

Báo cáo môn Lập Trình Giao Diện

VIẾT CHƯƠNG TRÌNH THI TRẮC NGHIỆM

GIÁO VIÊN: NGUYỄN THỊ MAI TRANG

Nguyễn Sinh Hùng | Lập trình giao diện | 10/4/2023

Giới thiệu

Trắc nghiệm là phương pháp đánh giá năng lực hoặc kiến thức của một người thông qua việc trả lời các câu hỏi có sẵn. Trong báo cáo này, chúng ta sẽ tìm hiểu về cách viết chương trình thi trắc nghiệm thông qua ngôn ngữ lập trình C# Windows Form .NET

Mô tả chương trình

- Người dùng sẽ tạo bộ câu hỏi bằng file text (.txt) theo định dạng chương trình yêu cầu
- Người dùng sẽ lưu bộ câu hỏi vào chương trình và chương trình cho phép người dùng tham gia thi trắc nghiệm bằng các trả lời các câu hỏi trong bộ câu hỏi đã được chọn.
- Sau khi người dùng làm xong, chương trình hiển thị kết quả (số câu đúng, số câu sai, điểm) của bài thi trắc nghiệm

Các thành phần của chương trình

1. Giao diện người dùng

Khi chương trình được chạy, sẽ hiện ra bảng thí sinh sẽ nhập tên và thời gian thi vào (chương trình sẽ cảnh báo nếu thí sinh nhập sai). Sau đó, thí sinh nhấp vào nút **VÀO THI**.

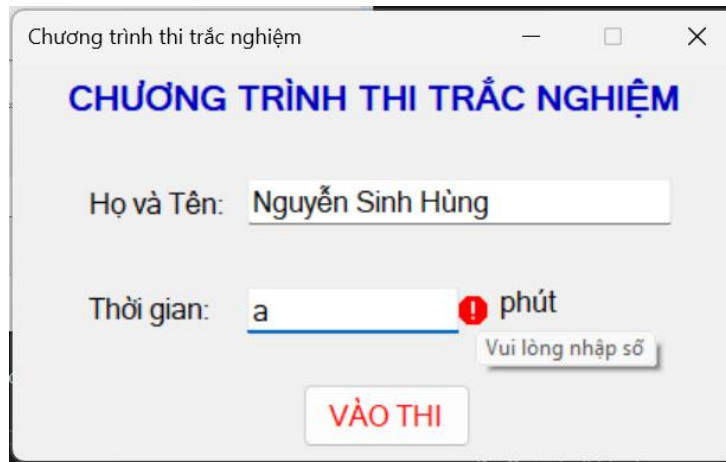
Chương trình thi trắc nghiệm

CHƯƠNG TRÌNH THI TRẮC NGHIỆM

Họ và Tên:

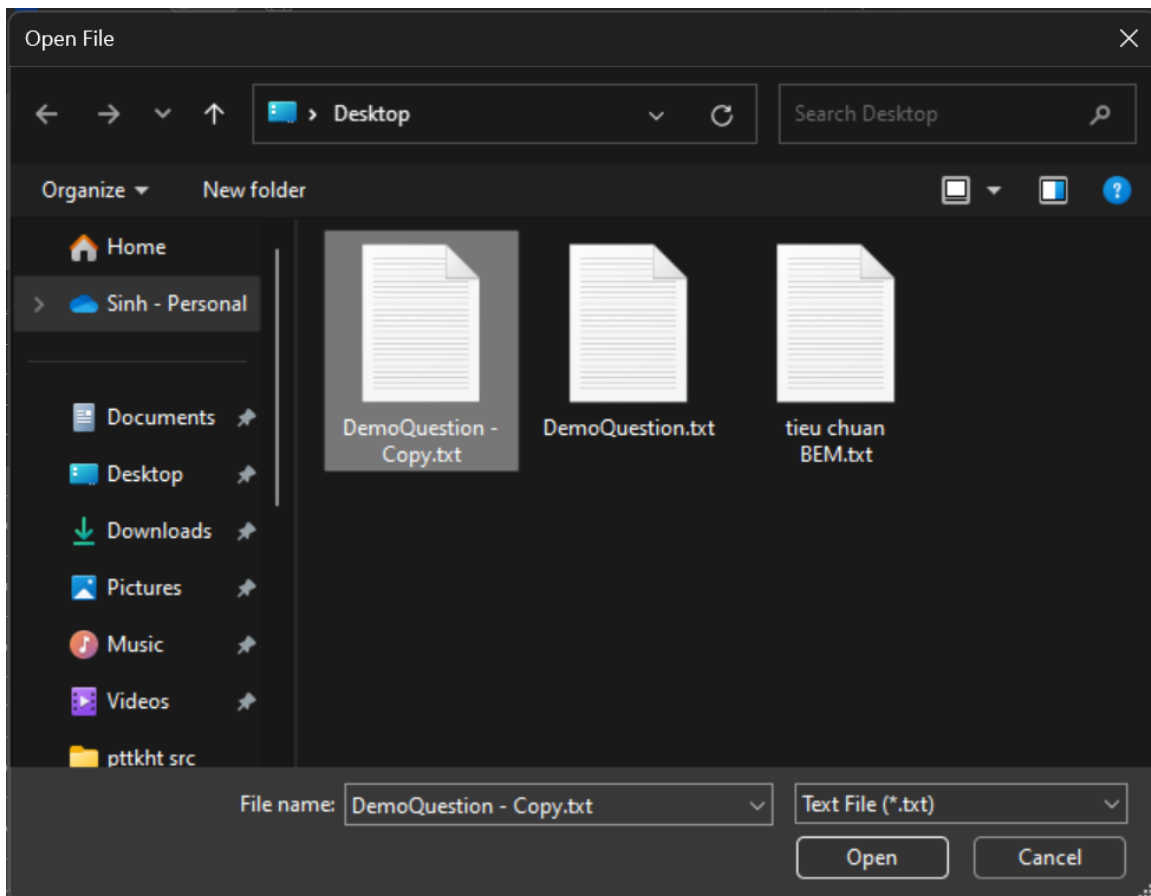
Thời gian: phút

VÀO THI



Chương trình cảnh báo nếu thí sinh nhập không đúng định danh số

Sau khi bấm nút **VÀO THI**, chương trình hiện ra cửa sổ để người dùng input file text của đề vào chương trình.



Sau đó người dùng chọn Open, chương trình hiện ra cửa sổ để thí sinh bắt đầu thực hiện bài thi trắc nghiệm.

Chương trình Thi trắc nghiệm

Thí sinh: Nguyễn Sinh Hùng
Thời gian: 01:29:53

Câu 1
1, Có bao nhiêu số có một chữ số

☐ a, 10 ☐ b, 9

☐ c, 8 ☒ d, 90

Câu trước Câu sau

NỘP BÀI

Sau khi thí sinh làm bài xong bấm vào nút **NỘP BÀI**, chương trình hiện ra MessageBox

Chương trình Thi trắc nghiệm

Thí sinh: Nguyễn Sinh Hùng
Thời gian: 01:29:54

Câu 1
1, Có bao nhiêu số có một chữ số

☒ a, 10 ☐ b, 9

☐ c, 8 ☐ d, 90

Câu trước Câu sau

NỘP BÀI

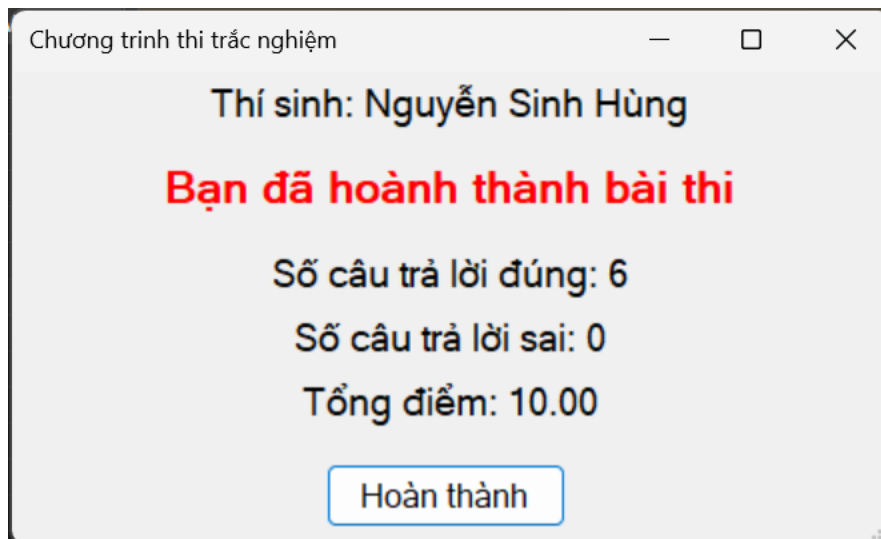
Chương trình trắc nghiệm

Bạn có chắc chắn muốn nộp bài thi không?

Yes No

Nếu có chọn Yes, chương trình tắt cửa sổ làm bài và hiển thị kết quả bài làm

Nếu không chọn No, thí sinh tiếp tục làm bài.



Tại kết quả làm bài thi, chương trình hiện số câu đúng, số câu sai và tính điểm của thí sinh trên thang điểm 10.

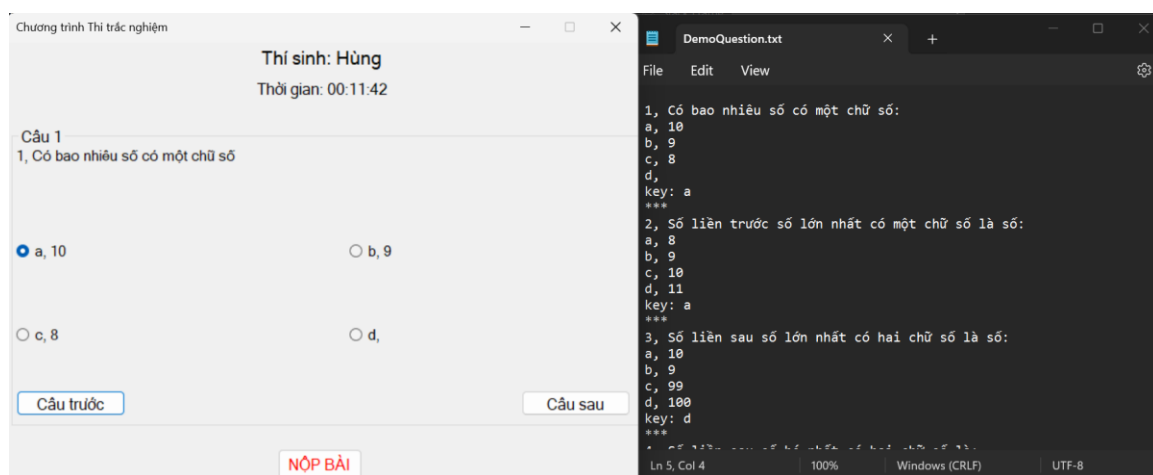
2. Các phương thức xử lý

Yêu cầu của chương trình

Chương trình yêu cầu nhập dữ liệu đúng định dạng file để có thể đọc được

Một câu trắc nghiệm cần phải có câu hỏi và các đáp án trả lời là a,b,c,d. Nếu câu hỏi khuyết đáp án (a,b,c không có d) thì cũng phải ghi d, và để trống

Ví dụ: câu 1 khuyết câu d



Và ở giữa mỗi câu cần phải có 3 dấu sao (***) và phải có đáp án của câu hỏi (key)

Phương thức xử lý dữ liệu đưa vào là file text (.txt)

Chương trình kiểm tra file có đuôi là .txt thì bắt đầu đọc file text và hiển thị câu hỏi đầu tiên. Việc thiết kế code thuận tiện cho việc mở rộng nếu đọc thêm file excel (.xlsx), word (.docx),...

```
private void readFile(string filepath)
{
    if(Path.GetExtension(filepath) == ".txt")
    {
        readFileTxt(filepath);
        loadQuestion(0);
    }
}
```

Thực hiện đọc file

```
private void readFileTxt(string filepath)
{
    string content = File.ReadAllText(filepath);
    string[] question = content.Trim().Split(new string[] { "****" }, StringSplitOptions.RemoveEmptyEntries);

    for(int i = 0; i < question.Length; i++)
    {
        string[] partsQues = question[i].Trim().Split(new string[] { ":", "\n" }, StringSplitOptions.RemoveEmptyEntries);
        Question quesItem = new Question(partsQues[0], partsQues[2], partsQues[3], partsQues[4], partsQues[5], (partsQues[7].Trim().ToCharArray())[0]);
        list.Add( quesItem );
    }
}
```

Chương trình đọc file.txt tất cả các dòng (phương thức ReadAllText) và sử dụng phương thức split("****") để cắt giữa các câu hỏi và lưu vào mảng question. Sau đó duyệt qua các thành phần của mảng và tiếp tục cắt dựa và các kí tự ":" và "\n" (xuống dòng

Chương trình có thiết kế một class đối tượng Question bao gồm các thuộc tính (id, câu hỏi, đáp án A, đáp án B, đáp án C, đáp án D và key). Chương trình khởi tạo *List<Question> list = new List<Question>()* để dễ quản lý đối tượng câu hỏi (Question).

Thời gian đếm ngược

Sau khi nhận thời gian làm bài thi của thí sinh (phút), chương trình sẽ chuyển đổi ra bao nhiêu giờ, bao nhiêu phút.

Ví dụ: 90 phút = 1 giờ 30 phút

Chương trình sử dụng bộ đếm thời gian timer để đếm ngược thời gian làm bài của thí sinh.

Sau khi hết thời gian làm bài thì chương trình tự động đóng cửa sổ làm bài thi của thí sinh và hiện ra kết quả làm bài thi của thí sinh.

Nút thực hiện Câu sau và Câu trước

Chương trình khởi tạo biến là `dem = 0`

Khi người dùng chọn **câu sau** thì kiểm tra `dem` có nhỏ hơn số câu không. Nếu có biến `dem` được tăng lên 1 (`dem++`).

Khi người dùng chọn **câu trước** thì kiểm tra `dem` có lớn hơn 0 (`dem > 0`) không. Nếu có biến `dem` được giảm đi 1 (`dem--`).

Sau đó thì load các câu hỏi của biên điểm.

Lưu vết các câu trả lời của thí sinh và tính kết quả

Chương trình tạo mảng rỗng số nguyên là `arrAns` khi form chạy. Sau khi `fileText` được đọc thì ta đã có số câu hỏi trong list (ở phần xử lý dữ liệu) thì khởi tạo mảng có kích thước bằng kích thước của list và cho tất cả các phần tử của mảng là -1.

Dựa vào biến `dem` sẽ biết người dùng đang ở câu nào và lưu câu trả lời của người dùng tại vị trí `dem--` của mảng `arrAns` những câu người dùng không chọn thì vẫn để giá trị là -1.

Quy đổi: A là 1 ; B là 2 ; C là 3 ; D là 4

Sử dụng sự kiện `CheckedChanged` cho 4 `radioButton` để kiểm tra nếu `radioBtn.Name` là radio A thì `arrAns[dem] = 1`, `radioBtn.Name` là radio B thì `arrAns[dem] = 2`. Tương tự với C và D. Với các câu không được chọn thì giá trị của phần tử tại vị trí `dem` không thay đổi (-1)

Ví dụ:

Câu 1 người dùng chọn câu B thì `arrAns[0] = 2`

Câu 2 người dùng bỏ qua thì `arrAns[1]` không thay đổi nên giá trị vẫn là -1

Câu 3 người dùng chọn D thì `arrAns[2] = 4`

Nếu người dùng quay trở lại câu trước hay câu tiếp mà chương vẫn giữ câu trả lời cũ của người dùng thì thực hiện kiểm tra ngay trước khi load các câu hỏi của các nút **Câu trước** và **Câu sau**.


```

private void btnNextQues_Click(object sender, EventArgs e)
{
    int n = list.Count;
    n = n - 1;
    if (dem < n)
    {
        dem++;
    }
    checkedAns(ref ansArr);
    loadQestion(dem);
}
//nút lùi câu hỏi
1 reference
private void btnPreQues_Click(object sender, EventArgs e)
{
    if (dem > 0)
    {
        dem--;
    }
    checkedAns(ref ansArr);
    loadQestion(dem);
}

```

Để kiểm tra ta truyền biến tham chiếu của mảng ansArr vào hàm và sử dụng vòng lặp để duyệt qua các câu hỏi. Nếu vị trí phần tử bằng với biến dem (số câu người dùng muốn đến) thì kiểm tra phần tử tại vị trí đó là 1 thì radioA.Checked = true , tương tự các đáp án B,C và D.

Sau khi thí sinh chắc chắn muốn nộp bài hoặc hết thời gian làm bài. Chương trình đóng cửa sổ làm bài. Sau đó, sẽ tạo ra biến lưu trữ số câu đúng (soCauDung) và khởi tạo mảng keyArr có kích thước bằng kích thước của list . Chương trình sẽ duyệt các phần tử của list qua các key của Question (được quản lí bởi list) tạo ra một mảng đáp án theo số và so sánh với mảng của arrAns (mảng câu trả lời của thí sinh) nếu phần tử keyArr có giá trị bằng với phần tử ansArr thì số câu đúng tăng 1. Từ đó tính ra điểm của thí sinh.

HẾT