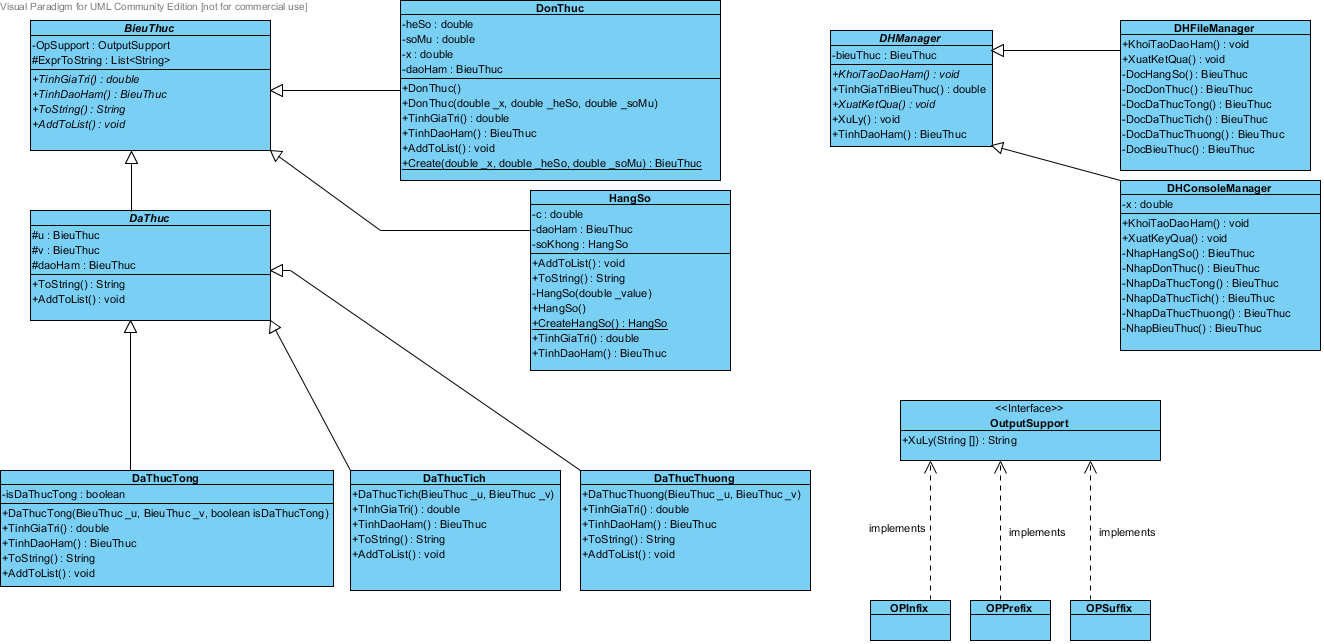
Bài tập Biểu thức

Nhóm thực hiện: 20

**Mục lục**

1. **Sơ đồ thiết kế (Class Diagram)**
2. **Giải thích các thành phần trong sơ đồ thiết kế**
   1. **Bài tập biểu thức 1**
      1. Các lớp đối tượng biểu diễn các loại biểu thức
      2. Xây dựng các phương thức khởi tạo cho từng loại biểu thức
      3. Cài đặt phương thức tính giá trị và tính đạo hàm cho từng loại biểu thức
   2. **Bài tập biểu thức 2**
      1. Trang bị khả năng trả về chuỗi biểu thức biểu diễn 1 đối tượng BieuThuc
      2. Giải pháp giúp nâng cao tốc độ tính đạo hàm của các biểu thức lồng nhau
   3. **Bài tập biểu thức 3**
      1. Cách thức khởi tạo giúp đơn thức có hệ số a hoặc số mũ bằng 0 được chuyển về hằng số tương ứng
      2. Bổ sung lớp hằng số để giúp hằng số 0 trở thành giá trị dùng chung toàn ứng dụng
   4. **Bài tập biểu thức 4**
      1. Phương án giúp cải tiến phương thức ToString()
3. **Sơ đồ thiết kế (Class Diagram)**
4. **Giải thích các thành phần trong sơ đồ thiết kế**
   1. **Bài tập biểu thức 1:**
      1. **Các yêu cầu:**
         * Các lớp đối tượng biểu diễn các loại biểu thức
         * Xây dựng các phương thức khởi tạo cho từng loại biểu thức
         * Cài đặt phương thức tính giá trị và tính đạo hàm cho từng loại biểu thức
      2. Hệ thống các lớp:
         * Lớp abstract BieuThuc
         * Lớp HangSo kế thừa lớp BieuThuc
         * Lớp DonThuc kế thừa lớp BieuThuc
         * Lớp abstract DaThuc kế thừa lớp BieuThuc
         * Lớp DaThucTong kế thừa lớp DaThuc
         * Lớp DaThucTich kế thừa lớp DaThuc
         * Lớp DaThucThuong kế thừa lớp DaThuc
      3. Mẫu sử dụng: Composite (Xem hình class diagram)
         * 1 Biểu thức có thể thuộc 1 trong 3 loại Hằng số, Đơn thức hoặc Đa thức. Vì vậy tạo lớp abstract BieuThuc và 3 lớp HangSo, DonThuc, DaThuc kế thừa từ lớp BieuThuc
         * 1 Đa thức có thể thuộc 3 loại Đa thức tổng, đa thức tích hoặc đa thức thương. Vì vậy tạo lớp DaThuc là lớp abstract và 3 lớp DaThucTong, DaThucTich, DaThucThuong kế thừa từ lớp DaThuc.
         * Mẫu composite được áp dụng ở chỗ trong lớp DaThuc có 2 thuộc tính mang kiểu BieuThuc là u và v.
         * Phương thức TinhDaoHam() và TinhGiaTri() là 2 lớp ảo trong lớp BieuThuc sẽ được override cụ thể theo từng loại HangSo, DonThuc, DaThucTong, DaThucTich, DaThucThuong.
   2. **Bài tập biểu thức 2**
      1. Trang bị khả năng trả về chuỗi biểu thức biểu diễn 1 đối tượng BieuThuc
         * Trong các lớp BieuThuc, HangSo, DonThuc, DaThuc, DaThucTong, DaThucTich, DaThucThuong đều override lại hàm ToString().
         * Hàm ToString() trong lớp BieuThuc và DaThuc khai báo là phương thức abstract. Việc override cụ thể ToString() được thực hiện trong HangSo, DonThuc, DaThucTong, DaThucTich, DaThucThuong
      2. Giải pháp giúp nâng cao tốc độ tính đạo hàm của các biểu thức lồng nhau
   3. **Bài tập biểu thức 3**
      1. Cách thức khởi tạo giúp đơn thức có hệ số a hoặc số mũ bằng 0 được chuyển về hằng số tương ứng
         * Trong lớp DonThuc, tạo 2 phương thức:
           1. **public static BieuThuc Create(double \_x, double \_heSo, double \_soMu)**
           2. **private DonThuc(double \_x, double \_heSo, double \_soMu)**
      2. Bổ sung lớp hằng số để giúp hằng số 0 trở thành giá trị dùng chung toàn ứng dụng
         * Trong lớp HangSo:
           1. Thêm 1 thuộc tính static mang kiểu HangSo đặt tên là **soKhong**
           2. Thêm 1 phương thức **static CreateHangSo()** trả về HangSo
   4. **Bài tập biểu thức 4**
      1. Phương án giúp cải tiến phương thức ToString()
         * Thêm vào 1 interface tên là OutputSupport, trong đó có 1 phương thức **public string XuLy(List<string>)**
         * 3 lớp OPInfix, OPPrefix, OPSuffix sẽ cài đặt interface OutputSupport, cụ thể là cài đặt phương thức XuLy trong interface này