

Bài thực hành 15

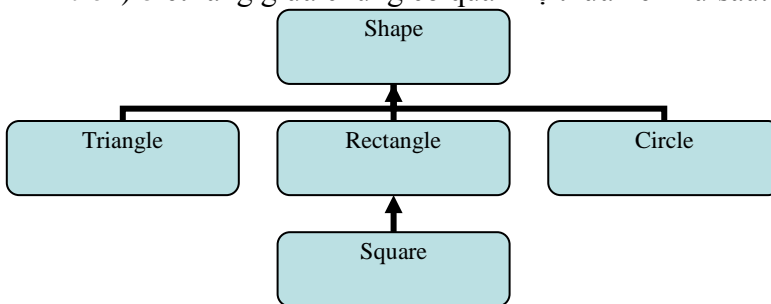
Mục tiêu

- Thừa kế
- Xử lý ngoại lệ

Bài tập

Câu 1. [shape.cpp]

Viết chương trình định nghĩa các lớp Shape (biểu diễn một hình trong hình học phẳng), Triangle (biểu diễn tam giác), Rectangle (biểu diễn hình chữ nhật), Square (biểu diễn hình vuông), Circle (biểu diễn hình tròn) biết rằng giữa chúng có quan hệ thừa kế như sau:



- Lớp Shape có 3 hàm thành viên:
 - `getArea()` tính diện tích của hình. Do chưa xác định là hình gì nên trả về -1.
 - `getPerimeter()` tính chu vi của hình. Do chưa xác định là hình gì nên trả về -1.
 - `draw()` thực hiện vẽ hình. Do chưa xác định là hình gì nên in ra “Vẽ hình”.
- Lớp Triangle với
 - 3 biến thành viên `a, b, c` biểu diễn độ dài 3 cạnh
 - Hàm kiến tạo nhận 3 tham số `double` là độ dài khởi tạo cho 3 cạnh
 - Định nghĩa lại hàm `getArea()` trả về diện tích tam giác
 - Định nghĩa lại hàm `getPerimeter()` trả về chu vi tam giác
 - Định nghĩa lại hàm `draw()`: in ra “Vẽ hình tam giác với độ dài 3 cạnh là `a, b, c`”.
- Lớp Rectangle với
 - 2 biến thành viên `length, width` biểu diễn chiều dài và chiều rộng
 - Hàm kiến tạo nhận 2 tham số `double` là độ dài khởi tạo cho 2 cạnh
 - Hàm `getLength()` trả về chiều dài
 - Hàm `getWidth()` trả về chiều rộng
 - Định nghĩa lại hàm `getArea()` trả về diện tích hình chữ nhật
 - Định nghĩa lại hàm `getPerimeter()` trả về chu vi hình chữ nhật
 - Định nghĩa lại hàm `draw()`: in ra “Vẽ hình chữ nhật với chiều dài `length` chiều rộng `width`”.
- Lớp Square
 - Hàm kiến tạo nhận 1 tham số `double` là độ dài khởi tạo cho `length`, đồng thời khởi tạo `width` bằng tham số này
 - Định nghĩa lại hàm `draw()`: in ra “Vẽ hình vuông chiều dài cạnh là `length`”.
- Lớp Circle với
 - 1 thành viên dữ liệu `radius` biểu diễn bán kính
 - Hàm kiến tạo nhận 1 tham số `double` là giá trị khởi tạo cho bán kính
 - Hàm `getRadius()` trả về bán kính

- Định nghĩa lại hàm `getArea()` trả về diện tích hình tròn
- Định nghĩa lại hàm `getPerimeter()` trả về chu vi hình tròn
- Định nghĩa lại hàm `draw()`: in ra “Vẽ hình tròn với bán kính radius”.

Câu 2. [exceptions.cpp]

- Cho trước tệp `18-04.cpp` chứa mã nguồn của `Display 18.4` trong giáo trình. Chương trình này cài đặt và sử dụng lớp `NegativeNumber` biểu diễn ngoại lệ số âm và lớp `DivideByZero` biểu diễn ngoại lệ phép chia cho 0. Hãy chạy thử chương trình và quan sát kết quả.
- Quan sát ví dụ `exception3.cpp` về cách cài đặt lớp ngoại lệ (`EmptyArrayException`) thừa kế từ lớp `exception` của thư viện chuẩn và định nghĩa lại hàm `what()` để chỉnh sửa các lớp ngoại lệ trong `18-04.cpp` (sao cho chúng cũng thừa kế từ lớp `exception` và định nghĩa hàm `what()`). Trong hàm `main()` thay tất cả các khối `catch` bằng một khối `catch(exception& e)` – trong đó in ra thông điệp trả về bởi `what()`.