Danh sách kiểm tra giữa kỳ ở nhóm 1: (9h20 đến 10h20)

<https://docs.google.com/spreadsheets/d/1de8jeqaPFp35TpNXjojlAzxdwQkmUNXsx1ZUkEM9j2Q/edit?usp=sharing>

(Nhóm các bạn tô màu xanh lá cây)

Phần còn lại sẽ kiểm tra ở nhóm 2 (10h35 đến 11h35)

Khi thi không sử dụng tài liệu, sẽ được phát đề và giấy thi

Không mang theo điện thoại, cặp sẽ để ở ngay dưới bảng của lớp

=================================11/6=================================

Một công trình (**Build**) có các thông số về diện tích (**sqr** kiểu **double**), số định danh (**id** kiểu **String**) và địa chỉ (**adr** kiểu **String**) và số người sống trong công trình (**peo** kiểu **int**). Một công trình xây dựng có thể là:

(i) một căn phòng khép kín (**Room**) với thông số về số cửa sổ (**wid** kiểu **int**)

(i) một căn hộ cụ thể (**Home**) với thông số về số tầng (**fls** kiểu **int**).

(iii) Một chung cư (**Cond**) là tập hợp nhiều **Room** bên trong. Chung cư có thông số về số tầng (**fls**) và số lượng **Room** trong một tầng.

(iv) Khoảng không gian trống (**Park**) có thêm thông số về tỉ lệ bao phủ cây xanh bên trong diện tích đó (**gre** kiểu **double**).

Hãy xây dựng biểu đồ lớp

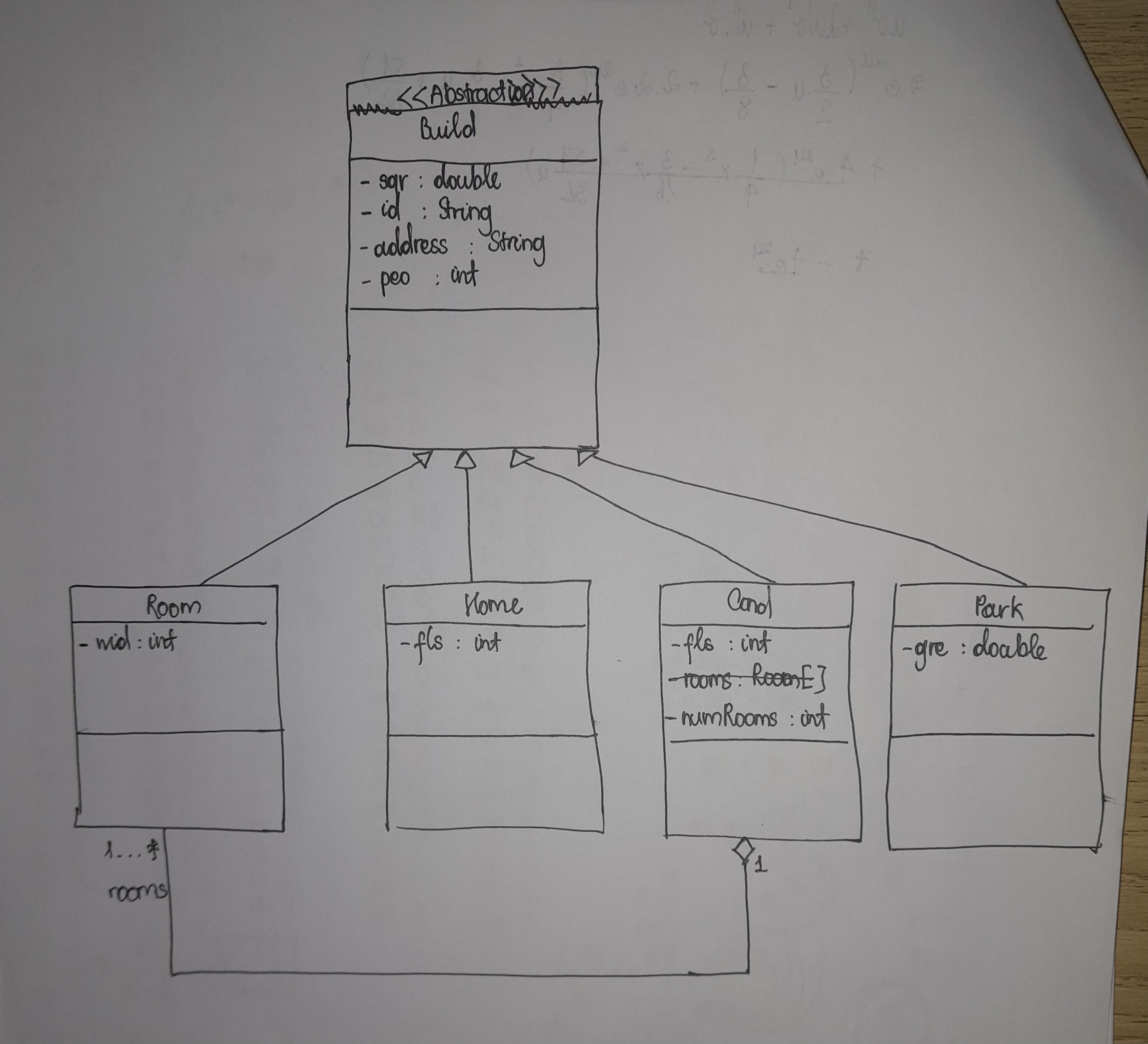
Bước 1: xác định các đối tượng cần được mô tả trong biểu đồ lớp: công trình (Build), ~~người~~ (?), căn phòng khép kín (Room), căn hộ tập thể (Home), ~~tầng~~ (?), chung cư (Cond), Không gian trống (Park), ~~cây xanh~~ (?)

Bước 2: Build có thuộc tính (sqr, id, adr, peo). Room có thêm thuộc tính wid, Home có thông số về fls, Cond có thông số về các rooms, fls và numRooms. Park có thuộc tính gre

Bước 3: Có Room kết tập bên trong Cond (bội số quan hệ 1..\*)

Bước 4: Xác định kế thừa/cài đặt - có cần tạo ra lớp mới hoặc giao diện mới không? => không cần. Nhưng có lớp trừu tượng là lớp? Build

Nếu lớp A kết tập bên trong lớp B dưới dạng B có thuộc tính là mảng các đối tượng của A có tên rooms thì trong biểu đồ lớp ta vẽ đường kết tập từ A đến B, tên vai trò của A là rooms



Người ta muốn xây dựng một phần mềm quản lý giao thông đô thị. Một làn đường (**Lane**) có các thông số về **id** (kiểu String) điểm nguồn (**fr** kiểu String) và điểm đến (**to** kiểu String). Ngoài ra còn có thông số về việc cho phép có lối đi của người đi bộ hay không (**allow** kiểu int với giá trị 0 ứng với việc không cho người đi bộ mà chỉ cho xe di chuyển, 1 ứng với việc chỉ dành cho người đi bộ và 2 ứng với việc cả người đi bộ và xe đều có thể di chuyển).

Một con đường (**Road**) có thông số về **id** (kiểu String) tối đa hai làn đường (**L1** và **L2** kiểu **Lane**), và tối thiểu một làn đường, nếu con đường chỉ có một làn thì một trong hai thuộc tính kia bằng null

Một vạch vượt đường (**Cross**) nối hai làn đường (**L1** và **L2** kiểu **Lane**) với nhau, hai làn này có thể cùng thuộc một con đường hoặc không. Khác với Road, Cross bắt buộc hai thuộc tính L1 và L2 đều khác null.

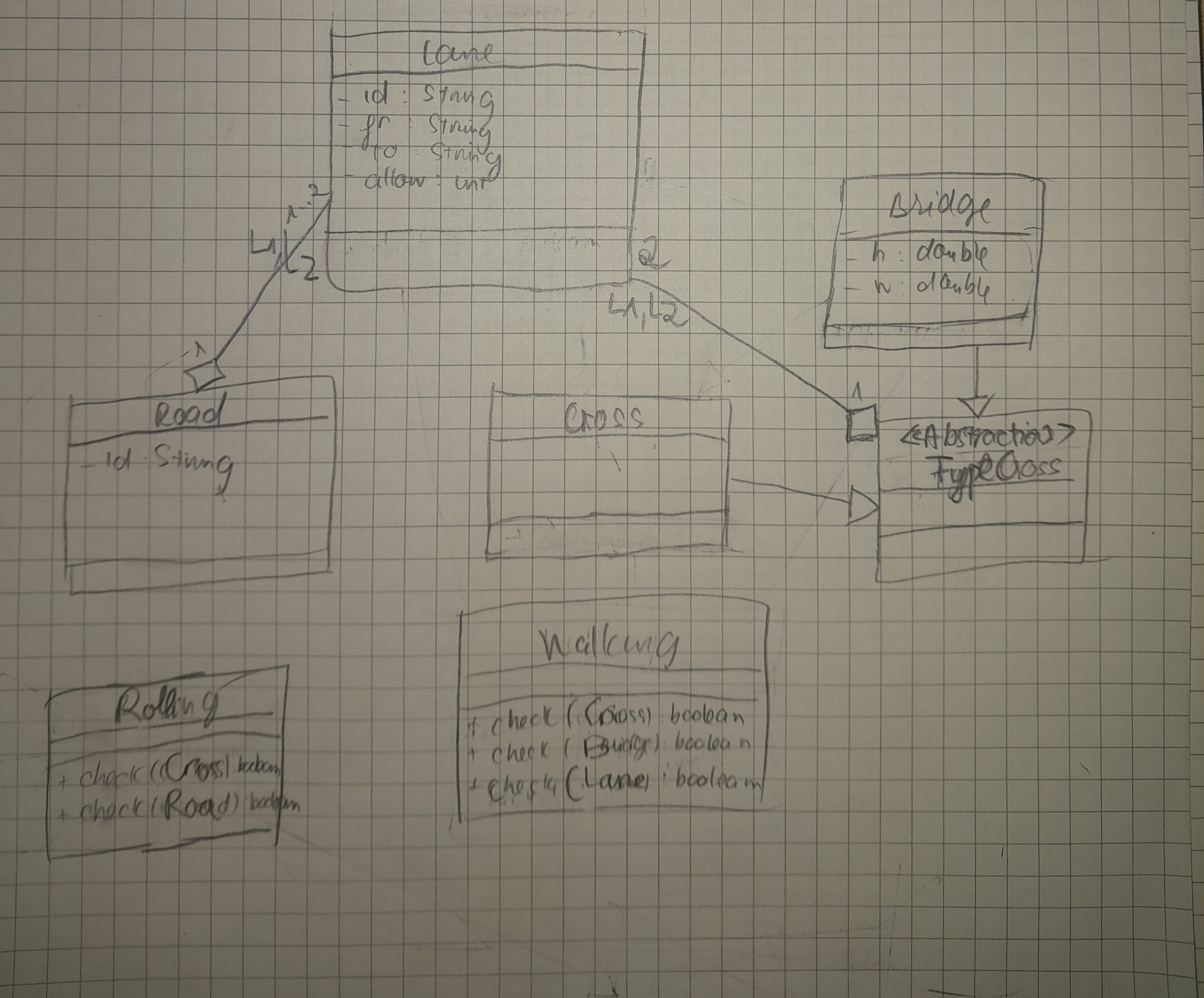
Một loại vạch vượt đường đặc biệt, cũng nối hai làn đường với nhau nhưng không tiếp đất là cầu vượt (Bridge). Ở loại này, cần có tham số về chiều cao (h kiểu double) và trọng tải (w kiểu double) của cầu.

Để đảm bảo lối đi cho khách bộ hành, một thành phần con của hệ thống (mang tên Walking) có nhiệm vụ theo dõi chất lượng của các làn có người đi bộ hoặc các vạch vượt đường. Thành phần này có phương thức public boolean check với một tham số là đối tượng kiểu Cross hoặc Bridge hoặc Lane.

Để đảm bảo lối đi cho xe, một thành phần con khác của hệ thống (mang tên Rolling) có nhiệm vụ theo dõi chất lượng của các đường có xe đi qua. Thành phần này có phương thức public boolean check với một tham số là đối tượng kiểu Cross (nhưng không nhận kiểu Bridge làm tham số) hoặc Road.

Có bạn thiết kế là Bridge kế thừa Cross, lớp Rolling có phương thức check(Cross) và check(Road) nhưng cách thiết kế này sai. Vì khi đó ta vẫn có thể truyền đối tượng của lớp Bridge làm tham số cho check(Cross) của Rolling được (Bridge kế thừa Cross). Do vậy bài này ko thể để Bridge kế thừa Cross được

Nếu lớp A là tham số của một phương thức trong lớp B hoặc trong phương thức của lớp B có khởi tạo một (một số) đối tượng của A thì ta nói quan hệ của A với B là dependency. Nhưng quan hệ này không cần phải vẽ trong biểu đồ lớp mà ta sẽ vẽ trong Collaboration Diagram (hay còn gọi là Communication Diagram)



=================================04/6=================================

Chương 10 đã xong

Lớp ta còn học 2 buổi nữa theo lịch

ngày 11/6 sẽ vẫn học, học chương 12, biểu đồ lớp

ngày 18/6 sẽ thi giữa kỳ, chia làm 2 nhóm. Mỗi nhóm làm bài trong 60ph, bài thi trắc nghiệm và có tự luận.

Sẽ có buổi học bù offline vào chủ nhật để lớp ta tìm hiểu chương 11. Buổi học bù này cho phép các bạn tham gia online

* Danh sách thi giữa kỳ sẽ gửi sau, ưu tiên những bạn nghỉ nhiều sẽ được thi trước.

Người ta thường viết ngoại lệ ở ngay hàm xảy ra lỗi khi mà số lượng hàm còn chưa nhiều (cỡ vài trăm hàm). Nếu số lượng hàm quá nhiều (cỡ vài ngàn) thì người ta sẽ dùng uỷ nhiệm ngoại lệ. Uỷ nhiệm ngoại lệ tức là trong vài ngàn hàm đó có những hàm Z không hề gọi hàm khác mà được hàm X nào đó gọi. Thì người ta ko cần xử lý ngoại lệ trong hàm Z mà đẩy lên (uỷ nhiệm) cho hàm X xử lý ngoại lệ.

Trên thực tế, nếu phát triển phần mềm chuyên nghiệp thì người ta sẽ vẫn dùng song song cả xử lý ngoại lệ tại chỗ lẫn uỷ nhiệm ngoại lệ.

Ưu điểm của việc viết try lồng trong try là lỗi xảy ra ở khối try bên trong sẽ vẫn khiến các chỉ thị còn lại của khối try ngoài được thực thi.

| try{  lệnh A  try{  lệnh B  }  catch(Exception1 e){ }  lệnh C  }  catch(Exception2 e){ } | try{  lệnh A  lệnh B  lệnh C  }  catch(Exception1 e1){ }  catch(Exception2 e2){ } |
| --- | --- |

Ở cách bên trái, nếu có ngoại lệ ở B thì khối C vẫn được thực thi. Ngược lại ở cách bên phải, nếu B có xảy ra ngoại lệ thì khối C sẽ không được thực thi

SLide trang 34 chỉ nêu ra một ví dụ về việc dùng try catch trong đóng mở file. Trên thực tế, với phiên bản Java mới thì ta không dùng try/catch mở file nữa mà dùng khối using.

Bài trang 60 của chương 10: <https://onlinegdb.com/LUf_aia5s>

=================================28/5=================================

C9D232: Theo em loại thao tác nào có thể làm được trên lập trình cấu trúc, cụ thể là ngôn ngữ C

A. Khai báo lớp con kế thừa lớp cha

B. Nạp chồng hàm

C. Lập trình tổng quát

D. Đa hình

E. Tất cả các thao tác đều làm được

F. Tất cả các thao tác đều không làm được

ANC: C

C9K232: Theo em đáp án nào khác với các đáp án còn lại

A. Khai báo lớp con kế thừa lớp cha

B. Nạp chồng hàm

C. Lập trình tổng quát

D. Đa hình

ANC: C

C9D232: Theo em mã nguồn sau có vấn đề gì không?

| 1:  2:  3:  4:  5:  6:  7:  8:  9:  10:  11:  12:  13:  14:  15:  16:  17: | class Information<T> {  private T value;  public Information(T value) {  this.value = value;  }  public T getValue() {  return value;  }  }  public class Program {  public static void main(String [ ]args) {  Information<String> mystring = new Information<String>("hello");  Information<Circle> circle = new Information<Circle>(new Circle());  Information<2DShape> shape = new Information<>(new 2DShape());  }  } |
| --- | --- |

A. Không vấn đề gì cả, mã nguồn vẫn thực thi được

B. Dấu [ ] phải đứng sau args ở dòng 12

C. Không có kiểu dữ liệu nào là T trong Java cả

D. Dòng 15 sai do viết new Information< >

E. Khai báo tên lớp String là sai

F. Khai báo tên lớp Circle là sai

G. Khai báo tên lớp 2DShape là sai

ANC: G

C9D232: Trong các lớp sau, lớp nào không thể dùng Iterator để duyệt các phần tử được?

A. ArrayList

B. TreeMap

C. Vector

D. LinkedList

E. Tất cả đều không dùng được

F. Tất cả đều dùng được

ANC: F

C9D232: Ở Java, trong các lớp sau, lớp nào có thể dùng chỉ số thay vì Iterator để duyệt các phần tử được?

A. SortedSet

B. TreeMap

C. Set

D. LinkedList

E. Không có đáp án nào đúng

F. Tất cả các lớp trên đều đúng

ANC: D

=================================21/5=================================

Đã xong chương 8

Việc tự động chuyển đổi kiểu từ đối tượng của lớp con sang đối tượng của lớp cha sẽ giúp:

Có thể khai báo kiểu trả về là lớp cha nhưng trả về đối tượng của lớp con

Có thể truyền tham số là đối tượng của lớp con vào phương thức mà tham số là đối tượng của lớp cha

class A{ }

class B extends A{ }

class C {

public A getA( ) { return new B( ); } //hoàn toàn được khi mà B kế thừa A

public void m(A obj) { }

}

C objC = new C( );

A objB = new B( );

objC.m(new B( )); //Vẫn được

objC.m(objB) ; //Vẫn được

C8D232: Cho đoạn mã nguồn Java gồm các lớp sau:

class A{ public void m( ) { Sysout(“A"); } }

class B extends A{

public void n( ) { Sysout(“B"); }

}

class C extends A{

public void h( ) { Sysout(“C"); }

}

class D {

public void g(A x) {

if(x instanceof B) {

///…A….

}

if(x instanceof C) {

///…B….

}

}

}

Hãy xem các đáp sau, đáp án nào đúng để điền vào chỗ ///…A… và ///…B…

A. Đáp án A điền: x.n( );

Đáp án B điền: C c = (B) x;

B. Đáp án A điền: (B)x.h( );

Đáp án B điền: x.h( );

C. Đáp án A điền: ((B)x).n( );

Đáp án B điền: C c = (A) x;

D. Đáp án A điền: ((B)x).n( );

Đáp án B điền: C c = (C) x;

ANC: D

Đáp án đúng là D.

Đáp án A sai vì thao tác ép kiểu C c = (B) x; là sai. C và B không thể tiến hành ép kiểu cho nhau như vậy vì C và B là 2 kiểu khác nhau cũng như C không phải là lớp cha của B.

Đáp án B sai vì thao tác ép kiểu (B)x thành công thì h( ) cũng ko phải là phương thức của B. Cũng như thao tác x.h( ) là sai vì x chưa được ép kiểu thành C

Đáp án C , chỉ thị ((B)x).n( ) hoàn toàn được vì n( ) có trong B. Nhưng chỉ thị C c = (A)x lại sai

Đáp án D thì tất cả các chỉ thị đều đúng.

Trong Java:

class Program{

public static void foo( ) { }

public static void foo(int x) { }

public static void main(String args) {

foo( );

foo(10);

}

}

Java sẽ liên kết ngay lập tức (liên kế tĩnh) giữa main và foo, foo(10) vì nó biết chắc chắn foo và foo(10) sẽ là hàm nào. Liên kết tĩnh này giúp Java compiler nhúng trực tiếp mã nguồn của foo và foo(10) vào trong hàm main khi biên dịch xong. Liên kết tĩnh sẽ giúp mã nguồn được thực hiện nhanh chóng hơn so với kiểu liên kết còn lại là liên kết động.

nếu tham chiếu đến phương thức tĩnh thì việc tham chiếu sẽ phụ thuộc vào kiểu dữ liệu khai báo của đối tượng

Human obj1 = new Boy( );

//Với walk là PT tĩnh của Human

obj1.walk ( ); //sẽ được gọi đến walk của Human

Boy obj3 = new Boy( );

obj1 = obj3;

obj1.walk( ) ; //sẽ vẫn được gọi đến walk của Human

//Tình trạng này là do phương thức walk được khai báo là static.

Ta hãy xem xét lại mã nguồn của lớp A, B, C, E như bài trước:

class A{ public void m( ) { Sysout(“A"); } }

class B extends A{

public void n( ) { Sysout(“B"); }

}

class C extends A{

public void h( ) { Sysout(“C"); }

}

class D {

public void g(A x) {

if(x instanceof B) {

((B)x).n( );

}

if(x instanceof C) {

C c = (C) x;

c.h( );

}

}

}

public class E{

public static void main(String args[ ]) {

A obj = args.length( ) == 1 ? new A( ) : (args.length( ) == 2 ? new B( ) : null);

obj = (obj == null : new C( ) : obj) ;

D objD = new D( );

objD.g(obj) ;

}

}

Ở tình huống phương thức g(A obj) của D. Trong giai đoạn compile thì Java không biết được sẽ phải liên kết tĩnh đến phương thức n của B hay h của C. Java compiler sẽ đứng trước 3 lựa chọn (a) liên kết tĩnh đến n của B, hay (b) liên kết tĩnh đến h của C hay (c) liên kết tĩnh đến cả 2 phương thức n và h. Đến đây thì Java compiler sẽ giao nhiệm vụ cho Java interpreter. Khi thực thi, đến lúc truyền tham số obj cho phương thức g (tức là đến lúc thực thi lệnh objD.g(obj)) thì Java interpreter mới tham chiếu đến mã nguồn của n hay của h.

Việc liên kết đến mã nguồn của phương thức nào tại thời điểm chạy, được gọi là liên kết động. Và việc này do Java intepreter thực hiện.

C1D232: một số ngôn ngữ không cần compiler vẫn thực thi được, như Python, nó chỉ có intepreter thôi. Vậy chuyện gì tồi tệ sẽ xảy ra nếu như Java cũng chỉ có Intepreter

A. Không chạy được mã nguồn

B. Sẽ mất rất nhiều thời gian để thực thi mã nguồn

C. Sẽ rất tốn bộ nhớ để không sử dụng compiler

D. Sẽ có lúc không phát hiện lỗi cú pháp.

E. Sẽ không thực hiện được liên kết động

ANC: D

C1D232: một số ngôn ngữ không cần intepreter vẫn thực thi được, như C, nó chỉ có compiler thôi. Vậy chuyện gì tồi tệ sẽ xảy ra nếu như Java cũng chỉ có Compiler

A. Không chạy được mã nguồn

B. Sẽ mất rất nhiều thời gian để thực thi mã nguồn

C. Sẽ rất tốn bộ nhớ để không sử dụng phần còn thiếu đó

D. Sẽ có lúc không phát hiện lỗi cú pháp

E. Sẽ không thực hiện được liên kết động.

ANC: E

Không thực hiện được liên kết động sẽ không có đa hình, không có việc khai báo một đằng và khởi tạo một nẻo: A obj = new B( );

C6D232:

public class Test {

public static void main(String arg[]){

Person p = new Person();

// ...

Employee e = new Employee();

// ...

Manager m = new Manager();

// ...

Person pArr[] = {p, e, m};

for (int i=0; i< pArr.length; i++){

System.out.println(

pArr[i].getDetail());

}

}

}

Nếu người ta không viết Person pArr thì có thể chọn cách nào dưới đây để kết quả vẫn giống hệt mã nguồn của đề bài (Giả sử các lớp Person, Manager và Employee đã cài đặt đúng cú pháp)

A. Viết Manager pArr[ ] = { p, e, m} và pArr[i].getDetail() sẽ là ((Person)pArr[i]).getDetail( )

B. Viết Employee pArr[ ] = { p, e, m} và pArr[i].getDetail() sẽ là ((Employee)pArr[i]).getDetail( )

C. Viết Object pArr[ ] = { p, e, m} và pArr[i].getDetail() sẽ là ((Person)pArr[i]).getDetail( )

D. Không có đáp án nào đúng

E. Viết Object pArr[ ] = { p, e, m} và không phải sửa gì nữa cả

F. Viết Object pArr[ ] = { p, e, m} và pArr[i].getDetail() sẽ là pArr[i]).toString( )

ANC: C

Chẳng hạn ta có mã nguồn:

abstract class Vehicle{

public abstract double getCost( );

}

class Car extends Vehicle{

public double getCost( ) { return 300\*1000; }

}

class Train extends Vehicle{

public double getCost( ) { return 600\*1000; }

}

class Plane extends Vehicle{

public double getCost( ) { return 1000\*1000; }

}

public class Program{

public static void main(String args[ ] ){

if(args[1].equals(“Car")) {

Car obj = new Car( );

Sysout(obj.getCost( )) ;

}

if(args[1].equals(“Train")) {

Train obj = new Train( );

Sysout(obj.getCost( )) ;

}

if(args[2].equals(“Plane")) {

Plane obj = new Plane( );

Sysout(obj.getCost( )) ;

}

}

}

Nếu sử dụng Polymorphism thì mã nguồn trên sẽ được viết gọn như sau:

public class Program{

public static void main(String args[ ] ){

Vehicle obj = (args[1].equals(“Car") ? new Car( ) :

( args[1].equals(“Train") ? new Train( ) : new Plane( )) );

Sysout(obj.getCost( )) ;

}

}

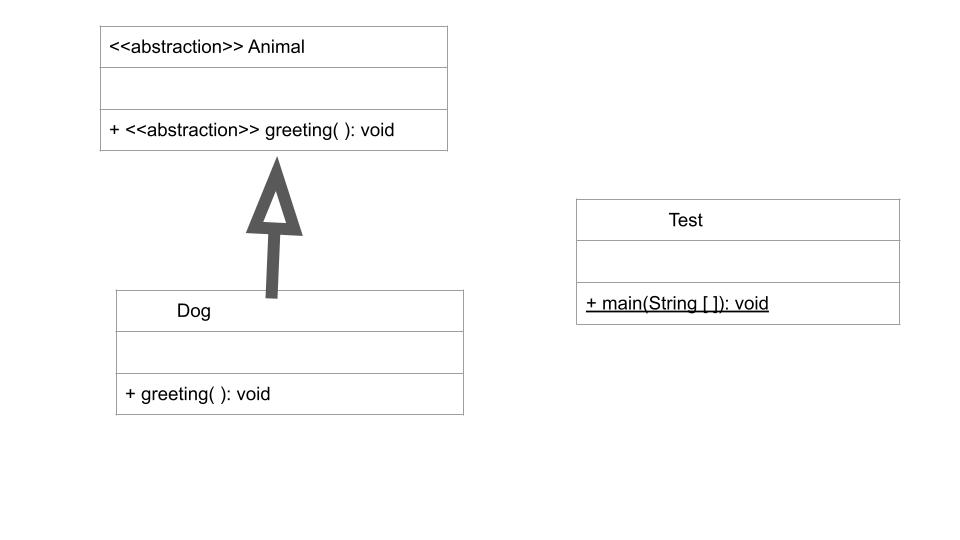
Design Pattern. Là tập hợp 23 mẫu thiết kế thường được sử dụng trong OOP. Do nhóm mang tên là GangOfFour đề xuất. Họ là nhóm 4 lập trình viên đời đầu của hướng đối tượng (C++), họ phát hiện ra rằng nhiều công việc viết code với C++ của họ thì đều quy về 1 trong 23 cách thiết kế khác nhau (hoặc sử dụng hỗn hợp).

Nhóm tác giả đó đã viết ra 23 cách đó và họ đặt tên là Design Pattern. Các LTV khác nhận ra rằng viết code theo Design Pattern đã giúp mã nguồn cuả họ trở nên dễ dàng nâng cấp, dễ chỉnh sửa, dễ bảo trì.

Nếu đi về OOP thì hãy sử dụng DP để viết mã nguồn.

DP bản chất của nó chỉ là sử dụng khéo léo Polymorphism, lớp Abstract, Interface (trong các NN không hỗ trợ Đa kế thừa) để ghép lại thành 23 mẫu.

một phần của Biểu lớp ở trang 36, chương 8:



=================================14/5=================================

Hôm sau sẽ sang chương 8

Đã đến slide trang 44 của chương 7

Phần còn lại SV tự đọc

Ta xây dựng phần mềm quản lý đăng ký tín chỉ của SV, ta sẽ có các tác nhân tham gia vào việc dạy và học tín chỉ: Student và Lecturer, cùng lắm có thêm Admin. Thì các lớp này đều có chung các đặc điểm của Person (tên, tuổi, cccd, sdt, email…)

Vậy Person sẽ được Student, Lecturer và Admin kế thừa. Vậy Person sẽ lớp abstract hay là không abstract??? Nếu Person là lớp bình thường (không abstract) thì khi hệ thống vận hành, sẽ vẫn có thể tạo ra đối tượng của Person mà ko phải là thể hiện của Student, Lecturer và Admin. Person p = new Person( )

Có điều như thế thì hoá ra vẫn tồn tại một người nào đó (không phải là Student, ko phải là Lecturer và ko phải là Admin) vẫn có quyền tham gia vào hệ thống tín chỉ???

Việc để một đối tượng của Person tồn tại như thế chỉ gây khó hiểu cho người đọc code về sau và nó vi phạm nguyên lý trừu tượng hoá của OOP (xem chương 3). Do đó Person bắt buộc phải là abstract.

| <<abstraction>> Animal |
| --- |
|  |
| + <<abstraction>>communicate( ) : void |

Giả sử ta có lớp trừu tượng A, interface I, có lớp B, C, D, E kế thừa A và chỉ có C, D là cài đặt giao diện I thôi

B obj1 = new B( )

C obj2 = new C( )

D obj3 = new D( )

E obj4 = new E( )

A arr = new A[4];//vẫn được phép

I brr = new I[2]; //vẫn được phép

ta có thể gán: arr[0] = obj1; arr[1] = obj2; arr[2] = obj3; arr[3] = obj4;

Được vì B, C, D, E kế thừa A

ta thậm chí có thể gán: brr[0] = obj2; brr[1] = obj3; //Được vì C, D cài đặt interface I

Nhưng không thể gán brr[0] = obj1 cũng như ko thể brr[1] = obj4 được. Vì B và E KHÔNG cài đặt giao diện I.

Phần mềm quản lý công việc của một công ty, trong đó có quản lý nhiều loại nhân viên (Employee) khác nhau. Nhân viên có thể bao gồm các công nhân (Worker) làm việc tại nhà máy, nhân viên văn phòng (Staff) làm việc tại văn phòng để phục vụ các nhu cầu liên quan giấy tờ. Nhân viên kinh doanh (Sale) đi tìm khách hàng. Loại nhân viên cuối cùng là người quản lý (Manager) để đảm bảo công việc thông suốt

Có hệ thống con SalarySubsystem thực hiện việc trả lương cho tất cả nhân viên. Hệ thống này có phương thức transfer( ) nhận vào tham số là mảng các nhân viên toàn công ty và trả về số tiền cần chuyển khoản trong một tháng

Có hệ thống con BonusSubsystem thực hiện việc thưởng doanh thu, hệ thống xác định số tiền thưởng cho nhân viên. Hệ thống này cũng có phương thức transfer chỉ nhận vào MỘT tham số là các nhân viên của toàn công ty và trả về số tiền cần dùng để thưởng

Hàng năm, một người quản lý có yêu cầu lập ra danh sách các nhân viên tham gia vào việc viết báo cáo. Lớp Manager sẽ có PT involveReport( ) trả về danh sách các nhân viên sẽ tham gia vào việc này. Biết rằng chỉ có Staff hoặc Sale có khả năng viết báo cáo.

1. Trong số các mô tả ở trên, hãy xác định các lớp có trong mã nguồn:

2. Theo em để thiết kế được hệ thống như trên thì chúng ta cần thêm giao diện nào, hãy xác định chi tiết (tên, phương thức) của giao diện đó

3. Hãy xác định xem hệ thống có các lớp trừu tượng nào không? Nếu có hãy liệt kê danh sách các thuộc tính của nó.

4. Hãy xây dựng biểu đồ lớp cho hệ thống trên (+1.5 BTL).

Một hệ thống lọc email (Filter) trao đổi giữa các nhân viên và khách hàng. Mỗi người đó có một tài khoản trên hệ thống với các thông tin: tên (name), tuổi (age), tên email (email), số căn cước công dân (pid). Nhân viên (Employee) có thêm thông số định danh nhân viên (eid), cấp bậc (level kiểu int). Khách hàng (Customer) có thêm thông tin mã khách hàng (cid) và doanh thu từ khách hàng đó (income kiểu double). Còn có một loại tài khoản nữa là tài khoản không nhận tin phản hồi (NoReply), tài khoản này chỉ có thuộc tính mang tên là email.

Khi một thông điệp được gửi từ người này đến người khác, thông điệp đó có một người gửi (sender) và ít nhất một người nhận (rec). Mỗi thông điệp có thông số về tiêu đề (title), nội dung (content), thời điểm gửi (time kiểu long).

Hệ thống lọc email sẽ tiến hành kiểm tra thông điệp với phương thức check nhận đầu vào là thông điệp và trả về true nếu thông điệp kia là hợp lệ (không phải spam). Phương thức classify trong hệ thống lọc sẽ nhận đầu vào là thông điệp, rồi trả về đối tượng của lớp Employee hoặc Customer nếu phía gửi mail là con người và họ gửi thông điệp không hợp lệ. Phương thức đó trả về null ở các trường hợp còn lại. Hệ thống lọc mail này còn có phương thức getNotifier, nhận đầu vào là mảng các thông điệp, và trả về một mảng các đối tượng kiểu NoReply hoặc Employee. Mảng này đại diện cho các đối tượng được trao quyền gửi các tin nhắn chúc mừng ngày lễ đến khách hàng.

a) Trong mô tả của đề bài, theo em có bao nhiêu lớp cần được xây dựng. Hãy liệt kê tên và danh sách các thuộc tính của chúng. Chú ý chỉ cần liệt kê tên ra chứ chưa cần vẽ biểu đồ lớp

b) Trong các lớp của câu (a) có bao nhiêu lớp là trừu tượng và các lớp nào sẽ kế thừa nó. Chú ý chỉ cần liệt kê tên ra thôi

c) Trong mô tả của đề bài, theo em có bao nhiêu giao diện (Interface) cần được xây dựng. Liệt kê ra danh sách các lớp cài đặt các giao diện này (nếu có) (+2 BTL)

1 số bạn đã xác định danh sách các lớp, 01 lớp trừu tượng, các interface (2 Interface). Các bạn còn thiếu mối quan hệ aggregation. Cụ thể hơn trong các lớp của đề bài, có 1 lớp mà cần phải được 2 lớp khác kết tập bên trong.

Dấu hiệu nhận biết 2 lớp có quan hệ association (mà Aggregation với Composition chỉ là trường hợp đặc biệt của association thôi) là lớp A có đối tượng hoặc mảng đối tượng làm thuộc tính của lớp B.

Thế nào là mối quạn hệ dependency: lớp A được dùng làm tham số (hoặc giá trị trả về hoặc biến cục bộ) cho PT của lớp B thì ta nói A và B có quan hệ dependency. Nhưng quan hệ này không cần phải vẽ trong class diagram mà nó thường xuất hiện trong collaboration diagram.

=================================7/5=================================

Đã đến slide trang 28 của chương 7

Đã giảng cho SV một lý do của

class A{

}

class B extends A{

public B( ) { } //Trong PTKD của B thì nó không cần chỉ rõ ra nó cần gọi

//PTKD của A

}

class C extends B {

public C( ) { } //PTKD của C không cần chỉ rõ ra nó cần gọi pTKD nào của B

public C(int x) { }

}

class D extends C{

public D( ) { }

//Ở đây JVM sẽ ko biết phải gọi đến PTKD của C nếu khởi tạo đối tượng của D

}

class E extends C{

//Ở đây JVM sẽ ko biết phải gọi đến PTKD của C nếu khởi tạo đối tượng của E

}

class F extends B{

public F(int x) { }//JVM ko báo lỗi vì JVM biết phải gọi đến PTKD nào của B

}

class G extends F{

public G( ) { }

//JVM báo lỗi. Vì sao? Dont be shy

//Mặc dù JVM biết F chỉ có 1 ptkd duy nhất nhưng pt này có tham số và JVM ko

//biết phải truyền tham số gì vào?

}

tốt nhất ta phải chỉ rõ cho JVM, chẳng hạn public G( ) { super(3); }

thì tương đương với việc gọi PTKD của F với tham số x = 3

Với bài tập cuối chương 6, khi viết mã nguồn, ta cần viết PTKD của lớp PhongBan với một tham số NhanVien, sau đó là một mảng nhân viên. Vì làm vậy mới đặc tả được ký hiệu NhanVien kết tập trong PhongBan với bội số quan hệ của NhanVien là 1..\*

class PhongBan{

private String tenPhongBan;

private byte soNhanVien;

public static final byte SO\_NV\_MAX = 100;

public String getTenPhongBan( ){

return this.tenPhongBan;

}

public void setTenPhongBan(String tenPhongBan ){

this.tenPhongBan = tenPhongBan;

}

public byte getSoNhanVien( ){

return this.soNhanVien;

}

public void setSoNhanVien(byte soNhanVien){

this.soNhanVien = soNhanVien;

}

private NhanVien nhanVien[ ];

public PhongBan(NhanVien nv, NhanVien nhanVien[ ]){

int N = nhanVien.length( ) + 1;

N = N > SO\_NV\_MAX ? SO\_NV\_MAX : N;

this.nhanVien = new NhanVien[SO\_NV\_MAX];

this.nhanVien[0] = nv;

for(int i = 1; i < N; i++){

this.nhanVien[i] = nhanVien[i-1];

}

this.soNhanVien = N;

}

public boolean themNV(NhanVien nv){

int N = this.nhanVien.length( ) + 1;

if(N > SO\_NV\_MAX)

return false;

this.nhanVien[N-1] = nv;

this.soNhanVien++;

return true;

}

public NhanVien xoaNV( ) {

if(this.soNhanVien == 0)

return null;

NhanVien tmp = this.nhanVien[this.soNhanVien - 1];

this.soNhanVien = this.soNhanVien - 1;

return tmp;

}

public double tongLuong( ) {

doule sum = 0;

for(int i = 0; i < this.soNhanVien; i++){

sum = sum + this.nhanVien[i].tinhLuong( );

}

return sum;

}

public void inTTin( ) {

System.out.println(this.tenPhongBan);

for(int i = 0; i < this.soNhanVien; i++){

this.nhanVien[i].inTTin( ) ;

}

}

}

=================================23/4=================================

Đã đến slide trang 42 của chương 6

Hãy xem ví dụ dưới đây về việc tái sử dụng mã nguồn trong lập trình cấu trúc

Giả sử ta có struct Person với

typedef struct Person{

char name[255];

char id[255];

int age;

double height, weight;

} Person;

Sau đó ta cần khai báo hai cấu trúc là Doctor và Patient với việc chúng cũng có các thông tin cần lưu trữ như Person.

Vậy nếu ta không tái sử dụng mã nguồn thì trong Doctor và Patient lại phải khai báo lại 5 thông số trên

typedef struct Doctor{

char name[255];

char id[255];

int age;

double yearsOfExperience;

char department[255];

double height, weight;

} Doctor;

Để tái sử dụng mã nguồn Person trong LT CT thì ta chỉ có cách

typedef struct Doctor{

Person basicInfo;

double yearsOfExperience;

char department[255];

} Doctor;

| //C  Person p;  if( hasSibling(p)) {  } | //C++  Person p;  if(p.hasSibling( )){  } |
| --- | --- |

<https://shorturl.at/FLP56>

C6K232: Có các lớp Graph, Vertex, Edge và Car trong đó:

Vertex sẽ kết tập trong Graph, bội số quan hệ của Vertex là 1..\*, và của Graph là 1

Edge sẽ kết tập trong Graph, bội số quan hệ của Edge là 1..\*, của Graph là 1

Vertex sẽ kết tập trong Edge, bội số quan hệ của Vertex là 2, của Edge là 1

Car có liên kết đến Graph, bội số quan hệ của Car là 1..\*, của Graph là 1.

Trong Graph có phương thức getOriginalSource(Car) nhận đầu vào là đối tượng c của Car và trả về đối tượng v của Vertex vốn là điểm xuất phát ban đầu của c.

Biết rằng thường thì số đối tượng của lớp Vertex trong Graph sẽ nhiều hơn hẳn số đối tượng của Car trong Graph. Người ta có các lựa chọn thiết kế mối quan hệ như sau:

A. Để Vertex liên kết một chiều đến với Car, bội số quan hệ của Car là 1..\*, và của Vertex là 1. Vai trò của Car là cars. Một chiều ở đây ám chỉ Vertex chỉ truy cập được đến Car chứ không phải ngược lại

Ta không dùng lựa chọn thiết kế này vì để phương thức trên tìm được đáp án thì nó phải tìm trong m đối tượng vertex của Graph, rồi với từng vertex đó lại phải tìm trong n đối tượng Car mà có liên kết đến => độ phức tạp tính toán trong trường hợp tồi nhất là O(m\*n)

B. Để Car liên kết một chiều đến với Vertex, bội số quan hệ của Car là 1..\*, và của Vertex là 1. Vai trò của Vertex là originalSource. Một chiều ở đây ám chỉ Car chỉ truy cập được đến Vertex chứ không phải ngược lại

C. Để Vertex kết tập trong Car, bội số quan hệ của Car là 1..\*, và của Vertex là 1. Vai trò của Car là cars. Vai trò của Vertex là originalSource

Đáp án C cũng sẽ gây ra độ phức tạp tính toán giống đáp án A. Cái tệ hơn của đáp án C là gì? Nó gây ra hiểu nhầm rằng Car chỉ có thể là Car nếu đã biết được vertex (vô lý)

D. Để Car kết tập trong Vertex, bội số quan hệ của Car là 1..\*, và của Vertex là 1. Vai trò của Car là cars. Vai trò của Vertex là originalSource

a) Trong các đáp án trên, đáp án nào đúng?

b) Hãy vẽ biểu đồ lớp cho đáp án mà em thấy đúng

ANC: B

Nếu liên kết mà ko nói gì thì đó là liên kết 2 chiều

Nếu lớp B có liên kết một chiều đến với A thì có điều đó có nghĩa là gì?

class A{

}

class B{

private A x;

}

Có nghĩa tên vai trò của A trong liên kết là x, bội số quan hệ của A là 1, B có liên kết một chiều đến A có nghĩa chỉ có B truy cập được A (B có thuộc tính là đối tượng của A) còn A không truy cập (không có thuộc tính là đối tượng của B)

Kết tập là trường hợp đặc biệt của liên kết. Như ở trên, ta nói B liên kết đến A chứ không nói A kết tập trong B. Như dưới đây thì ta có thể nói A kết tập trong C

class C{

private A x;

public C(A x) { this.x = x ; }

}

==================================9/4=================================

Cho đoạn mã nguồn sau:

Impor t java.util.\*;

class A{

public void m(String x) { Sysout(x); }

public void m(int x) { Sysout(x); }

}

public class Program{

public static void main(String args[ ]) {

Scanner sc = new Scanner(System.in);

int N = sc.nextInt( );

A obj = new A( );

if(N == 1){

obj.m("2");

}

else{

obj.m(2);

}

}

}

================================2/4===================================

Ngày 2/4 đã học xong slide trang 38 của chương 4

A[ ] arr = new A[100] ;

Nhiều người nghĩ rằng đây đã là khởi tạo => không phải. Câu lệnh trên chỉ khởi tạo mảng arr với tất cả các phần tử bằng null.

Tất nhiên thỉnh thoảng các LTV có kinh nghiệm vẫn quên không khởi tạo đối tượng (là biến riêng rẽ) khi viết mã.

Cũng rất may rằng các IDE phổ biến hiện nay đều có cảnh báo khi một đối tượng được khai báo nhưng chưa khởi tạo.

Với các đối tượng, ngoài toán tử gán (=) thì người LTV cần phải sử dụng cả toán tử new

Câu 4D232: Phương thức nào không cần phải dùng toán tử chấm để gọi nó

A. Phương thức khởi tạo

B. Phương thức tĩnh

C. Phương thức trừu tượng

D. Phương thức private

ANC: A

Câu 4D232: Đoạn mã sau sẽ in ra gì?

class A{

public void A( ) { System.out.println(“A”); }

}

Trong phương thức main, ta viết:

public static void main(String args[ ]){

A obj = new A( );

}

A. sẽ in ra “A”

B. Sẽ báo lỗi vì chưa khai báo phương thức khởi tạo

C. Sẽ báo lỗi vì không thể tạo phương thức khởi tạo mặc định

D. Sẽ không in ra gì cả.

ANC: D

Vì void A( ) không phải là PTKD, nên JVM vẫn tạo ra được PTKD mặc định cho lớp A

Phương thức khởi tạo có cách gọi khác hẳn với các phương thức khác, nên ta không coi PTKT là thành viên của lớp.

Câu 4M232: Có những ngôn ngữ có cho phép truyền tham số không theo đúng thứ tự của khai báo, chẳng hạn:

public void m(String a, float b) {

}

Và khi gọi phương thức obj.m(b: 7.3, a : “John”);

Theo em phát biểu nào đúng về ưu khuyết điểm của cách gọi trên?

A. Ưu điểm là không cần nhớ thứ tự tham số chứ không có khuyết điểm gì

B. Không có ưu điểm gì cả vì phải mất công nhắc lại tên tham số khi gọi. Khuyết điểm cũng không có

C. Ưu điểm là có thể không cần truyền giá trị mà tham số vẫn nhận giá trị mặc định. Khuyết điểm là không hỗ trợ nạp chồng được.

D. Ưu điểm là không cần nhớ thứ tự tham số và khuyết điểm là không hỗ trợ nạp chồng.

ANC: D

<https://shorturl.at/ryNX8>

//<https://onlinegdb.com/rlKltkpKve>

#include <iostream>

class A{ };

int main()

{ A obj = new A();

A x = A();

return 0;

}

/\*\* Lỗi xảy ra:

\* main.cpp: In function ‘int main()’:

main.cpp:10:13: error: conversion from ‘A\*’ to non-scalar type ‘A’ requested

10 | A obj = new A();

| ^~~~~~~

\* /

C4K232: Với mã nguồn C++ sau

#include <iostream>

class A{ };

int main()

{ A obj = new A();

A x = A();

return 0;

}

A. Mã nguồn trên sẽ báo lỗi do obj phải là con trỏ

B. Mã nguồn trên sẽ báo lỗi do A( ) phải đi kèm với new

C. Mã nguồn trên sẽ báo lỗi do không được phép viết ; sau }

D. Mã nguồn trên sẽ báo lỗi do C++ không tự tạo PTKD mặc định

ANC: A

C4K232: Một số ngôn ngữ dạng script có cho phép LTV không cần viết PTKT có tham số mà vẫn gọi được PTKT đó. Nói cách khác là ngôn ngữ đó có hỗ trợ tự tạo ra PTKT có tham số chứ không chỉ PTKT mặc định

Theo em khó khăn nào khiến Java không hỗ trợ đặc điểm này?

A. Làm thế sẽ khiến LTV lười viết PTKT có tham số

B. Làm vậy thì JVM sẽ hoạt động rất chậm vì rất khó để tạo ra một PTKT có tham số

C. Đề sai, Java có hỗ trợ tự tạo PTKT có tham số

D. Do sẽ có thể phải tạo ra rất nhiều PTKT có tham số

ANC: D

Do Java có hỗ trợ nạp chồng, nên nếu một lớp có 5 thuộc tính với 5 kiểu khác nhau chẳng hạn thì sẽ có tới bao nhiêu PTKT có tham số cần được tạo ra? 5! = 120 PTKT có 5 tham số.

Với các ngôn ngữ không hỗ trợ nạp chồng thì chỉ cần tạo ra một PTKT có 5 tham số. Và các tham số có thể truyền với thứ tự thay đổi được.

C4D232: Theo em đâu là đáp án đúng?

public class Main

{

public static void main(String[] args) {

String x = "Hello";

String y = new String("Hello");

String z = "He" + "llo";

if(x == z){

System.out.println("true");

}

else{

System.out.println("false");

}

}

}

A. Mã nguồn trên sẽ in ra “false” vì x và z sẽ trỏ đến 2 địa chỉ khác nhau

B. Mã nguồn trên sẽ lỗi và không chạy được

C. Mã nguồn trên sẽ in ra “true” vì cả x và z đều trỏ đến địa chỉ của “hello”

D. Tất cả các đáp án trên đều sai.

ANC: D

Hãy để ý đến; <https://onlinegdb.com/1Jbpe8qf2>

~~C4D232: Dưới đây là mã nguồn của một ngôn ngữ có cú pháp giống Java. Theo em phát biểu nào đúng về việc nên sử dụng this. để gọi đến phương thức của lớp~~

~~public void credit(double amount){~~

~~…. ///có nhiều dòng mã nguồn ở đây~~

~~this.setBalance(this.getBalance( ) + amount);~~

~~}~~

~~A. Vì lý do bảo mật, ta nên sử dụng this. để đảm bảo việc truy cập an toàn~~

~~B. Vì lý do tốc độ, ta nên sử dụng this. để đảm bảo việc truy cập nhanh nhất có thể~~

~~C. Vì lý do phân biệt, nếu ngôn ngữ đó có hỗ trợ khai báo phương thức trong phương thức~~

~~D.~~

======================================================================

Ngày 26/3 đã học xong slide trang … của chương 3

C3D232: Khi xây dựng lớp, công việc nào sau đây nên được thực hiện trước

A. Khai báo thuộc tính trước

B. Khai báo phương thức trước

ANC: A

<https://www.omg.org/spec/UML/2.5.1/PDF>

Bài 1: Viết chương trình Java hoặc C++ thực hiện khai báo lớp Rectangle với lớp Point trong đó Point sẽ có 2 thuộc tính private x, y kiểu double. Rectangle có 4 thuộc tính public pA, pB, pC, pD. Hãy viết đầy đủ get, set ở Point còn ở Rectangle thì có thể bỏ qua.

Hãy khai báo phương thức public boolean collide(Rectangle r) trả về true nếu hai HCN này va chạm nhau (phương thức này bên C++ có thể có kiểu dữ liệu khác). Hãy giả sử rằng hai HCN này có cạnh song song trục tọa độ

======================================================================

Ngày 19/3 đã học xong slide trang 98 của chương 2

Bài 1: Người ta làm game Puzzle với tổ hợp bắt đầu là các số trong các ô được sắp xếp lộn xộn và có một ô trống

| 1 | 5 | 3 |
| --- | --- | --- |
|  | 7 | 8 |
| 4 | 6 | 2 |

Yêu cầu cho người chơi là phải di chuyển các ô số sao cho đưa về được tổ hợp gốc sau

|  | 1 | 2 |
| --- | --- | --- |
| 3 | 4 | 5 |
| 6 | 7 | 8 |

Nhưng vấn đề là người ta đã chứng minh được không phải tổ hợp nào cũng đưa về được tổ hợp gốc. Ví dụ như tổ hợp dưới đây

|  | 1 | 2 |
| --- | --- | --- |
| 4 | 3 | 5 |
| 6 | 7 | 8 |

Hãy cài đặt thuật toán kiểm tra xem một tổ hợp có đưa được về tổ hợp gốc hay không? Nếu N lẻ (N số ô hàng ngang và dọc) thì ta cần đưa tổ hợp về dạng mảng một chiều sau đó đếm số lần số lớn đứng trước số bé trong tổ hợp đó. Nếu số lần này không cùng tính chất chẵn lẻ với tổ hợp gốc thì tổ hợp đó không đưa về tổ hợp gốc được.

Vậy tổ hợp gốc có số lần số lớn đứng trước số bé là bao nhiêu? 1,2,3,4,5,6,7,8 => 0 lần

Hãy cài đặt mã nguồn Java/C++ cho phép người dùng nhập N, sau đó nhập N dòng, mỗi dòng nhập N số phân tách bằng dấu cách. Số 0 sẽ đại diện cho ô trống. Giả sử rằng người dùng luôn nhập các số sao cho: (i) không trùng, (ii) từ 0 đến N\*N - 1 và (iii) N lẻ

Hãy in ra câu trả lời No nếu không thể đưa tổ hợp vừa nhập về tổ hợp gốc. In ra Yes nếu ngược lại

Input:

3

0 1 2

4 3 5

6 7 8

Output:

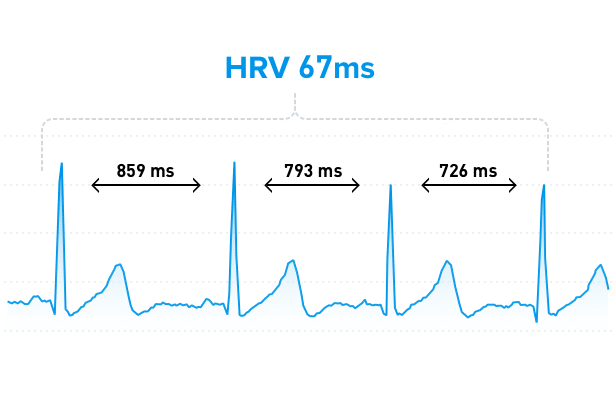
No

Chú ý rằng khi đếm số lần số lớn đứng trước số bé thì không tính số 0

| Java  //Program.java  import java.util.\*;  public class Program{  public static void main(String [ ] args){  int N;  Scanner sc = new  Scanner(System.in);  N = sc.nextInt();  int arr[N][N] = new int[N][N];  int b\_arr[N\*N] = new int[N\*N];  for(int i = 0; i < N; i++){  for(int j = 0; j < N; j++){  int x = sc.nextInt( );  arr[i][j] = x;  b\_arr[i\*N + j] = x;  }  }  int count = 0;  for(int i = 0; i < N\*N; i++){  for(int j = i+1; j < N\*N; j++)  if(b\_arr[i] > b\_arr[j]  && b\_arr[j] > 0){  count++;  }  }  if(count % 2 == 0)  Sysout(“Yes”);  else  Sysout(“No”);  }  } | C++  //Program.cpp  int main( ){  int N;  cin>>N;  int arr[N][N];  int b\_arr[N\*N];  for(int i = 0; i < N; i++)  for(int j = 0; j < N; j++){  cin>>arr[i][j];  b\_arr[i\*N + j] = arr[i][j];  }  int count = 0;  for(int i = 0; i < N\*N; i++){  for(int j = i+1; j < N\*N; j++)  if(b\_arr[i] > b\_arr[j]  && b\_arr[j] > 0){  count++;  }  }  if(count % 2 == 0)  cout<<”Yes”;  else  cout<<“No”;  return 0;  } |
| --- | --- |

Bài 2: Trong y tế làm thế nào để xác định Stress-level của một người và từ đó suy ra xem người đó có đang bị stress hay không?

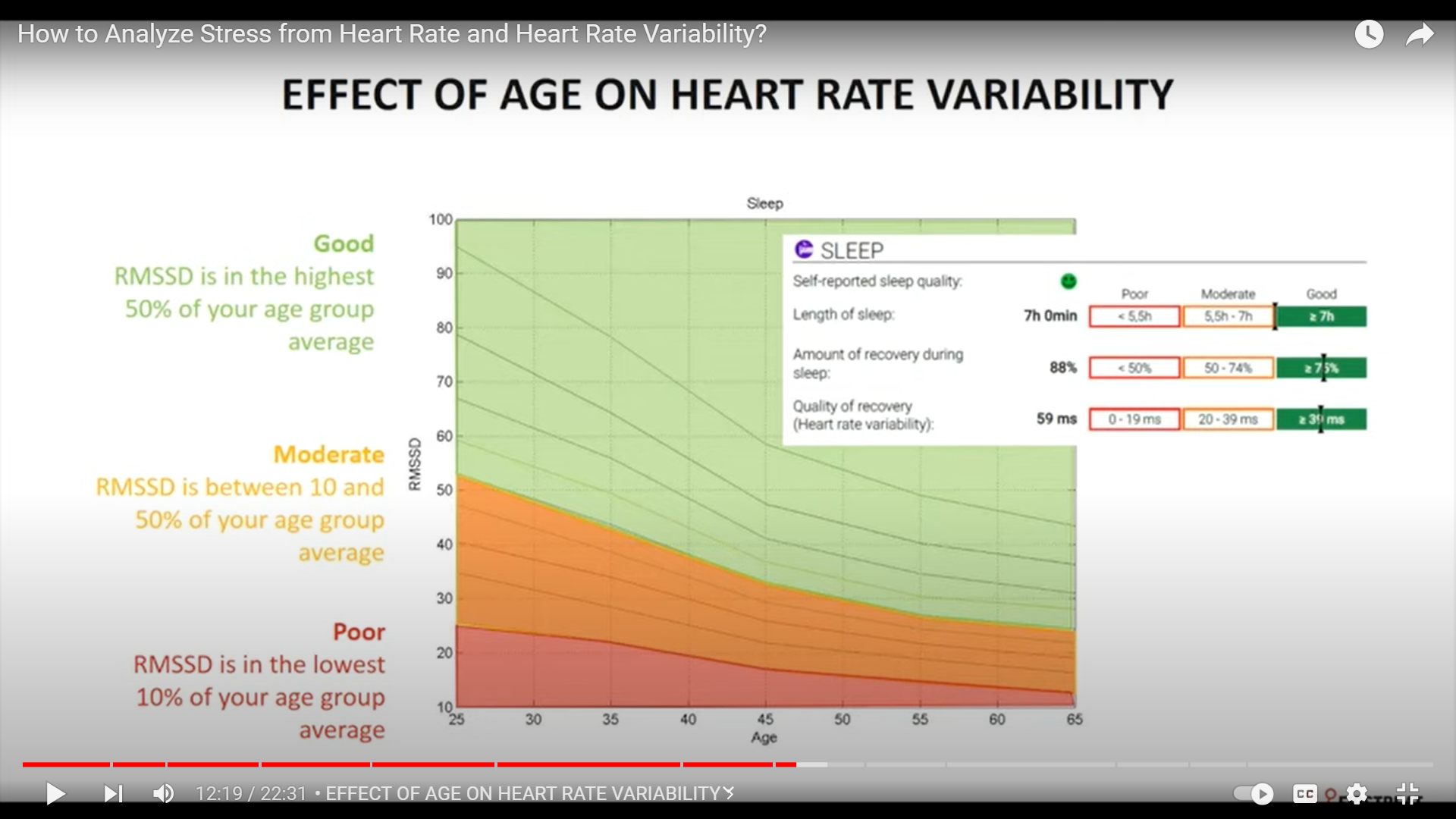
Có nhiều cách, một trong số đó là đo Heart Rate Variability. Chỉ số nhịp tim của một người sẽ có dạng đồ thị răng cưa như sau



Khoảng cách giữa hai lần đo được nhịp tim được gọi là chỉ số RR. Sau đó người ta tính ra chỉ số Root Mean Square of Successive Differences theo công thức

RMSSD = sqrt((RRi+1 - RRi)2/(N-1))

Sau đó người ta cần biết thông tin về độ tuổi của con người và tra bảng dưới đây



Viết mã nguồn Java/C++ cho phép người dùng nhập vào: giá trị N, N phần tử số nguyên đại diện cho giá trị RR, độ tuổi của con người đang đo. Chương trình sẽ in ra “Stress” nếu người đó có chỉ số RMSSD bé hơn hoặc bằng 0.1\*AverageRMSSD. In ra “Normal” nếu ngược lại. Giả sử rằng người dùng luôn nhập các giá trị hợp lệ và tuổi luôn nhập 25

Để tìm căn bậc hai thì ta có thể gọi hàm Math.sqrt(4) sẽ bằng 2

======================================================================

Ngày 20/2 đã học xong slide trang 41 của chương 1

Nguyễn Tiến Thành

Email: [nguyenthanh@soict.hust.edu.vn](mailto:nguyenthanh@soict.hust.edu.vn)

Sdt: 0987768886

Quy tắc:

Sẽ có điểm danh, và ai vắng từ 8 buổi trở lên sẽ 0 QT.

Được phép xin vắng dài hạn các buổi học nhưng phải: (i) làm đơn có kèm bản photo thẻ SV, (ii) nếu đến lớp trong thời gian xin phép vắng thì sẽ phải bị mời ra khỏi lớp; (iii) vẫn phải có mặt ở buổi bảo vệ BTL.

Được phép vắng đột xuất nhưng phải gửi mail (không nhận tin Zalo hoặc Team) trước 6h30 ngày T3. Vắng 2 buổi có phép coi như vắng 1 hôm không phép.

Trong lớp nếu có phát biểu: trả lời câu hỏi hoặc đóng góp câu hỏi (mà hay và có thể dùng cho ngân hàng đề) thì sẽ được cộng điểm vào QT

Sẽ có form đăng ký nhóm BTL gửi cho lớp, nhóm trưởng điền vào form để đăng ký thành viên, form này sẽ được gửi 1 link cho nhóm trưởng để nhóm trưởng edit lại. Do vậy nhóm trưởng chỉ cần submit 1 lần và vào mail lấy link edit thôi.

Bảo vệ BTL sẽ tiến hành vào các buổi cuối, nhóm nào có tổng số ngày nghỉ của các thành viên nhiều nhất sẽ được bảo vệ trước. Chính xác hôm nào bảo vệ thì tùy vào số lượng các nhóm. Bảo vệ có thể tiến hành on/off, cả vào T7, CN (nếu cần)

Khi bảo vệ xong thì cả lớp sẽ nhận được một form báo cáo, tất cả các thành viên phải điền vào form đó. Trong form chắc chắn sẽ có chỗ điền tên/MSSV thành viên, tên/MSSV nhóm trưởng, link github

Nếu nhóm nào bảo vệ trước thì sẽ được cộng điểm bảo vệ sớm, nhưng các nhóm bảo vệ cuối cùng sẽ không được cộng điểm nào.

Nếu nhóm có thành viên xin nghỉ học, xin không nhận điểm hoặc bị cả nhóm khai trừ thì điểm của bạn đó sẽ được chuyển sang các bạn khác. Đó điều đó giải thích tại sao nhiều nhóm làm tốt nhưng vẫn không được điểm cao nhưng nhóm khác.

Mô tả của bài tập lớn vẫn chưa được hoàn thiện, nhưng sẽ cố gắng xong trước 15/03. BTL đã có một phiên bản cũ (ĐATN của SV khóa trước) viết bằng C++, các nhóm được phép tùy chọn công nghệ (Java, Python, Javascript trên web, C/C++) cũng được.

Trong quá trình học, có thể thầy sẽ giao các BT thêm, SV được phép tùy chọn làm hay không làm các BT này, nhưng bạn nộp sớm nhất (chỉ dành cho 01 người) và đúng nhất sẽ được cộng điểm. Thường bài tập này bằng C/C++

Slide bài giảng:

<https://drive.google.com/drive/folders/1Dvz2pxLj5DNPeNdVWCNsUNqqJwZx-7Fp?usp=drive_link>

BTL nên code bằng C++:

<https://docs.google.com/document/d/1Xkt-bBM9zOu_kHUOBMuaMd-EJeA0KAmF9vwWCNhNqls/edit?usp=sharing>

Để làm được BTL thì SV cần cài hệ điều hành Linux

C1D232: Không phải hàm nào trong mã nguồn lớp ở C++ cũng được dùng làm thông điệp để đối tượng lớp này truyền thông điệp đến đối tượng của lớp khác. Vậy theo em thì các hàm đó là thuộc loại nào?

A. Hàm ảo

B. Hàm public

C. Hàm private

D. Hàm protected

E. Hàm static

ANC: C

Công cụ Sonar-Q là một công cụ phân tích mã nguồn để kiểm tra xem mã nguồn đó có được thiết kế tốt hay không?