**BÀI THỰC HÀNH – BÀI SỐ 8**

**Lab 01:**

1. Xây dựng class HinhChuNhat để mô tả hình chữ nhật gồm các thông tin sau:

|  |  |  |
| --- | --- | --- |
|  |  | Mô tả |
| Thuộc tính | Chiều dài | Kiểu double |
| Chiều rộng |
| Phương thức | Hàm khởi tạo không tham số |  |
| Hàm khởi tạo đầy đủ tham số | Khởi tạo hình chữ nhật do người dùng đặt |
| Gettter/setter cho chiều dài, chiều rộng |  |
| getChuVi() | Tính và trả về chu vi của hình chữ nhật hiện tại |
| getDienTich() | Tính và trả về diện tích của hình chữ nhật hiện tại |
| hienThiThongTin() | Trả về thông tin của hình chữ nhật hiện tại có định dạng “HCN(dai, rong): chu vi = getChuVi(), dien tich = getDienTich()” |

1. Xây dựng class HinhVuong kế thừa từ class HinhChuNhat để mô tả hình vuông gồm các thông tin sau:

|  |  |  |
| --- | --- | --- |
|  |  | Mô tả |
| Thuộct tính | Cạnh | Kiểu double |
| Phương thức | Hàm khởi tạo không tham số |  |
| Hàm khởi tạo đầy đủ tham số | Khởi tạo hình vuông do người dùng đặt |
| Getter/setter cho cạnh |  |
| getChuVi() | Ghi đè phương thức getChuVi() và trả về chu vi của hình vuông hiện tại |
| getDienTich() | Ghi đè phương thức getDienTich() và trả về diện tích của hình vuông hiện tại |
| hienThiThongTIn() | Ghi đè phương thức hienThiThongTIn() và trả về thông tin hình vuông hiện tại theo định dạng “HV(canh = getCanh()): chu vi = getChuVi(), dien tich = getDienTich()” |

1. Xây dựng class Lab01\_Main, bên trong có chứa phương thức main() nhằm kiểm tra sự hoạt động của chương trình theo kịch bản sau:

INPUT:

* Dòng đầu tiên là số nguyên T (1 <= T <= 100) cho biết số lượng testcase của bài toán.
* Mỗi testcase được cho trên 5 dòng với:
  + Dòng 1: 4 số tự nhiên M, N, P, Q (2 <= M, N <= 100; 1<=P<M ,1<=Q<N) lần lượt là số lượng Hình chữ nhật, hình vuông cần nhập, p vị trí hình chữ nhật cần hiển thị thông tin, q vị trí hình vuông cần hiển thị thông tin.
  + Dòng 2: 2\*M số thực với 2 số liên tiếp nhau cho biết thông tin chiều dài, chiều rộng của hình chữ nhật thứ i
  + Dòng 3: N số thực cho biết thông tin độ dài cạnh của hình vuông thứ i
  + Dòng 4: P số tự nhiên cho biết vị trí hình chữ nhật thứ i cần hiển thị thông tin.
  + Dòng 5: Q số tự nhiên cho biết vị trí hình vuông thứ i cần hiển thị thông tin.

OUTPUT:

* Dòng 1: chuỗi Case #case với: case là stt của testcase
* P dòng tiếp theo là kết quả của dòng thứ 4 trong input của mỗi testcase. Chu vi và diện tích lấy 3 chữ số sau dấu phẩy “,”
* Q dòng tiếp theo là kết quả của dòng thứ 5 trong input của mỗi testcase. Chu vi và diện tích lấy 3 chữ số sau dấu phẩy “,”

EXAMPLE:

* INPUT:

|  |  |
| --- | --- |
| Giá trị | Mô tả |
| 1 | T |
| 3 3 1 1 | M, N, P, Q |
| 19.99 92.60 72.12 12.92 31.29 81.28 | Mảng chiều dài-rộng của M hình chữ nhật |
| 14.07 93.85 79.84 | Mảng độ dài của N cạnh hình vuông |
| 0 | P |
| 0 | Q |

* OUTPUT:

Case #1:

HCN(19.990000, 92.600000): chu vi = 225.180, dien tich = 1851.074

HV(canh = 14.070000): chu vi = 56.280, dien tich = 197.965

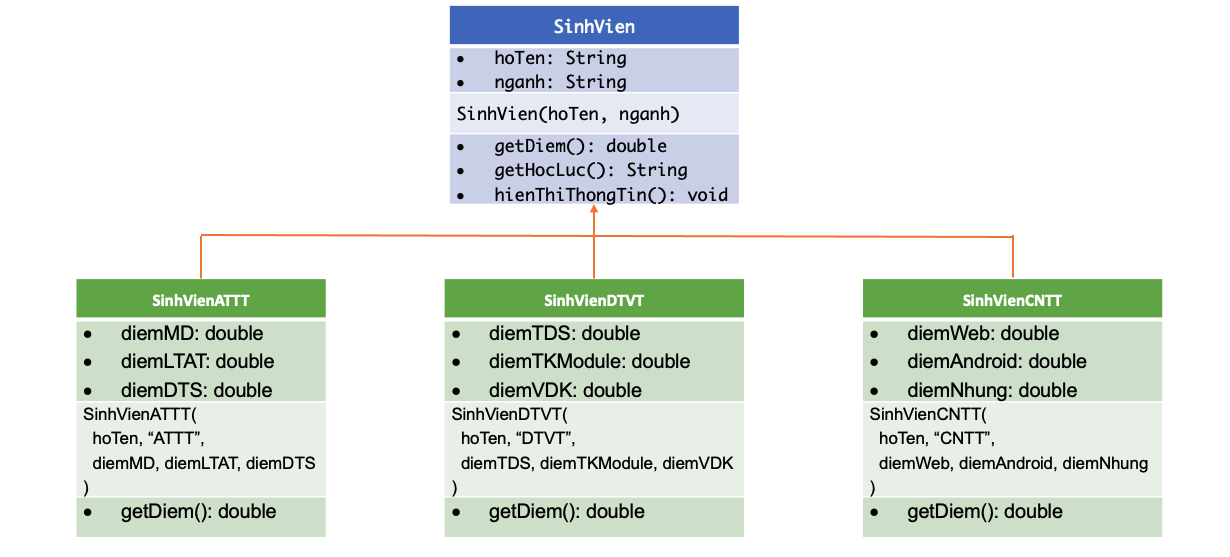
**Lab 02:**

1. Xây dựng class trừu tượng SinhVien để mô tả thông tin của một Sinh viên gồm những thông tin sau:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Mô tả | |
| Thuộc tính | Họ tên | Kiểu String | |
| Ngành học |
| Phương thức | Hàm khởi tạo đầy đủ tham số |  | |
| getDiem() | Phương thức trừu tượng (abstract). Kiểu double | |
| Gettter/setter cho họ tên và ngành học |  | |
| getHocLuc() | Tính và trả về học lực của sinh viên hiện tại: | |
| Yếu | Điểm < 4.0 |
| Trung bình | 4.0 <= điểm < 6.0 |
| Trung bình khá | 6.0 <= điểm < 7.0 |
| Khá | 7.0 <= điểm < 8.0 |
| Giỏi | 8.0 <= điểm < 9.0 |
| Xuất sắc | Điểm >= 9.0 |
| hienThiThongTin() | Trả về thông tin của Sinh viên hiện tại theo định dạng “SV: hoTen – nganhHoc – điểm – học lực”  Lưu ý: Họ tên lấy 20 vị trí, căn lề trái | |

1. Xây dựng class SinhVienATTT, SinhVienCNTT, SinhVienDTVT kế thừa từ class SinhVien với các thông tin sau:

* Sinh viên ATTT gồm:
  + Thuộc tính điểm: Mã độc, Lập trình An toàn, Điều tra số.
  + Ghi đè phương thức getDiem(): return (diemMD\*2+diemLTAT\*2+diemDTS)/5
* Sinh viên CNTT gồm:
  + Thuộc tính điểm: Web, Android, Nhúng.
  + Ghi đè phương thức getDiem(): return (diemWeb+diemAndroid\*2+diemNhung\*2)/5
* Sinh viên ĐTVT gồm:
  + Thuộc tính điểm: Truyền dẫn số, Thiết kế module, Vi điều khiển.
  + Ghi đè phương thức getDiem(): return (diemVDK\*2+diemTDS+diemTKModule)/4



1. Xây dựng class Lab02\_Main, bên trong có chứa phương thức main() nhằm kiểm tra sự hoạt động của chương trình theo kịch bản sau:

INPUT:

* Dòng đầu tiên là số nguyên T (1 <= T <= 100) cho biết số lượng testcase của bài toán.
* Mỗi testcase được cho trên xx dòng với:
  + Dòng 1: 3 số nguyên M, N, P cho biết số lượng sinh viên tương ứng được nhập của mỗi ngành ATTT, CNTT, ĐTVT
  + Dòng 2: 3 số nguyên x, y, z cho biết cần hiển thị thông tin của Sinh viên x thuộc khoa ATTT, sinh viên y thuộc khoa CNTT, sinh viên z thuộc khoa DTVT. (x, y, z đảm bảo nằm trong khoảng dữ liệu đã cho).
  + 2\*M dòng tiếp theo, dòng đầu tiên là Họ tên sinh viên, dòng thứ 2 gồm 3 thông tin liên tiếp tương ứng: điểm mã độc, điểm lập trình AT, điểm điều tra số.
  + 2\*N dòng tiếp theo, dòng đầu tiên là Họ tên sinh viên, dòng thứ 2 gồm 3 thông tin liên tiếp tương ứng: điểm Web, điểm Android, điểm Nhúng.
  + 2\*P dòng tiếp theo, dòng đầu tiên là Họ tên sinh viên, dòng thứ 2 gồm 3 thông tin liên tiếp tương ứng: điểm Truyền dẫn số, điểm Thiết kế module, điểm Vi điều khiển.

OUTPUT:

* Dòng 1: chuỗi Case #case với: case là stt của testcase
* Dòng 2: hiển thị thông tin sinh viên x
* Dòng 3: hiển thị thông tin sinh viên y
* Dòng 4: hiển thị thông tin sinh viên z

EXAMPLE:

* INPUT:

|  |  |
| --- | --- |
| **Giá trị** | **Mô tả** |
| 1 | T |
| 2 2 2 | M, N, P |
| 1 1 0 | x, y, z |
| Nguyen Van A  1.0 6.5 3.9 | 2 sinh viên ATTT |
| Nguyen Van B  0.4 0.5 4.1 |
| Tran Cong C  6.4 1.3 7.4 | 2 sinh viên CNTT |
| Tran Dinh D  3.0 9.8 7.4 |
| Dinh Manh E  4.6 3.7 7.7 | 2 sinh viên DTVT |
| Duong Dinh F  9.4 6.5 7.2 |

* OUTPUT

SV: Nguyen Van B – ATTT – 1.18 – Yeu

SV: Tran Dinh D – CNTT – 7.48 – Kha

SV: Dinh Manh E – DTVT – 5.93 – Trung binh