|  |  |
| --- | --- |
|  | **Java Core Final Exam Test duration**: 180 minutes |

***Thí sinh không được sử dụng mạng LAN, internet, USB Flash, Bluetooth.***

***Không được sao chép đề thi dưới mọi hình thức.***

**Họ tên: Lớp:**

1. **Mô tả nghiệp vụ**:

Một ứng dụng quản lý thông tin hành khách của một hãng hàng không có nghiệp vụ như sau:

Hành khách đi trên các chuyến bay được chia làm 3 loại:

1. - Hành khách thông thường (Adult)
2. - Trẻ em (Children)
3. - Người mang thai (Pregnant)

Đối với tất cả các những hành khách đều có các thuộc tính chung gồm: Mã hành khách (PassengerID), Họ tên (Name), Ngày sinh (BirthDate), Địa chỉ (Address), Số điện thoại (Phone), Ngày test Covid gần nhất (TestDate), Ngày khởi hành (FlightDate)

* Đối với **Hành khách thông thường** thì có thêm thông tin: Nghề nghiệp (Job)
* Đối với **Trẻ em** thì có thêm thông tin: Trường (School), Lớp (Grade)
* Đối với **Người mang thai** thì có thêm thông tin: Tuổi thai (GestationalAge), Tiền sử sinh non (PretermBirth)

Mỗi hành khách có 1 phương thức ShowInfo để biểu diễn thông tin bao gồm PassengerID, Name, BirthDate, Address, Phone, TestDate, FlightDate. Với **Hành khách thông thường** thì thêm thông tin Job; với **Trẻ em** thì thêm thông tin School, Grade; với **Người mang thai** thì thêm thông tin GestationalAge, PretermBirth.

1. **Yêu cầu của thí sinh:**
2. **Yêu cầu 1 (2.5 điểm):**

Thí sinh hãy thiết kế và viết code thể hiện đầy đủ các class, thuộc tính, phương thức của các **Passenger** sao để tuân thủ theo mô hình **OOP** đã học, áp dụng 4 tính

chất : bao đóng (*encapsulation)* , kế thừa (*inheritance)* , đa hình (*polymorphism)* , trừu tượng (*abstraction).*

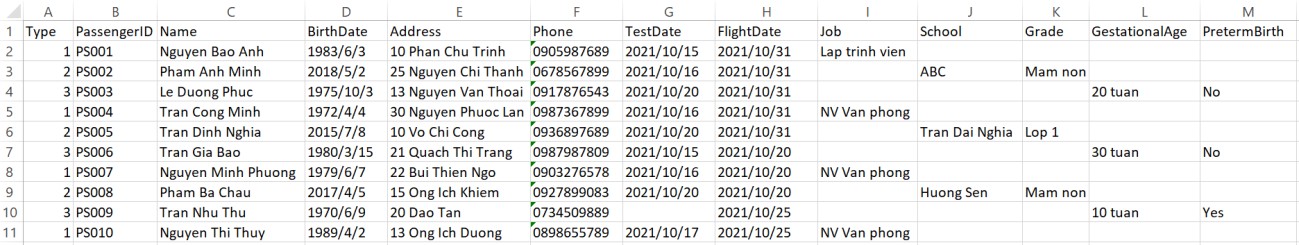
*\*\*\* Lưu ý:*

* 1. *Hãy tạo java doc (document comment) ở tất cả các phương thức được viết ra, trừ hàm getter, setter.*

*Nội dung java doc phải có : Account người tạo, ngày tháng năm sinh người tạo (định dạng YYYY\_MM\_DD), các nội dung khác thí sinh tự viết*

1. **Yêu cầu 2 (3.0 điểm)**: Chức năng Input Data

Thí sinh hãy nhập giá trị của các thuộc tính của các đối tượng trên có chứa data của hơn 10 hành khách (Passenger), yêu cầu phải có đầy đủ cả 3 loại đối tượng mô tả ở trên.



**điểm)**

* 1. Insert tất cả các đối tượng trên vào các List**(1.0**

Lưu ý: Dòng nào xảy ra exception thì không

insert vào list, những dòng còn lại insert bình thường.

* 1. Validate data trước khi insert vào Database:
     + Nếu Mã hành khách (PassengerID) không có định dạng PSxxx thì thông báo ra màn hình console nội dung “PassengerID must be in format of “PSxxx”!!!” **(0.5 điểm).**
     + Nếu Ngày xét nghiệm Covid gần nhất (TestDate) lớn hơn ngày hiện tại thì hãy catch exception và thông báo ra màn hình console nội dung “Test Date of Covid must be less than current date**!!!” (0.5 điểm).**
     + PassengerID không được trùng, nếu hành khách có PassengerID trùng với PassengerID đã tồn tại trước đó thì bỏ qua, không insert vào Database và thông báo ra màn hình console nội dung: “Passenger ” + PassengerID + “has duplicate ID”. **(0.5 điểm)**
     + Nếu có bất cứ exception nào khác trong quá trình chạy chương trình thì hãy thông báo ra màn hình console nội dung “Program have an unexpected error occurred !!!” **(0.5 điểm)**

1. **Yêu Cầu 3 (4.5 điểm)**: Thực hiện 2 chức năng sau:
   1. **Chức năng 1 (1 điểm)**: Giới thiệu thông tin của các hành khách hiện có.

Bạn hãy viết phương thức **ShowInformation**, có tham số là một ArrayList gồm danh sách các hành khách (Passenger).

Trong phương thức này hãy duyệt ArrayList những hành khách (Passenger) và cho từng đối tượng có thể hiển thị thông tin của mình (dùng phương thức showInfo đã nói ở trên)

Gọi phương thức **ShowInformation** để thực hiện cho list đối tượng đã nhập ở

trên.

* 1. **Chức năng 2 (3 điểm)**: Cập nhật ngày khởi hành.

Trong trường hợp hành khách có ngày test Covid quá 1 tuần so với ngày khởi

hành thì cần cập nhật lại ngày khởi hành (FlightDate) lùi lại 2 ngày (**1 điểm**)

Ngoài ra với tất cả các hành khách là người thân (ở cùng địa chỉ với các hành khách trên) thì cũng cập nhật Ngày khởi hành (FlightDate) lùi lại 2 ngày. (**1 điểm**)

Hiển thị toàn bộ danh sách những hành khách (Passenger) có đầy đủ các thuộc tính từ Database với thông tin vừa được cập nhật ra màn hình console, yêu cầu sắp xếp giảm dần theo **FlightDate** và tăng dần theo thuộc tính **PassengerID**. **(1 điểm)**

*Lưu ý:* Yêu cầu sử dụng **Collection.Sort** để thực hiện yêu cầu sắp xếp trên.

* 1. **Chức năng 3 (0.5 điểm)**: Xóa những hành khách chưa tiến hành test covid (TestDate có giá trị null).

1. **Yêu cầu 4 (điểm cộng: 0.5 điểm):**

Thí sinh tuân thủ được đầy đủ coding convention của Java đã học, code rõ ràng, sáng sủa thì được cộng 0.5 điểm.

\* Điểm tối đa: 10 điểm**.**

**- THE END -**