

Đề 1

Bài 1 (2 điểm). Để tính gần đúng tích phân xác định, người ta có thể sử dụng công thức hình thang được mô tả như sau:

Công thức hình thang (trapezoidal rule): Tích phân xác định có thể được tính xấp xỉ sử dụng công thức hình thang như sau:

$$\int_a^b f(x) dx \approx h \left[\frac{f(x_0) + f(x_1)}{2} + \frac{f(x_1) + f(x_2)}{2} + \dots + \frac{f(x_{n-1}) + f(x_n)}{2} \right]$$
$$= h \left[\frac{f(x_0) + f(x_n)}{2} + f(x_1) + \dots + f(x_{n-1}) \right] = l_n,$$

trong đó $h = \frac{b-a}{n}$, $x_i = a + ih$.

Dùng công thức hình thang, tính gần đúng tích phân xác định sau với độ chính xác 10^{-6} :

$$\int_0^{\pi/2} \sin^2(x) \cos(x) dx,$$

Kiểm chứng với cách tính trực tiếp:

$$\int_0^{\pi/2} \sin^2(x) \cos(x) dx = \frac{\sin^3(x)}{3} \Big|_0^{\pi/2} = \frac{1}{3} \left[\sin^3\left(\frac{\pi}{2}\right) - \sin^3(0) \right] = \frac{1}{3} \sin^3\left(\frac{\pi}{2}\right).$$

Độ chính xác ở đây được xác định như sau, chọn $n_0 = 2$, sau đó tính l_n với $n = n_0, 2n_0, 4n_0, \dots$. Việc tính toán dừng lại khi $|l_{2n} - l_n| < \epsilon = 10^{-6}$.

Ví dụ:

Trapezoidal Rule: 0.333333

Java Library: 0.333333

Bài 2 (3 điểm). File countries1.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 countries1.csv, được thiết kế như sau:

CountryData		
f	code	String
f	name	String
f	population	int
f	area	double
f	gdp	double
f	continent	String
m	CountryData(String)	
m	getArea()	double
m	getCode()	String
m	getContinent()	String
m	getGdp()	double
m	getName()	String
m	getPopulation()	int
m	setArea(double)	CountryData
m	setContinent(String)	CountryData
m	setGdp(double)	CountryData
m	setName(String)	CountryData
m	setPopulation(int)	CountryData
m	toString()	String

CountryArrayManager		
f	countryDataArray	CountryData[]
f	length	int
f	increment	int
m	CountryArrayManager()	
m	CountryArrayManager(int)	
m	add(CountryData, int)	void
m	allocateMore()	void
m	append(CountryData)	void
m	codeOfCountriesToString(CountryData[], int)	String
m	correct()	void
m	countryDataAt(int)	CountryData
m	filterContinent(String)	CountryData[]
m	filterHighestGdpCountries(int)	CountryData[]
m	filterLargestAreaCountries(int)	CountryData[]
m	filterLeastPopulousCountries(int)	CountryData[]
m	filterLowestGdpCountries(int)	CountryData[]
m	filterMostPopulousCountries(int)	CountryData[]
m	filterSmallestAreaCountries(int)	CountryData[]
m	getCountryDataArray()	CountryData[]
m	getLength()	int
m	print(CountryData[], int)	void
m	remove(int)	void
m	setLength(int)	void
m	sortDecreasingByArea()	CountryData[]
m	sortDecreasingByGdp()	CountryData[]
m	sortDecreasingByPopulation()	CountryData[]
m	sortIncreasingByArea()	CountryData[]
m	sortIncreasingByGdp()	CountryData[]
m	sortIncreasingByPopulation()	CountryData[]

App		
f	COMMA_DELIMITER	String
f	countryArrayManager	CountryArrayManager
m	App()	
m	init()	void
m	main(String[])	void
m	parseDataLineToArray(String)	String[]
m	parseDataLineToList(String)	List<String>
m	readArrayData(String)	void
m	testFilterContinent()	void
m	testFilterHighestGdpCountries()	void
m	testFilterLargestAreaCountries()	void
m	testFilterLeastPopulousCountries()	void
m	testFilterLowestGdpCountries()	void
m	testFilterMostPopulousCountries()	void
m	testFilterSmallestAreaCountries()	void
m	testOriginalData()	void
m	testSortDecreasingByArea()	void
m	testSortDecreasingByGdp()	void
m	testSortDecreasingByPopulation()	void
m	testSortIncreasingByArea()	void
m	testSortIncreasingByGdp()	void
m	testSortIncreasingByPopulation()	void

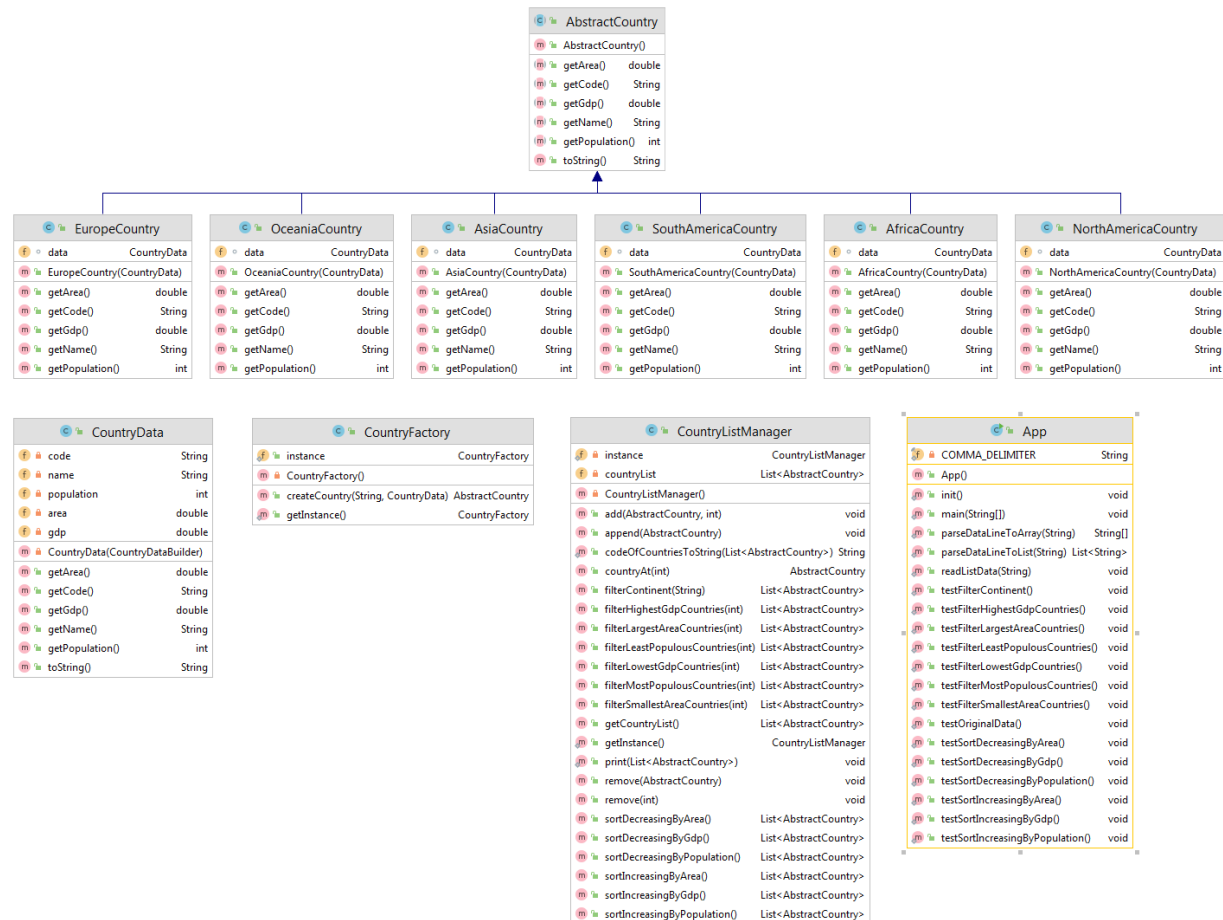
Powered by yFiles

- Lớp **CountryData** chứa dữ liệu một nước, tương ứng với một dòng dữ liệu được đọc vào từ file countries1.csv.
- Lớp **CountryArrayManager** là lớp để quản lý dữ liệu các nước (được lưu trong **CountryData**). Lớp **CountryArrayManager** sử dụng cấu trúc dữ liệu kiểu mảng 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 countries2.csv.
- Sử dụng dữ liệu được đọc từ file, tạo các đối tượng **CountryData**, sau đó đưa các đối tượng này vào **CountryArrayManager** để quản lý.
- Viết code để hoàn thiện các chức năng quản lý trong **CountryArrayManager**.
- 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 **CountryArrayManager<Mã sinh viên>.txt**.

Bài 3 (3 điểm). File countries2.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 countries2.csv, được thiết kế như sau:

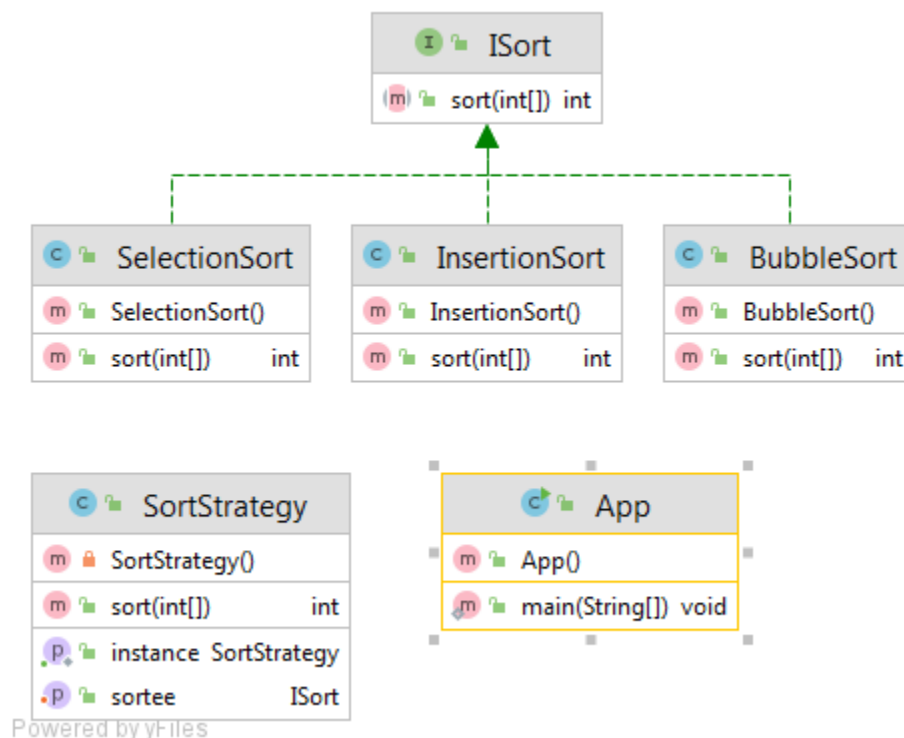


- Lớp **CountryData** chứa dữ liệu một nước, tương ứng với một dòng dữ liệu được đọc vào từ file countries2.csv. Lớp **CountryData** được thiết kế dùng Builder Pattern, nên chú ý dùng Builder Pattern để tạo đối tượng loại **CountryData**.
- Lớp **AbstractCountry** là lớp trừu tượng, chứa những đặc điểm 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 kế 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 countries2.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 kiểm tra và so sánh việc sắp xếp dữ liệu sử dụng một số thuật toán sắp xếp đơn giản, selection sort, bubble sort, insertion sort, được thiết kế như sau:



- Hoàn thiện code cho trong các file source code được cung cấp.
- Thực hiện các yêu cầu trong file **App.java**. Lưu kết quả chạy chương trình vào file đặt tên SortStratery<Mã sinh viên>.txt.

Chú ý: Sau khi hoàn thiện chương trình, nộp lại file nén các file course code và file text kết quả chạy chương trình.
