kiểu dữ liệu có cấu trúc

```
ComplexType add (ComplexType c1, ComplexType c2) {  
    ComplexType c12;  
    c12.real = c1.real + c2.real;  
    c12.image = c1.image + c2.image;  
    return c12;  
}
```

```
ComplexType multiply (ComplexType c1, ComplexType c2) {  
    ComplexType c12;  
    c12.real = (c1.real * c2.real) – (c1.image * c2.image);  
    c12.image = (c1.real * c2.image) + (c1.image * c2.real);  
    return c12;  
}
```

# Trừu tượng hóa dữ liệu

(abstraction)

1. Đặc tả đối tượng dữ liệu (các thành phần dữ liệu của đối tượng)  
Ví dụ: đối tượng số phức (Complex)
  - real
  - image
2. Đặc tả các phép toán trên đối tượng dữ liệu (operations)  
Ví dụ: đối tượng số phức (Complex)
  - createComplex (real, image)
  - getReal (complexNumber)
  - getImage (complexNumber)
  - add (complexNumber1, complexNumber2)
  - multiply (complexNumber2, complexNumber2)
  - print (complexNumber)

# Trừu tượng hóa dữ liệu

## Trừu tượng hóa đối tượng sinh viên (Student)

1. Đặc tả đối tượng dữ liệu  
name, age, sex, address
2. Đặc tả các phép toán trên đối tượng dữ liệu  
createStudent (name, age, sex, address)  
compare (student1, student2)  
getName (student)  
getAge (student)  
getSex (student)  
getAdd (student)

# Trừu tượng hóa dữ liệu

- StudentClass
  1. Đặc tả đối tượng dữ liệu  
className, numberStudent, studentArr, address
  2. Đặc tả các phép toán trên đối tượng dữ liệu  
addStudent (studentClass, student)  
findStudent (studentClass, student)  
deleteStudent (studentClass, student)  
getClassName (studentClass)  
getNumberStudent (studentClass)  
getStudentArr (studentClass)  
getClassAddress (studentClass)



# Giải một bài toán tin học

- Đặc tả vấn đề
- Thiết kế cấu trúc dữ liệu
- Thiết kế giải thuật
- Cài đặt (C++, Java...)
- Thử nghiệm và sửa lỗi
- Tối ưu chương trình

# Ví dụ

- Bài toán: Giả sử chúng ta cần viết chương trình lập lịch thi. Vấn đề như sau. Mỗi người dự thi đăng kí thi một số môn trong số các môn tổ chức thi. Chúng ta cần xếp lịch thi, mỗi ngày thi một số môn trong cùng một thời gian, sao cho mỗi người dự thi có thể thi tất cả các môn họ đã đăng kí.
- Đặc tả bằng danh sách?
- Đặc tả bằng đồ thị?

# Lập trình hướng đối tượng

## Object oriented programming (OOP)

- Lập trình hướng đối tượng giúp chúng ta cài đặt các mô tả trừu tượng (đối tượng dữ liệu và các phép toán) thành các đoạn mã chương trình
- Chương trình được thiết kế thành từng đoạn nhỏ, mỗi đoạn mô tả về một đối tượng (thuộc tính dữ liệu, các phép toán trên dữ liệu)
- Hai thuộc tính quan trọng: đóng gói (encapsulation) và thừa kế (inheritance)

# OOP: Tính đóng gói (encapsulation)

- **Class:** Cài đặt một lớp đối tượng dữ liệu trừu tượng. Việc cài đặt bao gồm cài đặt các thành phần dữ liệu và các phép toán trên dữ liệu

Ví dụ:

```
class Complex {  
    double real;  
    double image;  
public:  
    void create (double newReal, double newImage) {  
        real = newReal; image = newImage;  
    }  
    double getReal () {  
        return real;  
    }  
    .....  
    void print {  
        cout << real << " +i " << image << " \n" ;  
    }  
};
```

- Liên kết chặt chẽ giữa dữ liệu và phép toán
- Che giấu dữ liệu
- Dễ dàng tìm lỗi
- Các đối tượng liên kết với nhau thông qua các phép toán

# OOP: Tính đóng gói (encapsulation)

**Object:** Biểu diễn cho một đối tượng cụ thể của một lớp

Complex     c1;

Complex     c2;

# C++

- Lập trình tổng quát (Generic programming)
- Con trỏ và cấp phát động
- Lớp