

Đề 2

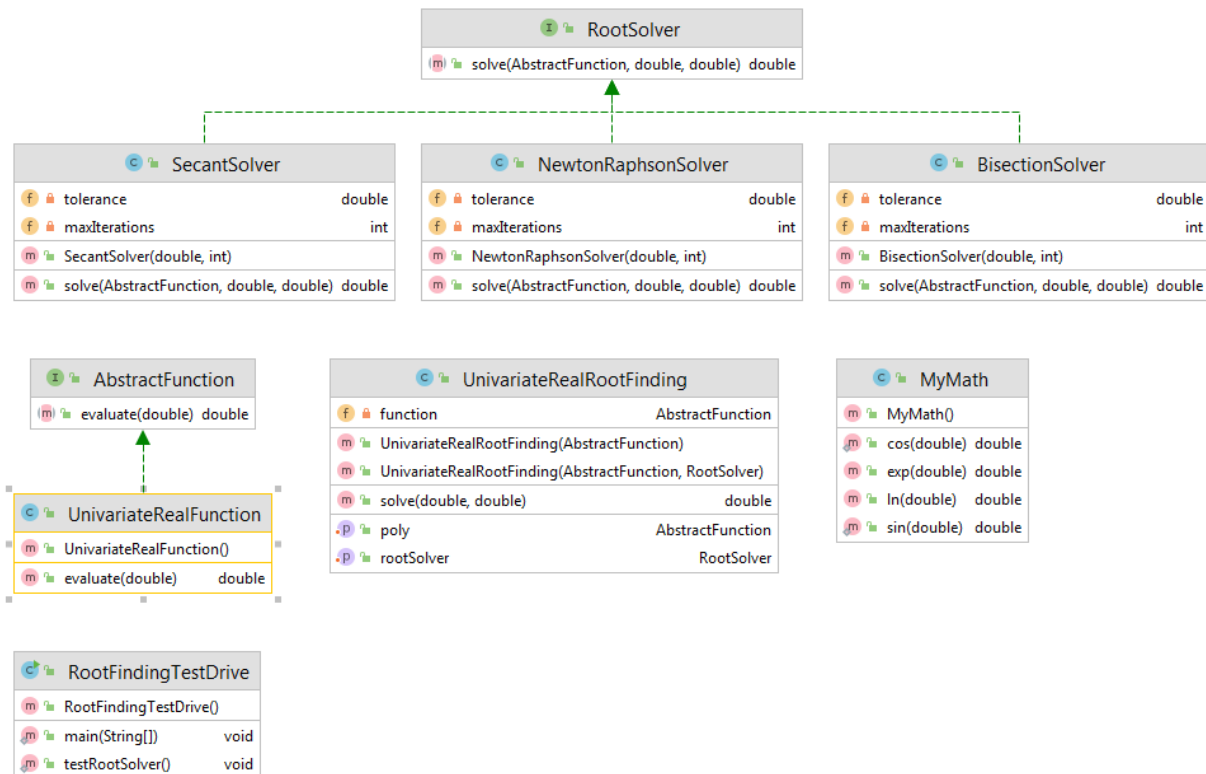
Bài 1 (3 điểm). Viết chương trình thư viện tính xấp xỉ các hàm sau đã cho theo khai triển Taylor trong lớp **MyMath**:

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots$$

$$\exp(x) = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots$$

$$\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots$$



Sau đó hoàn thiện chương trình theo code mẫu đã cho và thực hiện các yêu cầu tìm nghiệm của một số hàm theo các phương pháp tìm nghiệm đã cho:

- Tìm nghiệm của hàm $x.\sin(x) - 3$ trong đoạn $[12, 14]$.
- Lấy ví dụ một hàm nào đó có nghiệm trong một đoạn $[a, b]$ nào đó. Sử dụng các phương pháp tìm nghiệm đã cho thực hiện việc tìm nghiệm của hàm trong đoạn $[a, b]$. Các phương pháp tìm nghiệm khác nhau được thay đổi trong thời gian chạy. In ra các thông tin: phương pháp sử dụng, hàm cần tìm nghiệm, khoảng tìm nghiệm, và giá trị nghiệm tìm được.
- Lưu kết quả chạy chương trình vào file `<Ten_MaSinhVien_RootFinding>.txt` (ví dụ, `NguyenVanA_123456_RootFinding.txt`).
- Nén các file source code và file chạy kết quả theo định dạng zip, đặt tên `<Ten_MaSinhVien_RootFinding>.zip` (ví dụ, `NguyenVanA_123456_RootFinding.zip`), nộp lên classroom.

Chú ý: Các phương pháp tìm nghiệm của hàm một biến có thể tham khảo tại:

- https://en.wikipedia.org/wiki/Bisection_method
- https://en.wikipedia.org/wiki/Newton%27s_method
- https://en.wikipedia.org/wiki/Secant_method

Bài 2 (3 điểm). Viết chương trình Java được thiết kế như sơ đồ bên dưới.

- Hoàn thiện code cho trong các file source code mẫu được cung cấp.
- Thực hiện tất cả các phương thức test trong lớp client **App**.
- Lưu kết quả chạy chương trình vào file text có tên `<Ten_MaSinhVien_StudentManager>.txt` (ví dụ, `NguyenVanA_123456_StudentManager.txt`).
- Nén tất cả các file source code và file kết quả chạy chương trình theo định dạng zip và đặt tên `<Ten_MaSinhVien_StudentManager>.zip` (ví dụ, `NguyenVanA_123456_StudentManager.zip`), nộp lên classroom.

StudentManager	App	Student
<div>StudentManager()</div> <div>add(Student, int) void</div> <div>append(Student) void</div> <div>filterStudentsHigherThanAverageGrade(int) List<Student></div> <div>filterStudentsHigherThanChemistryGrade(int) List<Student></div> <div>filterStudentsHigherThanMathsGrade(int) List<Student></div> <div>filterStudentsHigherThanPhysicsGrade(double) List<Student></div> <div>filterStudentsHighestAverageGrade(int) List<Student></div> <div>filterStudentsHighestChemistryGrade(int) List<Student></div> <div>filterStudentsHighestMathsGrade(int) List<Student></div> <div>filterStudentsHighestPhysicsGrade(int) List<Student></div> <div>filterStudentsLowerThanAverageGrade(int) List<Student></div> <div>filterStudentsLowerThanChemistryGrade(int) List<Student></div> <div>filterStudentsLowerThanMathsGrade(double) List<Student></div> <div>filterStudentsLowerThanPhysicsGrade(double) List<Student></div> <div>filterStudentsLowestAverageGrade(int) List<Student></div> <div>filterStudentsLowestChemistryGrade(int) List<Student></div> <div>filterStudentsLowestMathsGrade(int) List<Student></div> <div>filterStudentsLowestPhysicsGrade(int) List<Student></div> <div>idOfStudentsToString(List<Student>) String</div> <div>print(List<Student>) void</div> <div>remove(Student) void</div> <div>remove(int) void</div> <div>sortAverageGradeDecreasing() List<Student></div> <div>sortAverageGradeIncreasing() List<Student></div> <div>sortChemistryGradeDecreasing() List<Student></div> <div>sortChemistryGradeIncreasing() List<Student></div> <div>sortMathsGradeDecreasing() List<Student></div> <div>sortMathsGradeIncreasing() List<Student></div> <div>sortPhysicsGradeDecreasing() List<Student></div> <div>sortPhysicsGradeIncreasing() List<Student></div> <div>sortYearOfBirthDecreasing() List<Student></div> <div>sortYearOfBirthIncreasing() List<Student></div> <div>studentAt(int) Student</div> <div>instance StudentManager</div> <div>studentList List<Student></div>	<div>COMMA_DELIMITER String</div> <div>App()</div> <div>init() void</div> <div>main(String[]) void</div> <div>parseDataLineToArray(String) String[]</div> <div>parseDataLineToList(String) List<String></div> <div>readListData(String) void</div> <div>testFilterStudentsHigherThanAverageGrade() void</div> <div>testFilterStudentsHighestAverageGrade() void</div> <div>testFilterStudentsHighestMathsGrade() void</div> <div>testFilterStudentsLowerThanAverageGrade() void</div> <div>testFilterStudentsLowestAverageGrade() void</div> <div>testFilterStudentsLowestMathsGrade() void</div> <div>testOriginalData() void</div> <div>testSortAverageGradeDecreasing() void</div> <div>testSortAverageGradeIncreasing() void</div> <div>testSortYearOfBirthDecreasing() void</div> <div>testSortYearOfBirthIncreasing() void</div>	<div>Student()</div> <div>averageGrade double</div> <div>chemistryGrade double</div> <div>firstname String</div> <div>id String</div> <div>lastname String</div> <div>mathsGrade double</div> <div>physicsGrade double</div> <div>yearOfBirth int</div>

Powered by yFiles

Bài 3 (4 điểm). Viết chương trình Java được thiết kế như sơ đồ dưới đây.

- Hoàn thiện code cho trong các file source code mẫu được cung cấp.
- Thực hiện tất cả các phương thức test trong lớp client **App**.
- Lưu kết quả chạy chương trình vào file text có tên <Ten_MaSinhVien_CandidateManager>.txt (ví dụ, NguyenVanA_123456_CandidateManager.txt).
- Nén tất cả các file source code và file kết quả chạy chương trình theo định dạng zip và đặt tên <Ten_MaSinhVien_CandidateManager >.zip (ví dụ, NguyenVanA_123456_CandidateManager.zip), nộp lên classroom.

