***Đề 1:***

|  |  |  |
| --- | --- | --- |
| ***Bảng điểm*** | | ***Thời gian*** |
| Câu 1: 5 điểm | Câu 2: 5 điểm | 60 phút |
| a. 0.5 đ | a. 1.0 đ |
| b. 1.5 đ | b. 2.5 đ |
| c. 1.5 đ | c. 1.5 đ |
| d. 1.5 đ | d. X |

1. ***Xây dựng Lớp TamGiac gồm có các phương thức:***
   1. Phương thức khởi tạo không tham số: TamGiac();  
      Phương thức khởi tạo có 3 tham số TamGiac(float a, float b, float c)
   2. Nhập các cạnh của tam giác: public void nhập()  
      Tính chu vi của tam giác: public float getChuVi()  
      Tính diện tích của tam giác: public float getDienTich()
   3. Kiểm tra xem 3 cạnh vừa nhập vào có phải là 3 cạnh của tam giác hay không: public bool kiemtra():
   4. In ra màn hình đó là loại tam giác nào: cân, vuông, vuông cân, đều hay thường: public string phanloaiTG()
2. ***Thực hiện các yêu cầu sau:***
   1. Tạo một lớp điểm biểu diễn một điểm trong không gian bao gồm các thuộc tính:
      * Hoành độ

* Tung độ
* Cao độ
  1. Bổ sung các phương thức sau cho lớp
* Phương thức khởi tạo không có tham số nào.
* Phương thức khởi tạo nhận 3 tham số thực làm tham số
* Phương thức nhập vào một điểm từ bàn phím
* Phương thức in thông tin một điểm ra màn hình.
  1. Viết phương thức Main kiểm tra các phương thức đã cài đặt cho lớp
* Yêu cầu: Toạ độ của một điểm có kiểu giá trị là float

***Đề 2:***

|  |  |  |
| --- | --- | --- |
| ***Bảng điểm*** | | ***Thời gian*** |
| Câu 1: 3.5 điểm | Câu 2: 6.5 điểm | 75 phút |
| a. 0.5 đ | a. 0.5 đ |
| b. 0.5 đ | b. 0.75 đ |
| c. 0.5 đ | c. 0.75 đ |
| d. 1.0 đ | d. 0.75 đ |
| e. 1.0 đ | e. 0.75 đ |
| f. X | f. 1.5 đ |
| g. X | g. 1.5 đ |

1. ***Xây dựng lớp Stack để mô phỏng một Stack bao gồm:***
2. Hàm tạo Stack(int n) để khởi tạo Stack có tối đa n phần tử
3. Phương thức isEmpty để kiểm tra xem Stack có phần tử nào hay không?
4. Phương thức isFull để kiểm tra xem Stack đã đầy hay chưa?
5. Phương thức push để thêm một phần tử vào Stack
6. Phương thức pop để lấy một phần tử ra khỏi Stack

1. ***Xây dựng lớp thực hiện các phép toán trên ma trận, bao gồm:***
2. Hàm tạo ma MaTran(int soHang, int soCot) dùng để khởi tạo một ma trận có số hàng là soHang và số cột là soCot.
3. Phương thức “nhapMaTran” để nhập ma trận từ bàn phím
4. Phương thức “xuatMaTran” để in ma trận ra màn hình
5. Phương thức “congMaTran” để cộng hai ma trận
6. Phương thức “truMaTran” để thực hiện phép hiệu hai ma trận
7. Phương thức “doiDauMaTran” để đổi dấu các phần tử của ma trận
8. Phương thức Main để kiểm tra ma trận trên

***Đề 3:***

|  |  |  |
| --- | --- | --- |
| ***Bảng điểm*** | | ***Thời gian*** |
| Câu 1: 6.0 điểm | Câu 2: 4.0 điểm | 75 phút |
| a. 0.5 đ | a. 0.5 đ |
| b. 0.5 đ | b. 0.5 đ |
| c. 0.5 đ | c. 0.5 đ |
| d. 2.0 đ | d. 0.75 đ |
| e. 0.5 đ | e. 0.75 đ |
| f. 1.0 đ | f. 1.0 đ |
| g. 1.0 đ | g. X |

* + - 1. ***Tạo một lớp biểu diễn phân số*** (có tử và mẫu là các số nguyên) gồm các thuộc tính:

Tử số

Mẫu số

Bổ sung các phương thức sau cho lớp:

1. Phương thức khởi tạo không có tham số nào, tạo phân số bằng 0.
2. Phương thức khởi tạo nhận 2 số nguyên làm tham số.P
3. Phương thức khởi tạo nhận một đối tượng thuộc chính lớp này làm tham số.
4. Phương thức rút gọn phân số
5. Phương thức cộng, trừ, nhân, chia phân số này với một phân số khác.
6. Phương thức in phân số ra màn hình.
7. Viết hàm main kiểm tra các phương thức đã tạo.  
     
   * + 1. ***Xây dựng lớp Queue*** 
          1. Hàm tạo Queue(int n) để khởi tạo Queue có tối đa n phần tử
          2. Phương thức isEmpty để kiểm tra xem Queue có phần tử nào hay không?
          3. Phương thức isFull để kiểm tra xem Queue đã đầy hay chưa?
          4. Phương thức push để thêm một phần tử vào Queue.
          5. Phương thức pop để lấy một phần tử ra khỏi Queue.
          6. Viết hàm main kiểm tra các phương thức đã tạo.