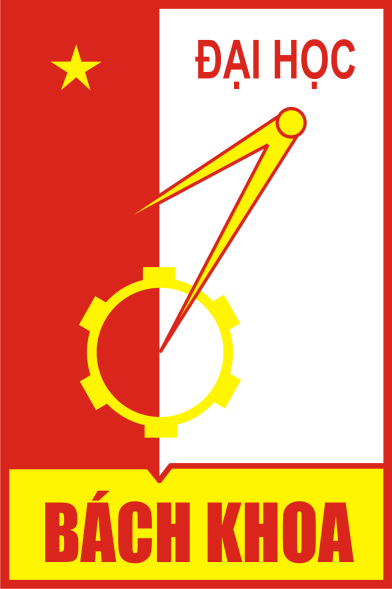
# 

ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHỆ THÔNG TIN – TRUYỀN THÔNG

---🙠🕮🙢---



BÁO CÁO BÀI TẬP LỚN LẬP TRÌNH

HƯỚNG ĐỐI TƯỢNG

Sinh viên thực hiện:

Cao Văn Thiện 20144262

Hoàng Văn Nam 20143022

Trần Văn Thanh 20144000

Nguyễn Văn Ngọc 20143223

Roãn Văn Thụ 20144416

Giảng viên hướng dẫn: ThS Trịnh Thành Trung

HÀ NỘI, 12/2016

**MỤC LỤC**

[1](#_Toc468963875)

[Chương I - Tổng quan 3](#_Toc468963876)

[I. Giới thiệu về đề tài 3](#_Toc468963877)

[1. Đề bài 3](#_Toc468963878)

[Chương II - Phân tích yêu cầu 4](#_Toc468963879)

[I. Phân tích đề tài 4](#_Toc468963880)

[1. Mục đích 4](#_Toc468963881)

[2. Yêu cầu và chức năng cần xây dựng 4](#_Toc468963882)

[Chương III - Thiết kế 5](#_Toc468963883)

[I. Các lớp cùng phương thức và thuộc tính chính 5](#_Toc468963884)

[1. Class CauHoi 5](#_Toc468963885)

[2. Class CauHoiTL 5](#_Toc468963886)

[3. Class CauHoiTN 6](#_Toc468963887)

[4. Class DapAnTN 7](#_Toc468963888)

[5. Class DeThi 7](#_Toc468963889)

[6. Class MonHoc 7](#_Toc468963890)

[7. Class LayDuLieu 8](#_Toc468963891)

[8. Class LuuDanhSachMonHocRaNgoai 9](#_Toc468963892)

[10. Các class giao diện 9](#_Toc468963893)

[II. Biểu đồ và mối quan hệ các lớp 10](#_Toc468963894)

[Chương IV - Kết quả đạt được 12](#_Toc468963895)

[I. Các chức năng đã xây dựng được 12](#_Toc468963896)

[1. Giao diện soạn câu hỏi 12](#_Toc468963897)

[2. Giao diện soạn đề thi 13](#_Toc468963898)

[3. Giao diện xuất đề thi 14](#_Toc468963899)

[II. Các hướng phát triển 15](#_Toc468963900)

# Chương I - Tổng quan

## I. Giới thiệu về đề tài

### 1. Đề bài

Đề tài: 3

Xây dựng hệ thống tạo đề thi cho giáo viên. Hệ thống có khả năng quản lý một ngân hàng câu hỏi bao gồm cả câu hỏi trắc nghiệm và câu hỏi tự luận. Mỗi câu hỏi tự luận có phần đề bài và gợi ý đáp án. Mỗi câu hỏi trắc nghiệm có phần đề bài, các phương án trả lời và đáp án của câu hỏi đó. Số lượng các phương án là không giới hạn. Một câu hỏi trắc nghiệm cũng có thể có nhiều đáp án đúng. Yêu cầu mỗi câu hỏi phải thuộc về một môn học nhất định. Mỗi câu hỏi cũng có thể được quy định một độ khó, nằm trong một hay nhiều chương của môn học đó. Khi sinh đề thi, giáo viên có thể tạo đề thi trộn lẫn các câu hỏi trắc nghiệm và tự luận hoặc có các phần tự luận và trắc nghiệm riêng biệt. Các câu hỏi được chọn bằng tay hoặc sinh ngẫu nhiên theo các chương và độ khó giáo viên đã chọn trước.  
Ghi chú: Sinh viên có thể phát triển thêm các chức năng như đảo đáp án, đảo câu hỏi cho cùng một đề, sinh ra file doc hoặc pdf…

# Chương II - Phân tích yêu cầu

## I. Phân tích đề tài

### 1. Mục đích

Xây dựng một chương trình cho phép giáo viên có thể tạo ra đề thi

### 2. Yêu cầu và chức năng cần xây dựng

* Hệ thống có khả năng quản lý một ngân hàng câu hỏi bao gồm cả câu hỏi trắc nghiệm và câu hỏi tự luận
* Câu hỏi tự luận có phần đề bài và gợi ý đáp án.
* Câu hỏi trắc nghiệm có câu hỏi, phương án trả lời, đáp án câu hỏi, không giới hạn phương án trả lời và mỗi câu có thể có nhiều đáp án đúng.
* Mỗi câu hỏi phải thuộc một môn học nhất định
* Mỗi câu hỏi có một độ khó nhất định
* Câu hỏi nằm trong một hay nhiều chương của môn học.
* Đề thi có thể trộn lẫn các câu hỏi trắc nghiệm và tự luận hoặc có các phần tự luận và trắc nghiệm riêng biệt.
* Có ba kiểu đề: trắc nghiệm, tự luận, đề tổng hợp có cả phần trắc nghiệm và tự luận. .
* Các câu hỏi được chọn bằng tay hoặc sinh ngẫu nhiên theo các chương và độ khó giáo viên đã chọn trước
* Các câu hỏi trong đề có thể trộn lẫn lộn
* Có thể đảo đáp án trong câu trắc nghiệm
* Đề thi tạo xong có thể xuất ra file doc, pdf, txt…

# Chương III - Thiết kế

## I. Các lớp cùng phương thức và thuộc tính chính

### 1. Class CauHoi

public abstract class CauHoi implements Serializable{

String deBai;

int doKho;

int chuong;

public abstract String inCauHoi();

public abstract String inCauHoi(int x);

}

}

* Lớp trừu tượng CauHoi implements giao diện Serializable có 2 phương thức trừu tượng inCauHoi có tham số và không tham số
* Các thuộc tính deBai, doKho, chuong
* Các phương thức get, set deBai, doKho, chuong
* Kĩ thuật nạp chồng phương thức inCauHoi có tham số và không có tham số

### 2. Class CauHoiTL

public class CauHoiTL extends CauHoi {

String goiYDapAn;

public CauHoiTL(String goiYDapAn, String deBai, int doKho, int chuong){

super(deBai, doKho, chuong);

goiYDapAn;

}

@Override

public String inCauHoi() {

}

@Override

public String inCauHoi(int x) {

}

}

* Lớp CauHoiTL kế thừa abstract class CauHoi, thêm thuộc tính goiYDapAn
* Ghi đè phương thức inCauHoi() và inCauHoi(int x) của lớp cha CauHoi

### 3. Class CauHoiTN

public class CauHoiTN extends CauHoi {

ArrayList<DapAnTN> arrCacDapAn;

public CauHoiTN(ArrayList<DapAnTN> arrCacDapAn, String deBai, int doKho,int chuong) {

super(deBai, doKho,chuong );

this.arrCacDapAn = arrCacDapAn;

}

@Override

public String inCauHoi() {

}

@Override

public String inCauHoi(int x) {

}

}

* Lớp CauHoiTN kế thừa abstract class CauHoi, thêm arrCacDapAn
* Sử dụng kĩ thuật kết tập: Lớp này bao gồm các đối tượng thuộc lớp DapAnTN
* Ghi đè phương thức inCauHoi() và inCauHoi(int x) của lớp cha CauHoi

### 4. Class DapAnTN

public class DapAnTN {

private String noiDungDA;

private boolean isTrue = false;

}

* Thuộc tính noiDungDA, isTrue gán bằng false xét tính đúng sai của đáp án
* Các phương thức get, set

### 5. Class DeThi

public class DeThi {

private String tenDeThi;

private String mon;

private ArrayList<CauHoi> dsCauHoi=new ArrayList<>();

public String inDeThi(){

return deThi;

}

}

* Các thuộc tính tenDeThi, mon.
* Khai báo mảng ArrayList kiểu CauHoi, khởi tạo đối tượng
* Các phương thức get, set và inDeThi()
* Sử dụng kĩ thuật kết tập các đối tượng lớp CauHoi

### 6. Class MonHoc

public class MonHoc implements Serializable{

String tenMonHoc;

ArrayList<CauHoi> arrCauHoi=new ArrayList<>();

ArrayList<DeThi> dsDeThi=new ArrayList<>();

public String toString() {

return this.tenMonHoc

}

}

* Thuộc tính tenMonHoc
* Khai báo mảng ArrayList kiểu CauHoi và kiểu DeThi, khởi tạo đối tượng các mảng đó
* Các phương thức get, set và toString
* Sử dụng kĩ thuật kết tập đối tượng lớp CauHoi và DeThi

### 7. Class LayDuLieu

public class LayDuLieu {

ArrayList<MonHoc> dsMonHoc=new ArrayList<>();

MonHoc mh=null;

public LayDuLieu() {

layDuLieuMonHoc();

}

public LayDuLieu(int indexMonHoc) {

layDuLieuMonHoc();

mh=dsMonHoc.get(indexMonHoc);

}

private void layDuLieuMonHoc() {

}

public ArrayList<MonHoc> getDanhSachMonHoc(){

return dsMonHoc;

}

public String[] toArrayString(){

return listMonHoc;

}

public String[] toArrayString(int indexMonHoc){

return listCauHoi;

}

}

* Khai báo mảng ArrayList kiểu MonHoc, khởi tạo tượng
* Gán mh kiểu MonHoc bằng null
* Các constructor LayDuLieu; phương thức layDuLieuMonHoc
* Kĩ thuật nạp chồng phương thức toArrayString

### 8. Class LuuDanhSachMonHocRaNgoai

public class LuuDanhSachMonHocRaNgoai {

private ArrayList<MonHoc> dsMonHoc;

public LuuDanhSachMonHocRaNgoai(ArrayList<MonHoc> dsMonHoc) {

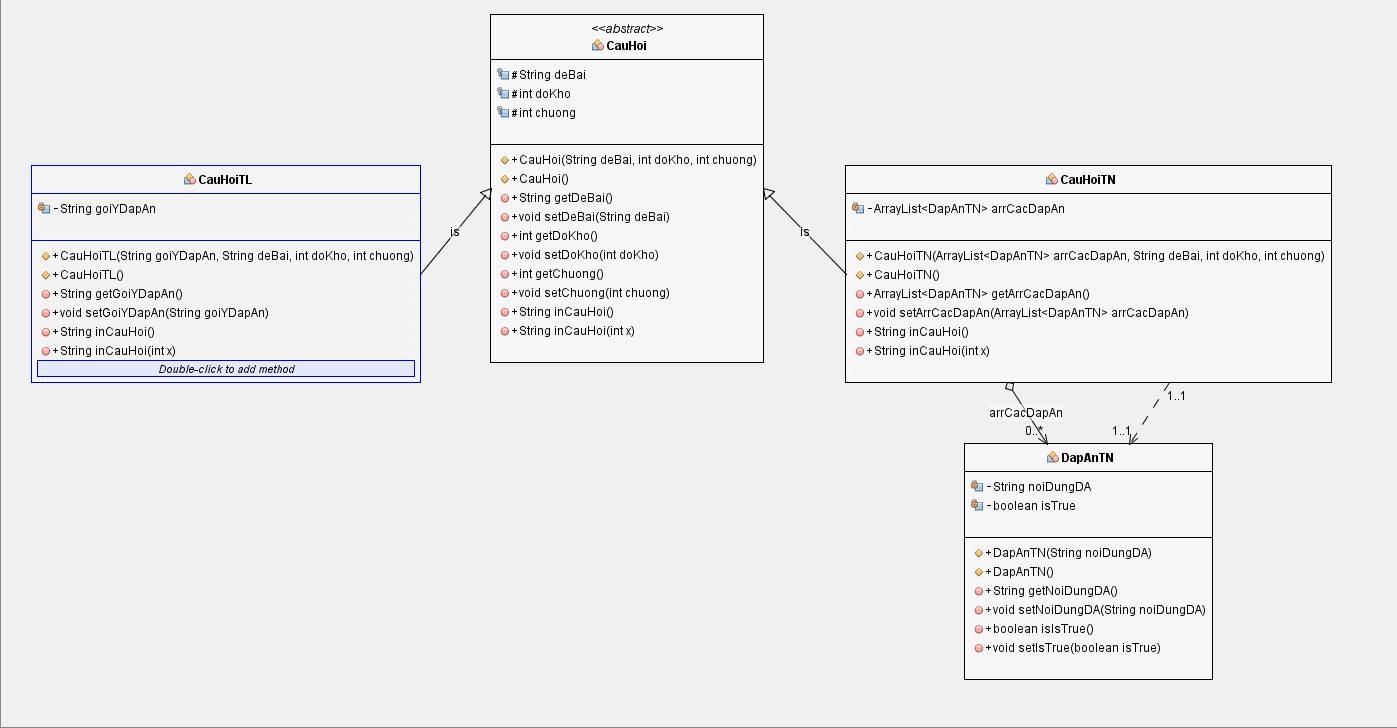
}

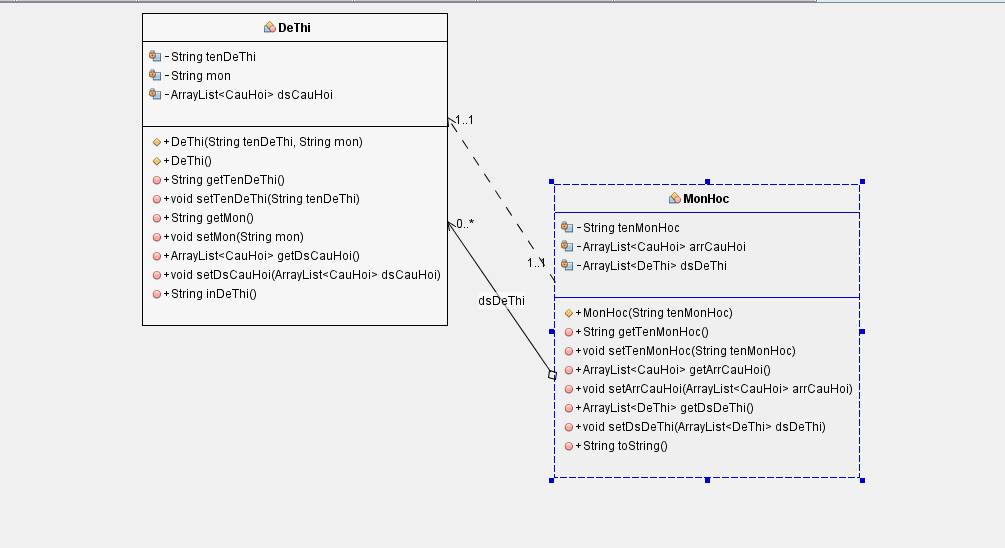
* Khai báo mảng ArrayList kiểu MonHoc , khởi tạo đối tượng mảng
* Hàm LuuDanhSachMonHocRaNgoai(ArrayList<MonHoc> dsMonHoc)

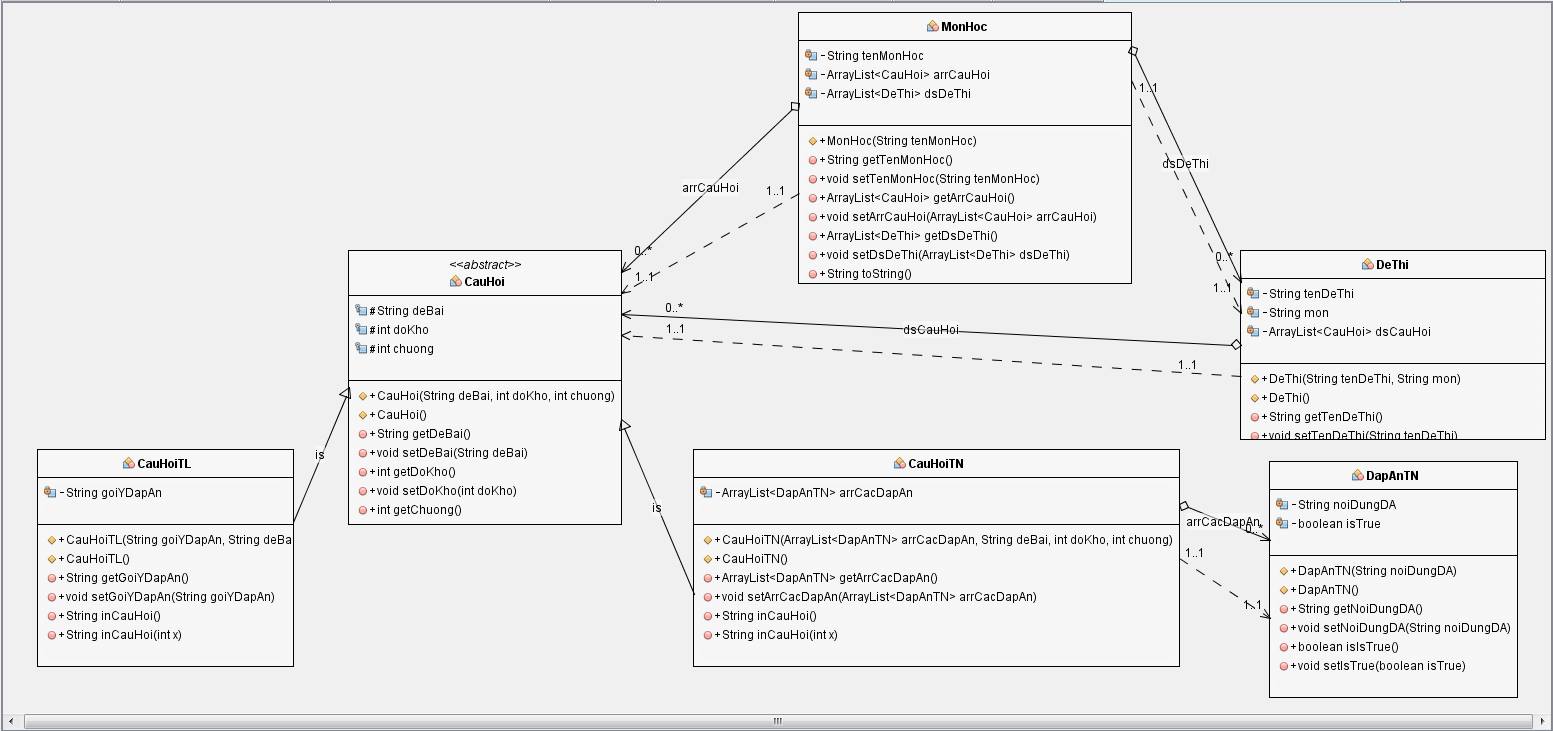
### 10. Các class giao diện

* Class GiaoDienChinh
* Class GiaoDienSoanCauHoi
* Class SoanDeThi
* Class GiaoDienXuatDeThi

## II. Biểu đồ và mối quan hệ các lớp





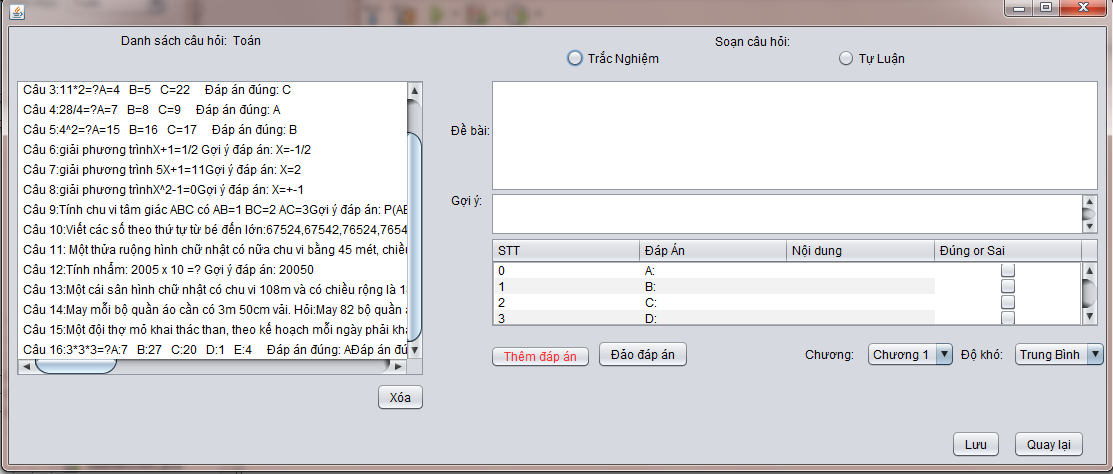


# Chương IV - Kết quả đạt được

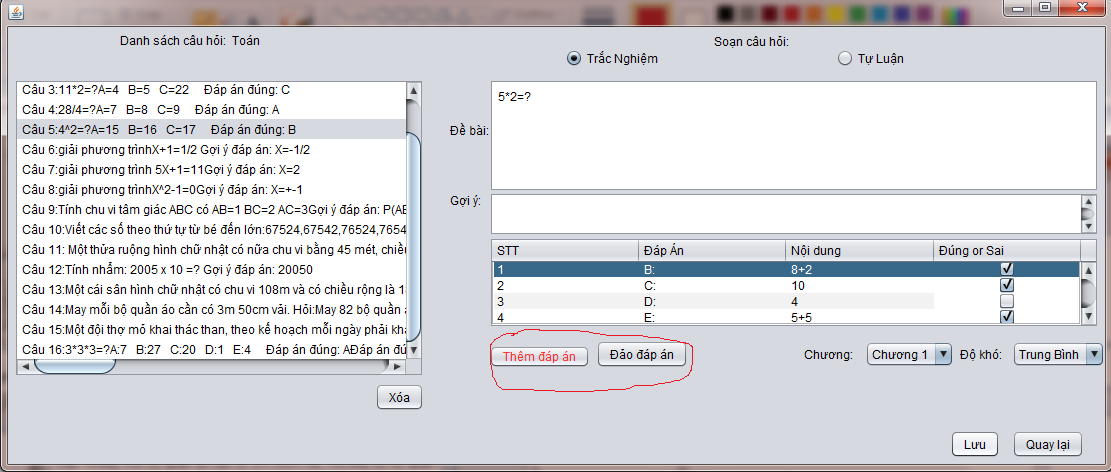
## I. Các chức năng đã xây dựng được

### 1. Giao diện soạn câu hỏi

* Tạo một ngân hàng câu hỏi bao gồm cả câu hỏi trắc nghiệm và câu hỏi tự luận

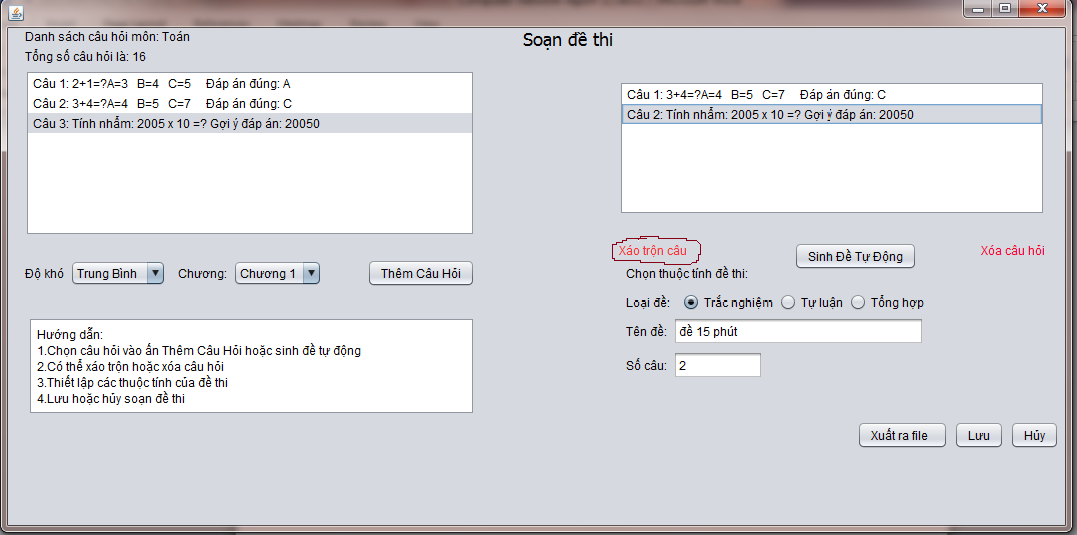


* Chức năng thêm một hay nhiều phương án cho câu hỏi, đảo đáp án lưa, xóa câu hỏi đã tạo và một câu hỏi trắc nghiệm có thể có nhiều đáp án đúng.

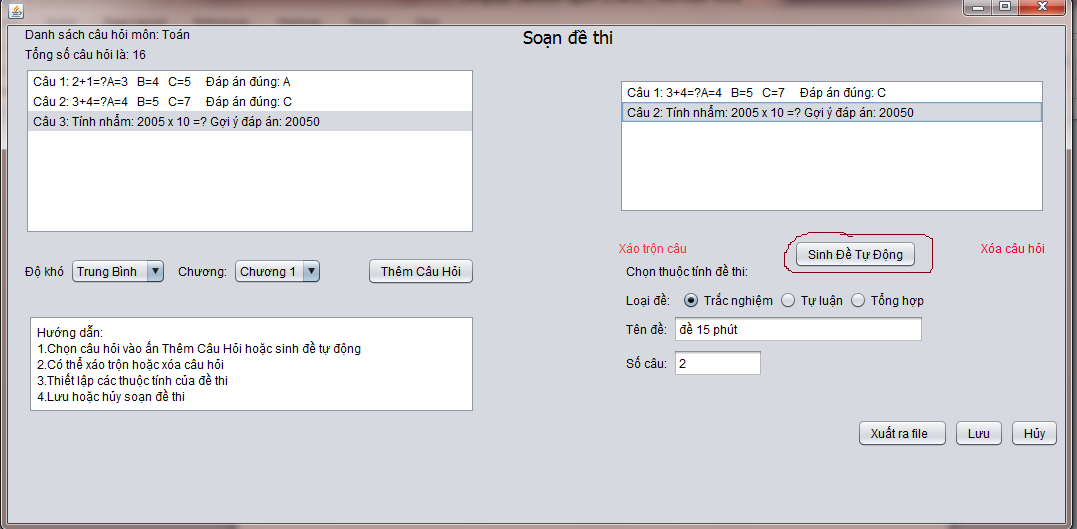


### 2. Giao diện soạn đề thi

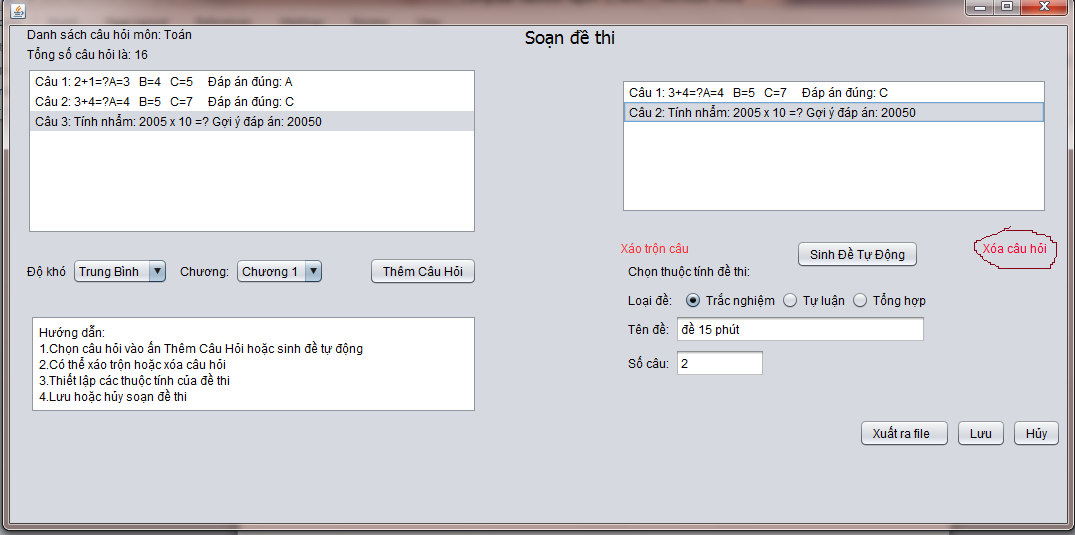
* Chức năng xáo trộn các câu hỏi đã chọn khi soạn đề thi



* Đề thi có thể tạo bằng tay hoặc sinh ngẫu nhiên.



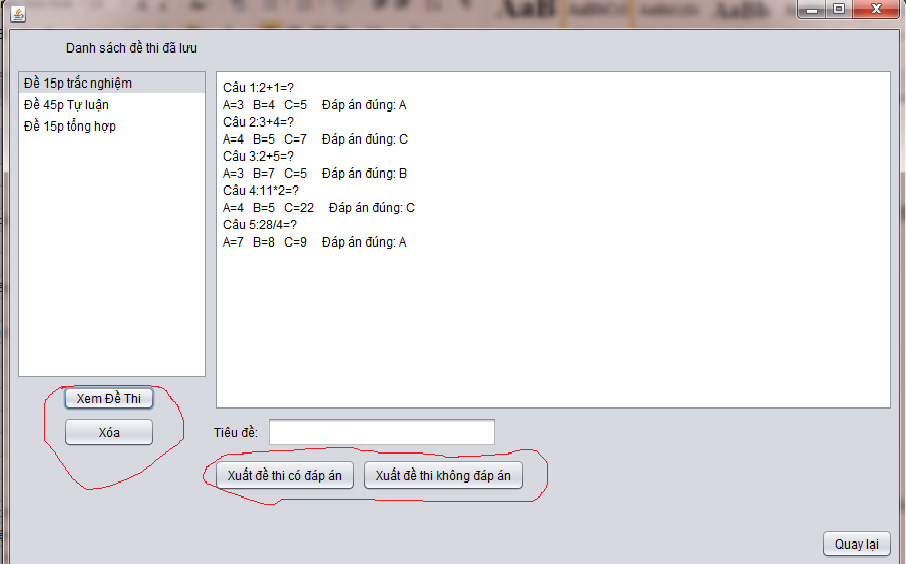
* Chức năng xóa câu hỏi đã chọn sau khi thêm câu hỏi hoặc sinh đề tự động



### 

### 3. Giao diện xuất đề thi

* Xem các đề thi đã lưu
* Xuất ra file theo mong muốn có thể là doc, txt…
* Xóa đề thi khỏi dánh sách



## II. Các hướng phát triển

* Phát triển thêm chức năng in ra các loại file như pdf…
* Phát triển giao diện thân thiện với người dùng hơn.
* Phát triển chức năng chỉnh sửa, thêm bớt câu hỏi trong đề thi đã lưu.