Chương 02:

Lập trình hướng đối tượng

1. *Class & Object*
2. *Property & Method*
3. *Overloading method*
4. *Constructor*
5. *Inheritance*
6. *Overriding method*
7. *Access Modifier*
8. *Static variable & static method*
9. *Xây dựng class Fraction*
10. *Tình huống thực hành có đáp án*

Chương 02:

Lập trình hướng đối tượng

1. ***Class & Object***
2. *Property & Method*
3. *Overloading method*
4. *Constructor*
5. *Inheritance*
6. *Overriding method*
7. *Access Modifier*
8. *Static variable & static method*
9. *Xây dựng class Fraction*
10. *Tình huống thực hành có đáp án*

Chương 02:

Lập trình hướng đối tượng

Phần 1 – Class & Object

Lập trình hướng thủ tục & hướng đối tượng

Lập trình hướng thủ tục

- Giải quyết vấn đề từng bước đến khi đạt yêu cầu
- Lập trình từ trên xuống
- Lập trình theo hàm → chỉ tạo ra hàm xử lý khi gặp vấn đề nào đó

Lập trình hướng đối tượng

- Dựa trên nền tảng các lớp đã xây dựng sẵn
- Xác định trước các chức năng cần phải thực hiện

Lập trình hướng đối tượng

- OOP – Object oriented programming là kiểu lập trình lấy đối tượng làm nền tảng
- Đơn giản hóa việc phát triển chương trình
- Tạo ra các chương trình có tình mềm dẻo và linh động cao
- Dễ dàng phát triển, bảo trì và nâng cấp.

Phân biệt Class & Object

Class chỉ một cái gì đó chung chung, object là một cái cụ thể

- Công thức làm bánh quy là một Class → bánh quy là một Object
- Con gái là một Class → Hồng là một Object
- Con mèo là một Class → Con mèo Mimi nhà tôi là một Object
- Bản vẽ là một Class → Ngôi nhà của tôi là một Object

Khai báo Class

Tạo class Student nằm trong Main.java

```
package chap02.oop;
public class Main {
    public static void main(String[] args) {
        System.out.println("Main.main");
    }
}
class Student {
}
```

→ Thêm từ khóa public vào trước class Student ?

Khởi tạo đối tượng của Class

Tạo class Student nằm trong Main.java

```
package chap02.oop;
public class Main {
    public static void main(String[] args) {
        Student studentOne      = new Student();
        Student studentTwo      = new Student();
    }
}
```

→ Tách class Student thành 1 file riêng (cùng package)

→ Tách class Student thành 1 file riêng (khác package)

Chương 02:

Lập trình hướng đối tượng

1. *Class & Object*
2. ***Property & Method***
3. *Overloading method*
4. *Constructor*
5. *Inheritance*
6. *Overriding method*
7. *Access Modifier*
8. *Static variable & static method*
9. *Xây dựng class Fraction*
10. *Tình huống thực hành có đáp án*

Chương 02:

Lập trình hướng đối tượng

Phần 2 – Property & Method

Hiểu thế nào về Property & Method

Property (Thuộc tính)

- Là các đặc điểm, đặc tính của một lớp

Method (Phương thức)

- Là các hành động có thể được thực hiện từ lớp
- Phương thức cũng giống như hàm, nhưng là hàm riêng của lớp

Ví dụ về phương thức và thuộc tính

Xét Object "Sinh viên Yến Trang"

- Đặc điểm: tóc dài, má lúng đồng tiền, cao 1.7 m, ... → thuộc tính
- Hành động: ăn, ngủ, dạy anh văn, đánh đàn, ... → phương thức

Xét Object "Con cún Ken nhà tôi"

- Đặc điểm: lông xoăn, màu xám, đuôi ngắn, ... → thuộc tính
- Hành động: ăn, ngủ, bắt chuột, ... → phương thức

Khai báo property

```
public class Student {
        public String name;
        public String code;
        public int birthday;
}
```

Access modifiers (phạm vi truy xuất)

Data type (kiểu dữ liệu)

Property (tên của thuộc tính)

→ Truy cập và gán giá trị cho các thuộc tính

Truy cập và gán giá trị cho các thuộc tính

```
public static void main(String[] args) {
    Student studentOne     = new Student();
    studentOne.birthday  = 1994;
    studentOne.name      = "John";
    studentOne.code      = "S001";

    System.out.println("birthday: " + studentOne.birthday);
    System.out.println("name: "     + studentOne.name);
    System.out.println("code: "     + studentOne.code);
}
```

Phương thức getter và setter

```
public static void main(String[] args) {
    Student studentOne     = new Student();
    studentOne.birthday  = 1994;
    studentOne.name      = "John";
    studentOne.code      = "S001";

    System.out.println("birthday: " + studentOne.birthday);
    System.out.println("name: "     + studentOne.name);
    System.out.println("code: "     + studentOne.code);
}
```

setter

getter

Khai báo method

```
public class People {
    public String name;

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
}
```

```
[access] [data] [method] ([param], [...]) {

        // your code here

}
```

Chương 02:

Lập trình hướng đối tượng

1. *Class & Object*
2. *Property & Method*
3. ***Overloading method***
4. *Constructor*
5. *Inheritance*
6. *Overriding method*
7. *Access Modifier*
8. *Static variable & static method*
9. *Xây dựng class Fraction*
10. *Tình huống thực hành có đáp án*

Chương 02:

Lập trình hướng đối tượng

Phần 3 – Overloading Method

Overloading method (nạp chồng phương thức)

Overloading method là những phương thức nằm trong cùng một class có cùng tên nhưng khác các tham số (khác kiểu dữ liệu, khác số lượng tham số)

```
public void setCode(String code) {}
public void setCode() {}
public void setCode(String first, int last) {}
public void setCode(int code) {}
```

Chương 02:

Lập trình hướng đối tượng

1. *Class & Object*
2. *Property & Method*
3. *Overloading method*
4. ***Constructor***
5. *Inheritance*
6. *Overriding method*
7. *Access Modifier*
8. *Static variable & static method*
9. *Xây dựng class Fraction*
10. *Tình huống thực hành có đáp án*

Chương 02:

Lập trình hướng đối tượng

Phần 4 – Constructor

Constructor (phương thức khởi tạo)

- Contructor **được gọi tự động** và được gọi đầu tiên khi một object được khởi tạo
- Contructor không có giá trị trả về, có thể có tham số hoặc không có
- Constructor phải có **cùng tên với lớp**
- Constructor **có thể bị nạp chồng** (overloading)
- Nếu một class chưa khai báo constructor thì sẽ được JAVA cung cấp một constructor mặc định (default constructor).

Constructor (phương thức khởi tạo)

```
public class Student{
    public String name;

    public Student(){
        System.out.println("Constructor 1");
        this.setName("John");
    }

    public Student(String name){
        System.out.println("Constructor 2");
        this.setName(name);
    }
}
```

Chương 02:

Lập trình hướng đối tượng

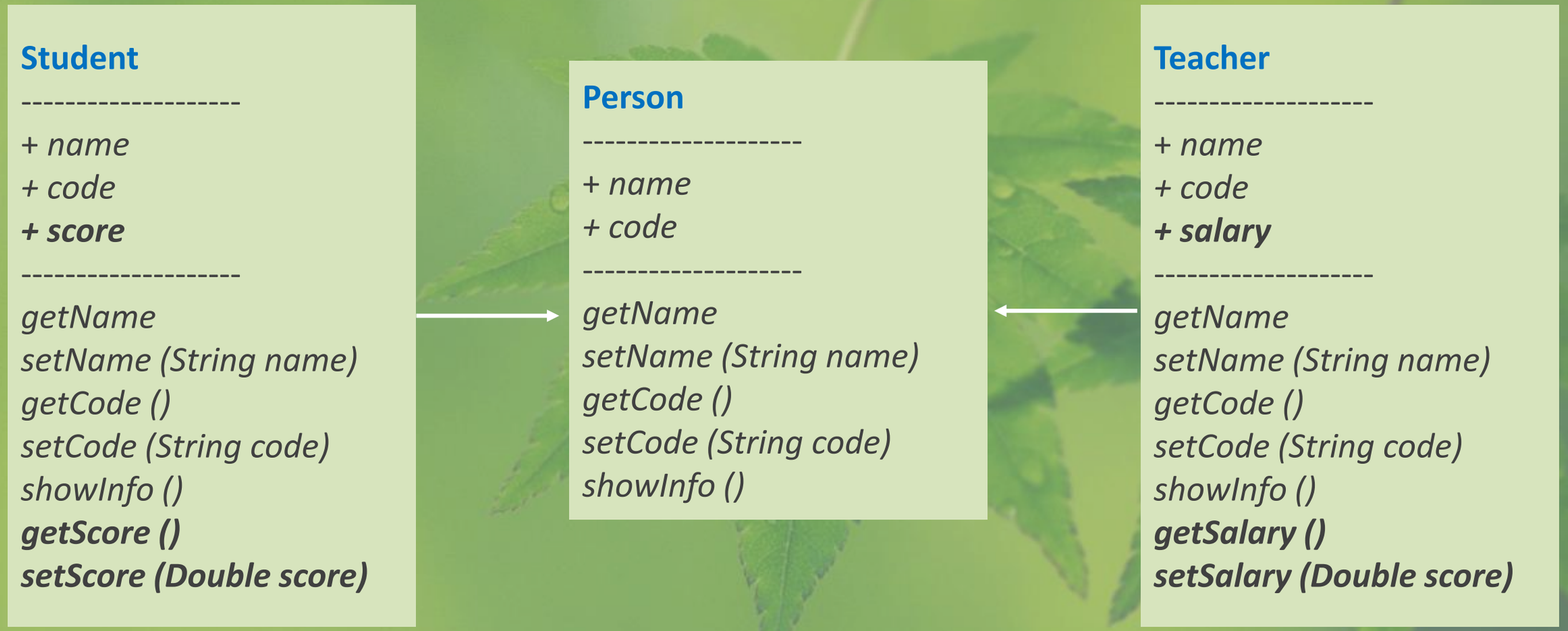
1. *Class & Object*
2. *Property & Method*
3. *Overloading method*
4. *Constructor*
5. ***Inheritance***
6. *Overriding method*
7. *Access Modifier*
8. *Static variable & static method*
9. *Xây dựng class Fraction*
10. *Tình huống thực hành có đáp án*

Chương 02:

Lập trình hướng đối tượng

Phần 5 – Inheritance

Tình huống

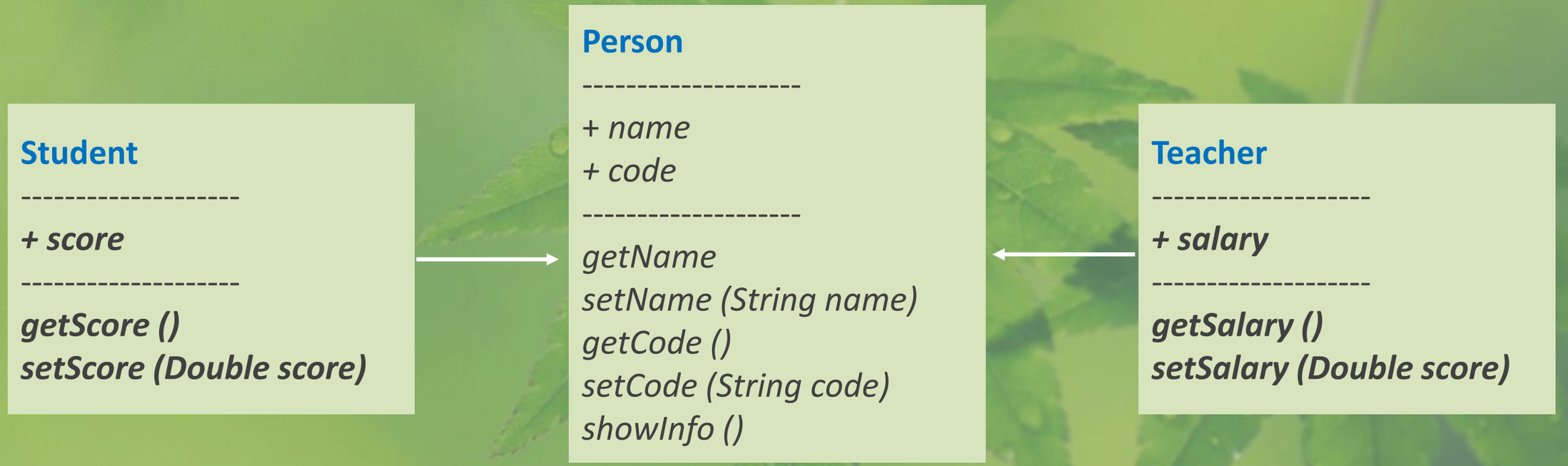

Student

+ name
+ code
+ score
---
getName
setName (String name)
getCode ()
setCode (String code)
showInfo ()
getScore ()
setScore (Double score)
Person

+ name
+ code
---
getName
setName (String name)
getCode ()
setCode (String code)
showInfo ()
Teacher

+ name
+ code
+ salary
---
getName
setName (String name)
getCode ()
setCode (String code)
showInfo ()
getSalary ()
setSalary (Double score)

Tình huống


Person

+ name
+ code
---
getName
setName (String name)
getCode ()
setCode (String code)
showInfo ()
Student

+ score
---
getScore ()
setScore (Double score)
Teacher

+ salary
---
getSalary ()
setSalary (Double score)

Inheritance (sự kế thừa)

Student kế thừa từ Person được biểu diễn qua từ khóa extends

```
class Student extends Person{

}
```

- Student gọi là class con (subclass), Person gọi là class cha (superclass)
- Student nếu đã kế thừa từ Person thì không thể kế thừa thêm lớp khác

Chống kế thừa

```
final public class Person{

}
```

- Từ khóa final cho biết Person là lớp hằng → không lớp nào có thể kế thừa từ lớp Person được nữa

Chương 02:

Lập trình hướng đối tượng

1. *Class & Object*
2. *Property & Method*
3. *Overloading method*
4. *Constructor*
5. *Inheritance*
6. ***Overriding method***
7. *Access Modifier*
8. *Static variable & static method*
9. *Xây dựng class Fraction*
10. *Tình huống thực hành có đáp án*

Chương 02:

Lập trình hướng đối tượng

Phần 6 – Overriding

Overriding (sự ghi đè)

- Phương thức đã xuất hiện ở lớp cha và xuất hiện tiếp ở lớp con.
- Khi đối tượng thuộc lớp con gọi phương thức thì sẽ chọn lựa và chạy theo phương thức trong lớp con.
- Nếu lớp con không có phương thức đó thì mới gọi phương thức đó ở lớp cha

Overloading vs Overriding

Overloading (nạp chồng)	Overriding (ghi đè)
• Xuất hiện: trong cùng lớp • Tên phương thức: giống nhau • Số lượng tham số và kiểu dữ liệu của tham số: có thể khác nhau • Đa hình (polymorphism) trong quá trình biên dịch	• Xuất hiện: ở class cha và class con • Tên phương thức: giống nhau • Số lượng tham số và kiểu dữ liệu của tham số: giống nhau • Đa hình (polymorphism) trong quá trình thực thi

Chương 02:

Lập trình hướng đối tượng

1. *Class & Object*
2. *Property & Method*
3. *Overloading method*
4. *Constructor*
5. *Inheritance*
6. *Overriding method*
7. ***Access Modifier***
8. *Static variable & static method*
9. *Xây dựng class Fraction*
10. *Tình huống thực hành có đáp án*

Chương 02:

Lập trình hướng đối tượng

Phần 7 – Access modifier

Access modifier

Access modifier cấp độ truy cập cho class, *property* và method

public class Student {}

- **Private** chỉ truy cập được trong class *(property, method)*
- **Null** (rỗng) truy cập trong package *(class, property, method)*
- **Protected** truy cập trong package và các subclasses *(property, method)*
- **Public** truy cập từ bất kỳ đâu *(class, property, method)*

Chương 02:

Lập trình hướng đối tượng

1. *Class & Object*
2. *Property & Method*
3. *Overloading method*
4. *Constructor*
5. *Inheritance*
6. *Overriding method*
7. *Access Modifier*
8. ***Static variable & static method***
9. *Xây dựng class Fraction*
10. *Tình huống thực hành có đáp án*

Chương 02:

Lập trình hướng đối tượng

Phần 8 – Static variable & Static method

Static keyword

- **Static keyword** được sử dụng chủ yếu trong việc quản lý bộ nhớ
- Static có thể được áp dụng cho:
  - *variable (class variable)*
  - *method (class method)*
  - block
  - nested class

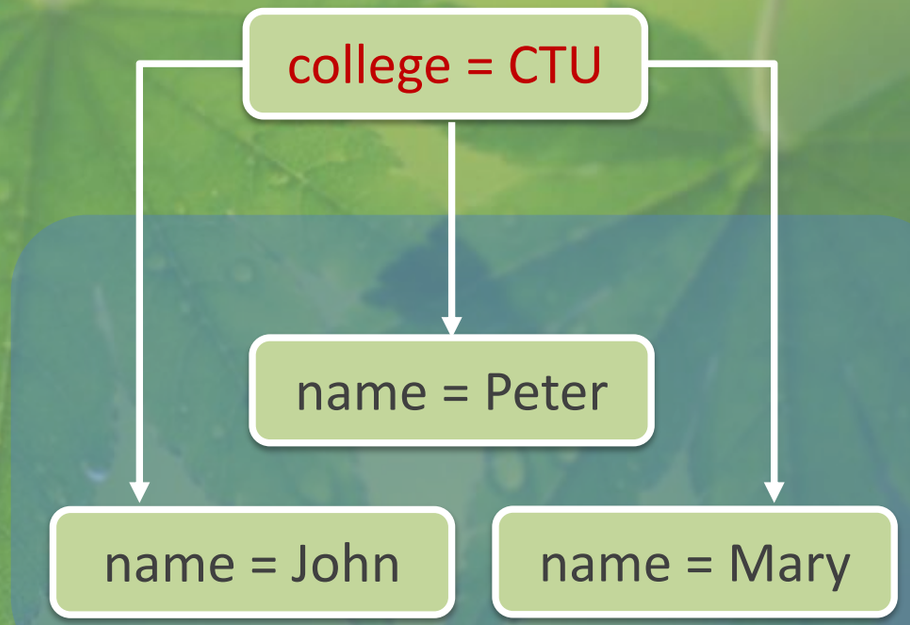
Static variable

```
public class Student{
   private String name;
   private String college = "CTU";
}
```

name = Peter
college = CTU

name = John
college = CTU

name = Mary
college = CTU



Static method

- Static method thuộc về lớp không thuộc về đối tượng
- Static method được gọi mà không cần tạo một thể hiện của lớp.
- Static method có thể truy cập và thay đổi giá tị của static variable
- Các property và method không thỏa static thì static method không có quyền truy cập vào

Chương 02:

Lập trình hướng đối tượng

1. *Class & Object*
2. *Property & Method*
3. *Overloading method*
4. *Constructor*
5. *Inheritance*
6. *Overriding method*
7. *Access Modifier*
8. *Static variable & static method*
9. ***Xây dựng class Fraction***
10. *Tình huống thực hành có đáp án*

Chương 02:

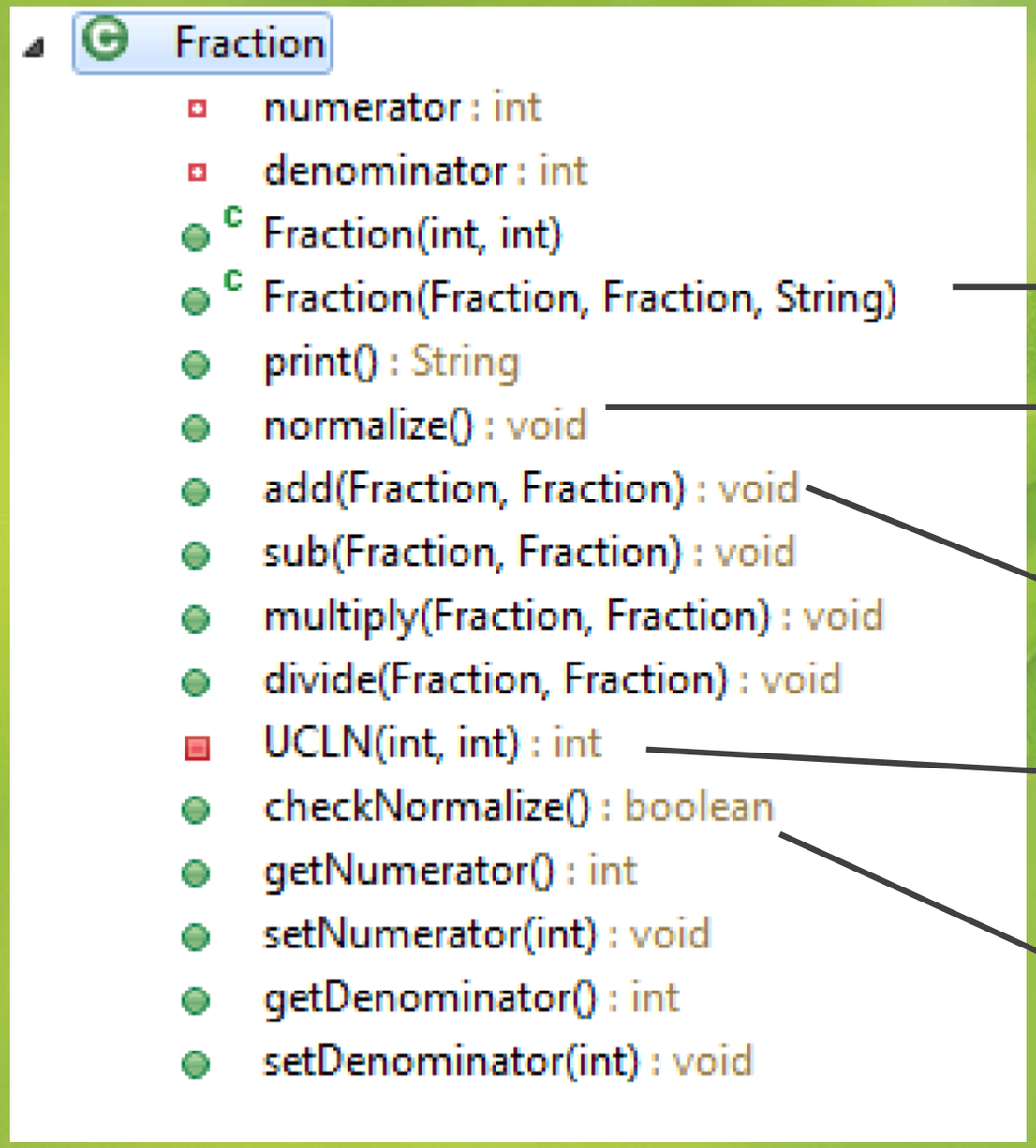
Lập trình hướng đối tượng

Phần 9 – Xây dựng class Fraction

Class Fraction

- Nhập phân số (tử số và mẫu số)
- In phân số
- Kiểm tra phân số tối giản
- Tối giản phân số
- Thực hiện phép cộng 2 phân số
- Thực hiện phép trừ 2 phân số
- Thực hiện phép nhân 2 phân số
- Thực hiện phép chia 2 phân số

Class Fraction



Thao tác trên hai phân số

Tối giản phân số

...

Cộng hai phân số

Tìm UCLN của hai số

...

Kiểm tra phân số tối giản

Chương 02:

Lập trình hướng đối tượng

1. *Class & Object*
2. *Property & Method*
3. *Overloading method*
4. *Constructor*
5. *Inheritance*
6. *Overriding method*
7. *Access Modifier*
8. *Static variable & static method*
9. *Xây dựng class Fraction*
10. ***Tình huống thực hành có đáp án***

Chương 02:

Lập trình hướng đối tượng

Phần 10 – Tình huống thực hành

Xây dựng class Book

- Nhập thông tin một quyển sách (id, tên và giá tiền của quyển sách đó)
- In thông tin quyển sách
- Cập nhật thông tin quyển sách

```
====================== BOOK MANAGER ======================
1. Add book
2. Edit book
3. Info book
4. Exit
Your choise [1-4]:
```