Trường Đại học Khoa học Tự nhiên

Khoa Toán – Cơ – Tin học

Học kỳ I – 2022 - 2023

Thi cuối kỳ: môn **Các Thành phần Phần mềm**

Thời gian làm bài: 150 phút

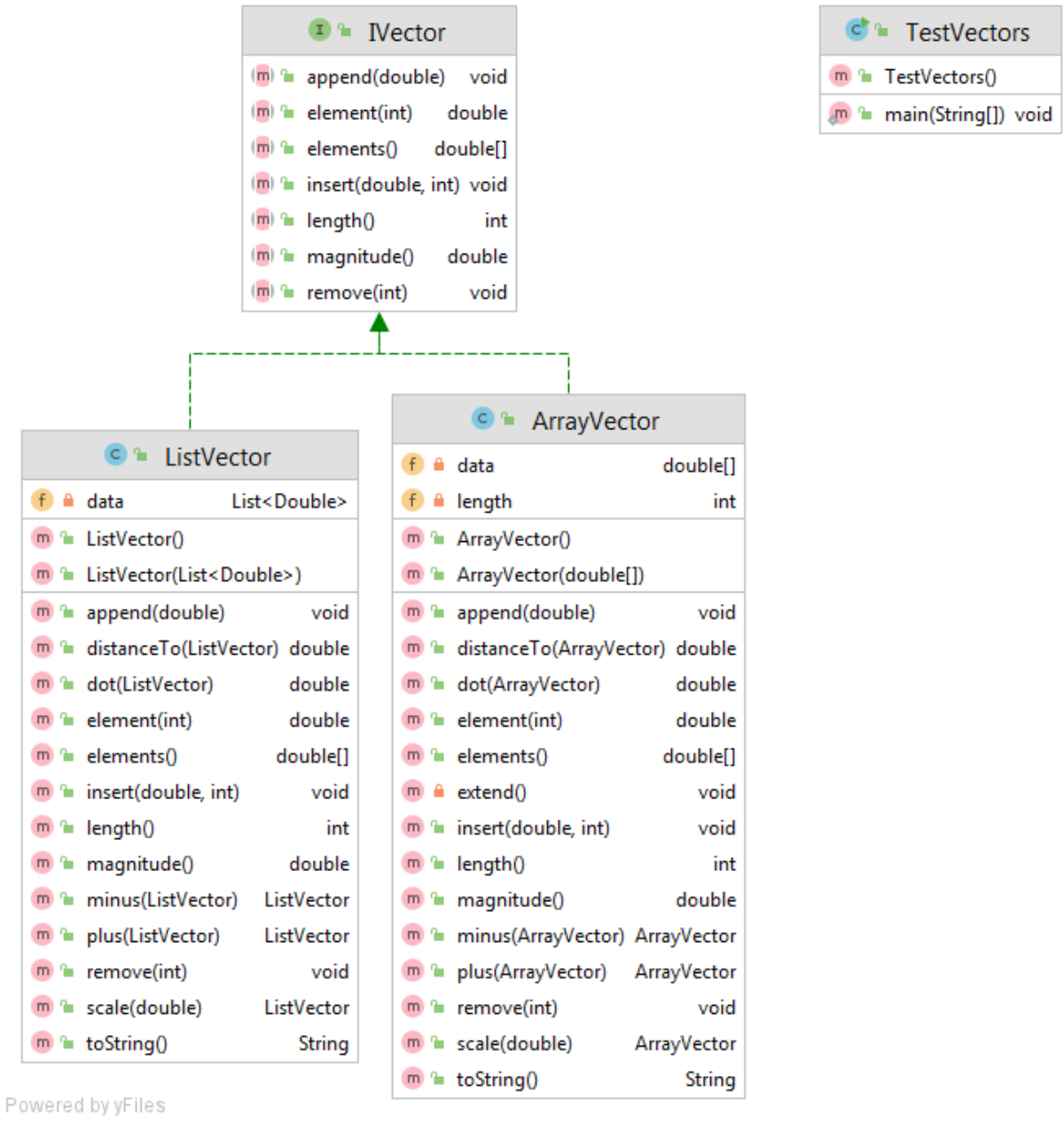
Đề 1

Bài 1 (3 điểm). Cho ví dụ minh họa và giải thích bốn nguyên lý chính của Lập trình Hướng đối tượng. Thực hiện theo hướng dẫn sau:

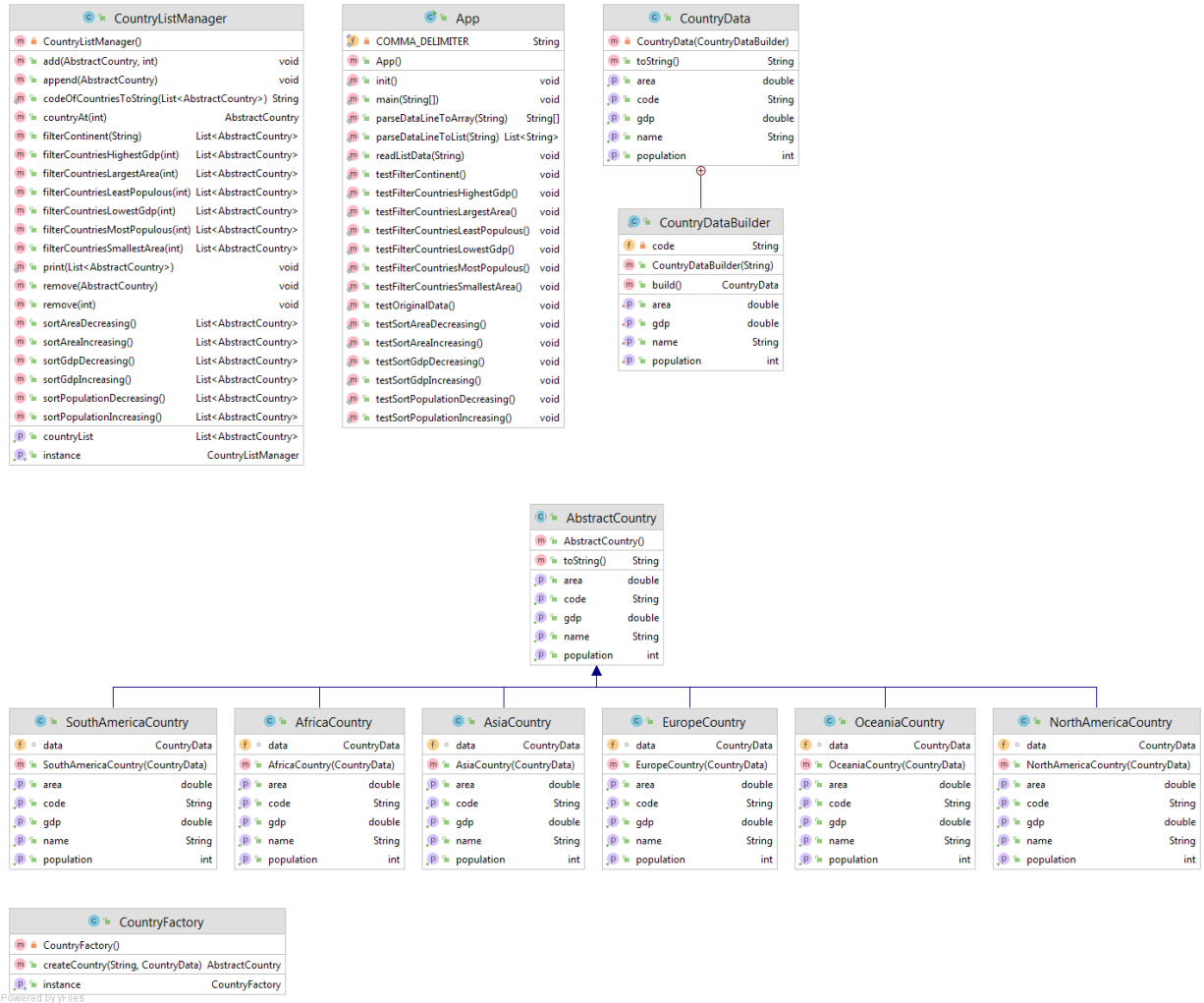
- Tạo một project có **tên OOPPrinciples<Mã sinh viên>** (Ví dụ OOPPrinciples1234).
- Tạo package có tên **oopprinciples.encapsulation**, trong package cho ví dụ về tính đóng gói.
- Tạo package có tên **oopprinciples.inheritance**, trong package cho ví dụ về tính thừa kế.
- Tạo package có tên **oopprinciples.polymorphism**, trong package cho ví dụ về tính đa hình.
- Tạo package có tên **oopprinciples.abstraction**, trong package cho ví dụ về tính trừu tượng.
- Trong mỗi package, viết file client driver để chạy demo chương trình. Trong file này, viết comment giải thích trong code chỗ nào thể hiện những nguyên lý chính của Lập trình Hướng đối tượng.

Bài 2 (2 điểm). Ta có thể thiết kế vector sử dụng cấu trúc dữ liệu kiểu mảng hoặc kiểu List. Chương trình được thiết kế như biểu đồ UML được cho dưới đây.

- Hoàn thiện code cho trong các file source code được cung cấp.
- Viết code để chạy demo chương trình. Lưu kết quả chạy chương trình vào file text có tên Vector<Mã sinh viên>.txt và nộp cùng source code.



Bài 3 (3 điểm). File countries.csv lưu dữ liệu về tên nước, dân số, diện tích, gdp, khu vực của một số nước trên thế giới. Chương trình quản lý thông tin các nước từ dữ liệu trong file countries.csv, được thiết kế như biểu đồ UML dưới đây:



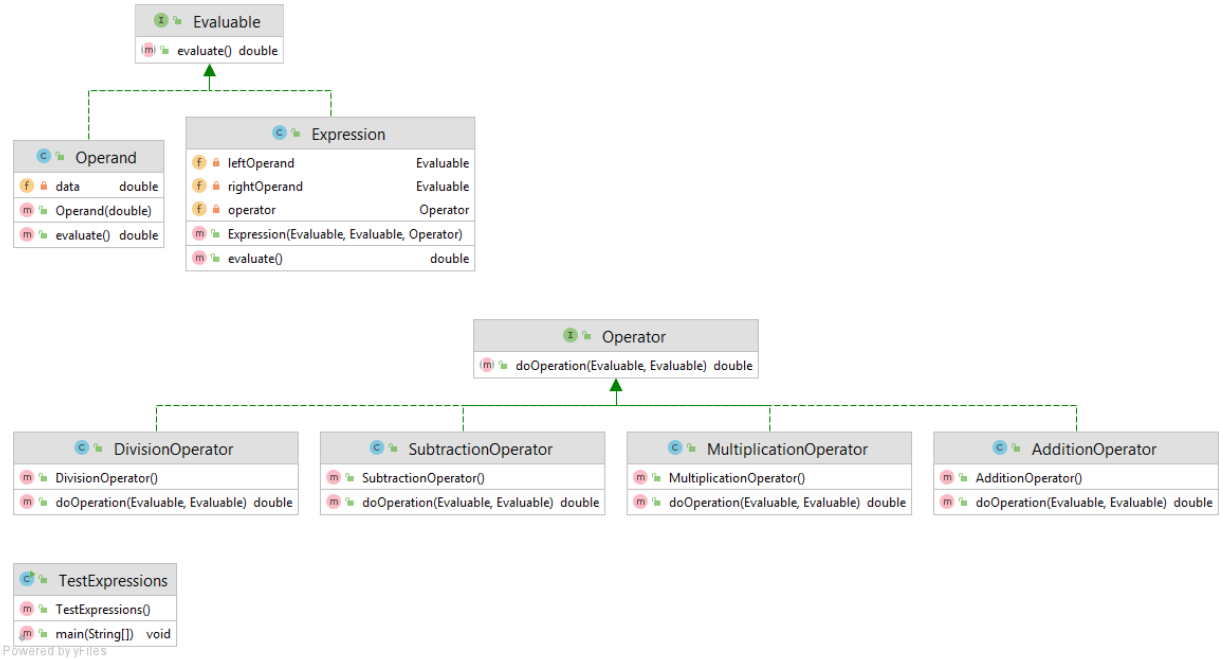
- Lớp **CountryData** chứa dữ liệu một nước, tương ứng với một dòng dữ liệu được đọc vào từ file countries.csv. Lớp **CountryData** được thiết kế dùng Builder Pattern, nên chú ý dùng Builder Pattern để tạo đối tượng loại **CountryData**.
- Lớp **AbstractCountry** là lớp trừu tượng, chứa những đặc điểm chung của các nước.
- Các lớp **AsiaCountry**, **AfricaCountry**, **NorthAmericaCountry**, **SouthCountry**, **EuropeCountry**, **OceaniaCountry** là một lớp chứa đầy đủ thông tin chi tiết về một nước, được thừa kế từ lớp **AbstractCountry**.
- Lớp CountryFactory là lớp được thiết kế theo Singleton Pattern và Factory Pattern, dùng để tạo các đối tượng country, như **AsiaCountry**, **EuropeCountry**, …

- Lớp **CountryListManager** là lớp được thiết thế theo Singleton Pattern để quản lý dữ liệu các nước. Lớp **CountryListManager** sử dụng cấu trúc dữ liệu kiểu **List** để quản lý dữ liệu.
- Lớp **App** là lớp client driver để thực thi chương trình.

Viết code để hoàn thiện chương trình theo thiết kế đã cho trong các file source code đi kèm để thực hiện:

- Đọc dữ liệu vào từ file countries.csv.
- Sử dụng dữ liệu được đọc từ file, tạo các đối tượng **CountryData**, và tạo các đối tượng kiểu **AsiaCountry**, **AfricaCountry**, **NorthAmericaCountry**, **SouthCountry**, **EuropeCountry**, **OceaniaCountry** sử dụng **CountryFactory**. Sau đó đưa các đối tượng vừa tạo vào **CountryListManager** để quản lý.
- Viết code để hoàn thiện các chức năng quản lý trong **CountryListManager**.
- Viết code để test chương trình trong các hàm test đã cho trong file client driver **App**.java. Lưu kết quả chạy chương trình vào file CountryListManager<Mã sinh viên>.txt.

Bài 4 (2 điểm). Chương trình dưới đây sử dụng Composite Pattern và Strategy Pattern để biểu diễn và tính giá trị của một biểu thức toán học. Hãy hoàn thiện chương trình theo biểu đồ UML đã cho và viết code chạy demo chương trình.



- Hoàn thiện code cho trong các file source code được cung cấp.

- Thực hiện các yêu cầu trong file **TestExpressions**.java. Lưu kết quả chạy chương trình vào file đặt tên CompositeStratery<Mã sinh viên>.txt.

Chú ý:

- Sinh viên được sử dụng tài liệu.
- Sinh viên được viết thêm các file, các method vào chương trình mẫu.
- Sau khi hoàn thiện chương trình, nộp lại file nén các file cource code và file text kết quả chạy chương trình.
- Bài nộp không có file text kết quả chạy chương trình được xem là chương trình chưa chạy được.
- Những bài bị phát hiện có gian lận sẽ được điểm 0.
