



JAVA CORE

CYBERSOFT
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH
CYBERSOFT ACADEMY

ĐẶC TẢ PROJECT

Xây dựng ứng dụng **Quản lý nhân viên** bằng Java với các yêu cầu sau:

- Công ty có 3 loại nhân viên: **sếp, trưởng phòng, nhân viên thường**
- Mỗi nhân viên phải có các thông tin: mã số, họ tên, số điện thoại, số ngày làm việc, hệ số lương, tính lương, ...
- Có thể thêm, xóa, sửa thông tin một nhân viên
- Sắp xếp nhân viên theo tên, lương
- Tìm kiếm nhân viên theo tên và mã nhân viên.
- Tư duy mở rộng ...

ĐỐI TƯỢNG

- Đối tượng là một thực thể (instance) phản ánh đúng vật đó trong đời sống, và được mô tả trong lập trình.
- Một đối tượng có 2 thứ cần lưu ý: **Thuộc Tính + Phương thức**
- **Thuộc tính** là những gì thuộc về đối tượng đó, mô tả cho đối tượng đó.
- **Phương thức** là những hành động mà đối tượng có thể làm được, luôn nhớ là chỉ liên quan đến đối tượng này mà thôi.
- VD: Nhân viên Nguyễn Văn A là một đối tượng
 - Thuộc tính: họ tên, chiều cao, cân nặng, bằng cấp, nghề nghiệp, lương, tuổi, vị trí, ...
 - Phương thức: tính lương, ăn, làm việc, báo cáo, ...

Ví dụ về đối tượng

Đối tượng	Thuộc tính / Dữ liệu	Hành vi/ Phương thức
Phân số	<ul style="list-style-type: none">- Tử- Mẫu	<ul style="list-style-type: none">- Cộng- Trừ- Nhân- Chia-
Xe	<ul style="list-style-type: none">- Màu- Kích thước- Trọng lượng- Mẫu- Nhà sản xuất-	<ul style="list-style-type: none">- Khởi động máy- Tăng tốc- Tắt máy-

Ví dụ về đối tượng (tt)

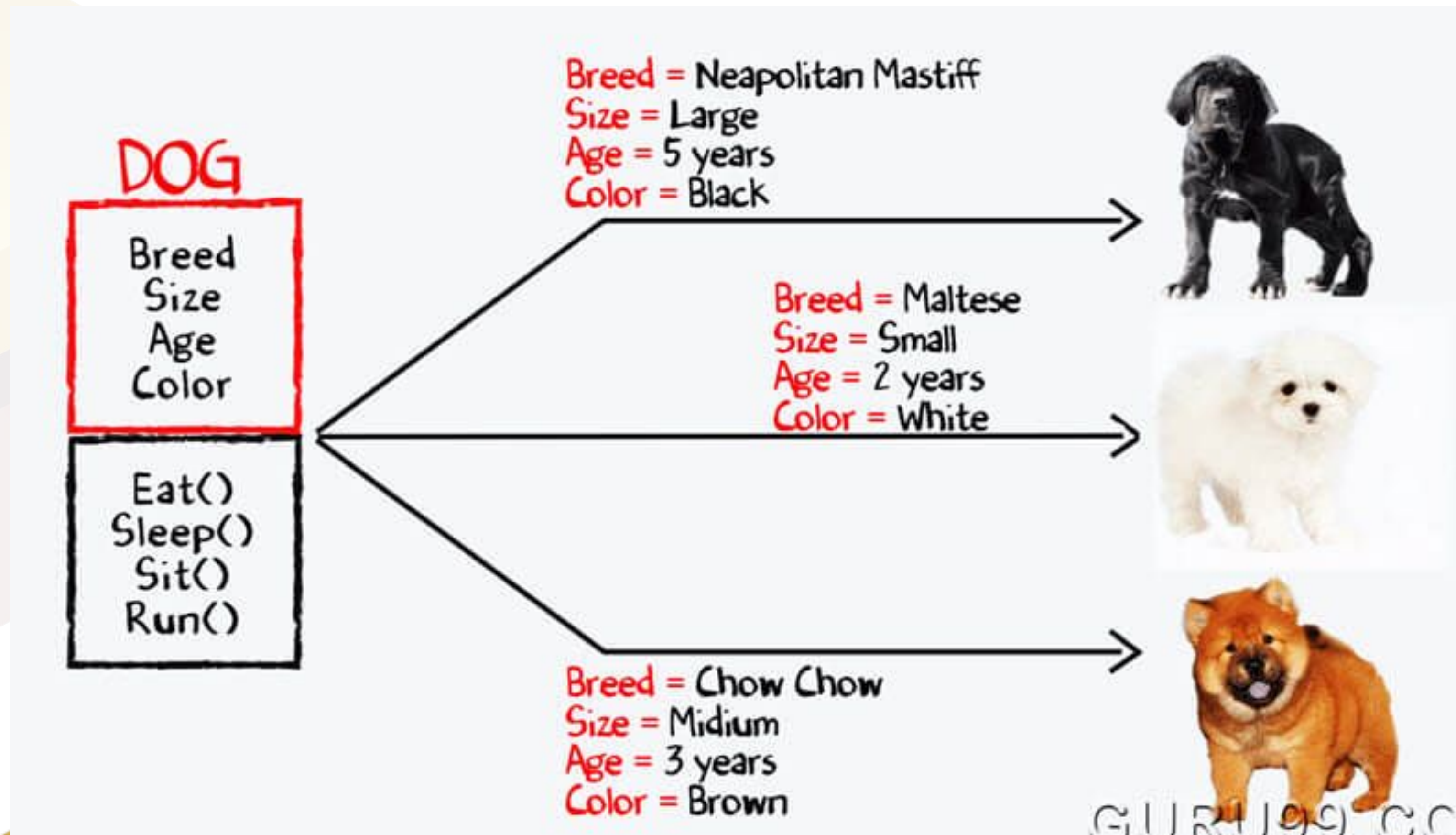
Đối tượng	Thuộc tính / Dữ liệu	Hành vi/ Phương thức
Tài khoản	<ul style="list-style-type: none"> - Tên tài khoản - Người sở hữu - Số tài khoản - Loại tài khoản - Số dư - 	<ul style="list-style-type: none"> - Hiện thị số dư hiện tại - Rút tiền mặt - Thống kê - Đóng tài khoản -
Nhân viên	<ul style="list-style-type: none"> - Mã nhân viên - Họ - Tên - CMND - Giới tính - 	<ul style="list-style-type: none"> - Hiện thị thông tin nhân viên - Thay đổi thông tin nhân viên -

Ví dụ về đối tượng (tt)

Đối tượng	Thuộc tính / Dữ liệu	Hành vi/ Phương thức
Sinh viên	<ul style="list-style-type: none"> - Mã sinh viên - Tên - Họ - Giới tính - Địa chỉ - 	<ul style="list-style-type: none"> - Thay đổi thông tin - Hiển thị thông tin - Lấy điểm trung bình học kì -
Trường học	<ul style="list-style-type: none"> - Mã trường - Tên trường - Địa chỉ - 	<ul style="list-style-type: none"> - Hiển thị thông tin trường - Đếm sinh viên trường -

LỚP ĐỐI TƯỢNG

Một tập hợp các đối tượng, khuôn mẫu/template → LỚP ĐỐI TƯỢNG



XÂY DỰNG CLASS NHÂN VIÊN

```
protected String _MaNV;  
protected String _HoTen;  
protected int _NamSinh;  
protected String _ChucVu;  
protected float _HeSoLuong;  
protected int _PhuCap;  
protected String _UserName;  
protected String _PassWord;  
  
public NhanVien(String _MaNV, String _HoTen, int _NamSinh, String _ChucVu, float _HeSoLuong, int _PhuCap,  
    String _UserName, String _PassWord) {  
    super();  
    this._MaNV = _MaNV;  
    this._HoTen = _HoTen;  
    this._NamSinh = _NamSinh;  
    this._ChucVu = _ChucVu;  
    this._HeSoLuong = _HeSoLuong;  
    this._PhuCap = _PhuCap;  
    this._UserName = _UserName;  
    this._PassWord = _PassWord;  
}
```

Constructor: hàm khởi tạo

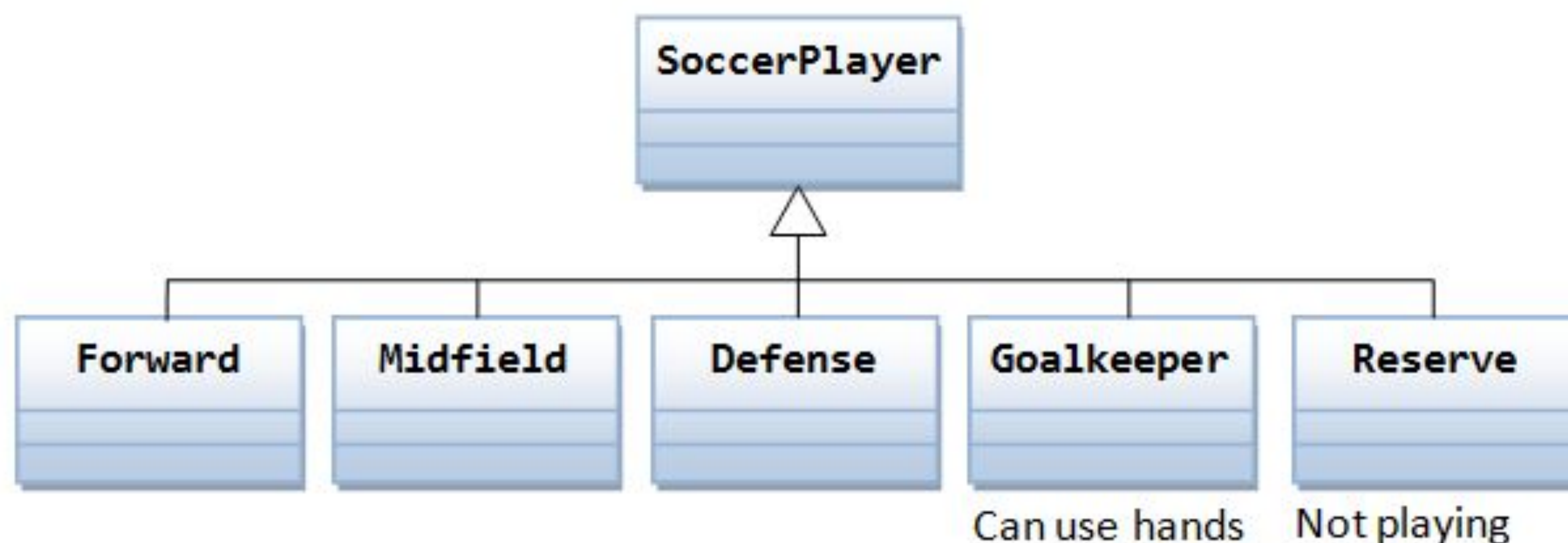
ENCAPSULATION

Tính đóng gói

```
public String get_MaNV() {  
    return _MaNV;  
}  
  
public void set_MaNV(String _MaNV) {  
    this._MaNV = _MaNV;  
}
```

INHERITANCE

VD về kế thừa trong thực tế



INHERITANCE

Xây dựng lớp đối tượng Sep kế thừa từ lớp đối tượng NhanVien

```
public class Sep extends NhanVien {  
    private int _SoNhanVienThuong;  
    private int _SoTruongPhong;  
    public Sep(String _MaNV, String _HoTen, int _NamSinh, float _HeSoLuong, int _PhuCap, String _UserName, String _PassWord) {  
        super(_MaNV, _HoTen, _NamSinh, "Sep", _HeSoLuong, _PhuCap, _UserName, _PassWord);  
    }  
}
```

Xây dựng lớp đối tượng TruongPhong kế thừa từ lớp đối tượng NhanVien

```
public class TruongPhong extends NhanVien {  
    private String Phong;  
    private int _SoNhanVienTrongPhong;  
  
    public TruongPhong(String _MaNV, String _HoTen, int _NamSinh, float _HeSoLuong, int _PhuCap, String _UserName, String _PassWord) {  
        super(_MaNV, _HoTen, _NamSinh, "TruongPhong", _HeSoLuong, _PhuCap, _UserName, _PassWord);  
    }  
}
```


INSTANCE

Tạo các instance (thể hiện) từ các template/class

```
CongTy congty = new CongTy();  
congty.ThemNhanVien(new Sep("001", "Phó Nghĩa Văn", 1995, 2.0f, 10000000, "hackagon", "hackagon"));  
congty.ThemNhanVien(new NhanVienThuong("002", "Minh Tuấn", 1995, 1.0f, 1000000, "minhtuan", "minhtuan"));  
congty.ThemNhanVien(new TruongPhong("003", "Trung Hiếu", 1995, 1.5f, 1000000, "trunghieu", "trunghieu"));
```


ĐĂNG NHẬP

- Sử dụng cấu trúc if else, đệ quy để thực hiện chức năng đăng nhập

Xây dựng hàm DangNhap() như sau:
Cho người dùng nhập username và password. Nếu tồn tại và khớp sẽ gọi hàm HienThiMenu(). Nếu không, sẽ gọi lại chính hàm DangNhap()



XÂY DỰNG MENU

Tùy theo _ChucVu của nhân viên mà hiển thị Menu cho thích hợp. Mỗi một tùy chọn sẽ gọi một hàm tương ứng. Cố gắng refactor code tốt nhất có thể.

```
1. Hiển thị danh sách nhân viên
2. Xem chi tiết thông tin của mình
3. Đăng xuất
Chon:
```

Menu nhân viên thường

```
1. Hiển thị danh sách nhân viên
2. Xem chi tiết thông tin của mình
3. Xem chi tiết thông tin của một nhân viên thường
4. Đăng xuất
Chon:
```

Menu trưởng phòng

```
1. Hiển thị danh sách nhân viên
2. Xem chi tiết thông tin của mình
3. Xem chi tiết thông tin của một nhân viên
4. Thêm nhân viên
5. Sửa thông tin nhân viên
6. Xóa nhân viên
7. Đăng xuất
Chon:
```

Menu sếp

SORT

SIMPLE SELECTION SORT

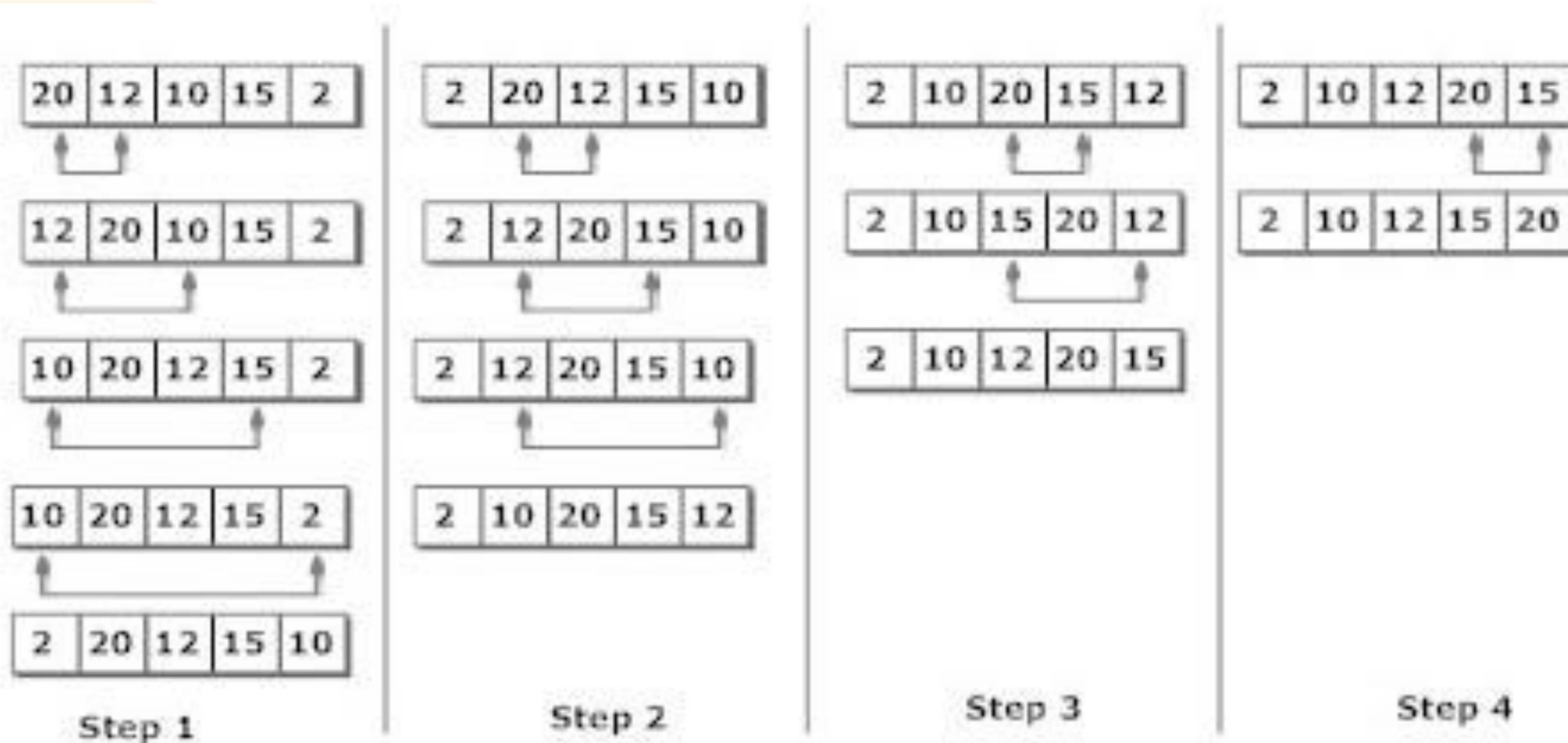


Figure: Selection Sort

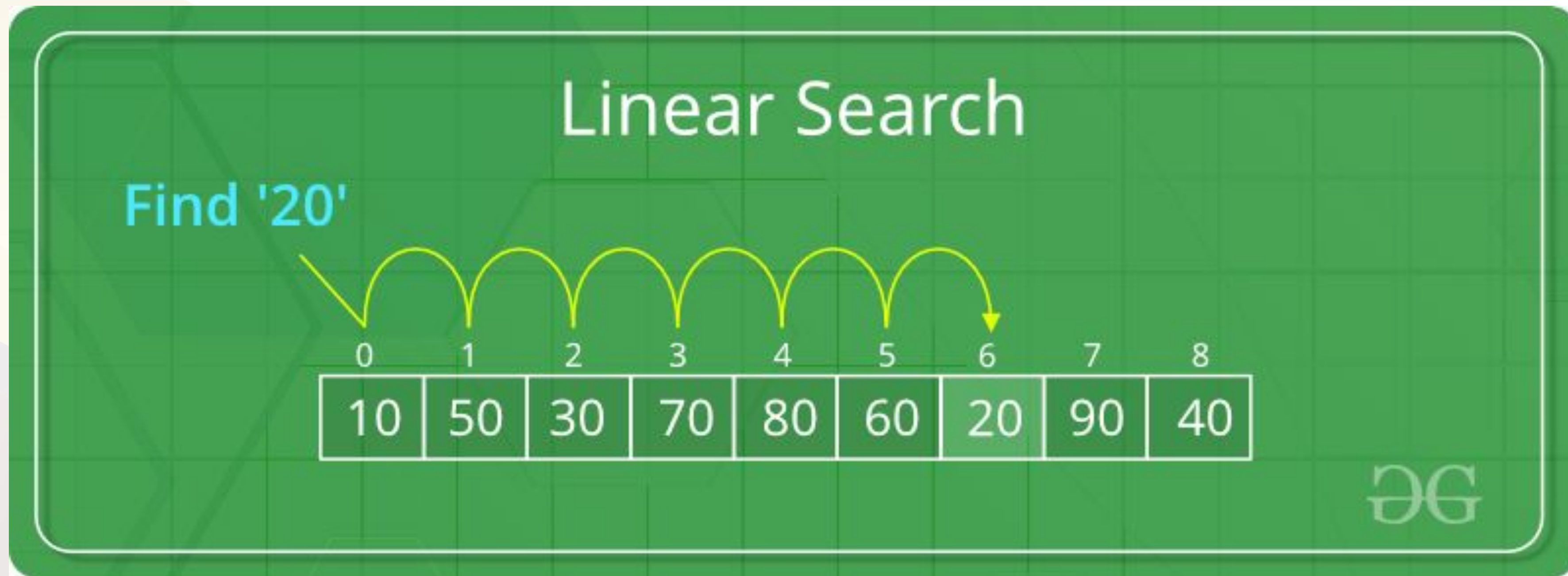


□ □ □



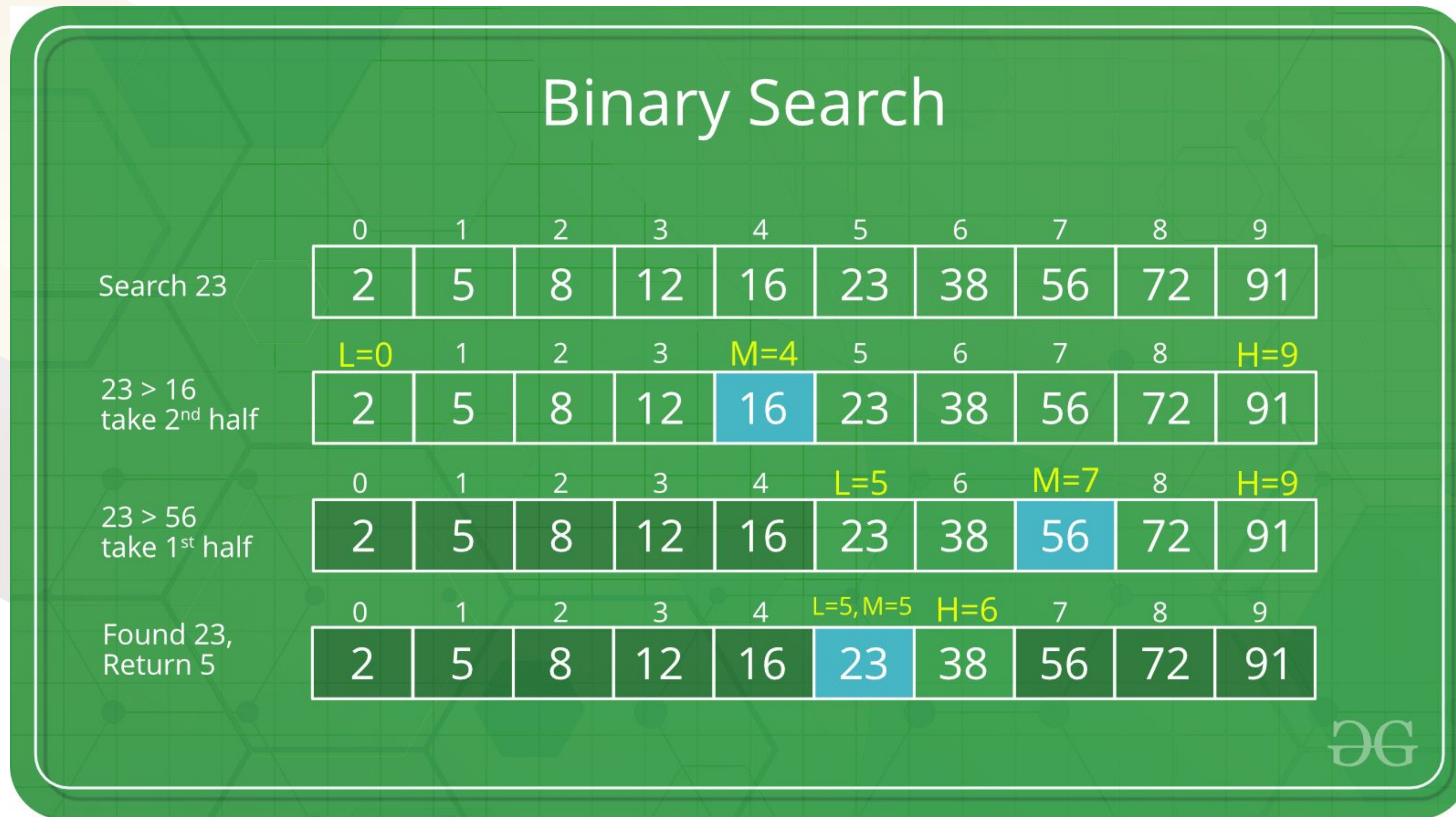
SEARCH

LINEAR SEARCH



SEARCH

BINARY SEARCH



BÀI TẬP

1. Viết chương trình tính diện tích và chu vi hình chữ nhật với các yêu cầu sau:

- Xây dựng lớp đối tượng **HìnhChuNhat** gồm có:
 - Các thuộc tính: chiều dài, chiều rộng
 - Tạo constructor nhận thông tin về chiều dài, chiều rộng để khởi tạo instance
 - Phương thức get, set
 - Phương thức tính diện tích hình chữ nhật
- Xây dựng **lớp đối tượng chứa hàm main** để tạo và tính chu vi, diện tích các hình chữ nhật được nhập từ bàn phím

BÀI TẬP

2. Viết chương trình quản lý sinh viên đơn giản với các yêu cầu sau:

- Tạo lớp SinhVien:

- Các thuộc tính: mã sinh nguyên, họ tên, điểm toán, điểm lý, điểm hóa
- Constructor nhận đầy đủ thông tin để khởi tạo giá trị cho các biến instance
- Phương thức get và set
- Phương thức tính điểm trung bình
- Phương thức xếp loại sinh viên: xuất sắc, giỏi, khá, trung bình khá, trung bình, yếu

- Tạo lớp LopHoc:

- Thuộc tính: danh sách sinh viên
- Phương thức xuất ra danh sách sinh viên
- Phương thức xuất ra danh sách sinh viên xuất sắc/ giỏi/ khá/ trung bình khá/ trung bình/ yếu
- Phương thức xuất ra danh sách sinh viên có điểm môn toán/lý/hóa từ x trở lên (với x làm tham số)

- Xây dựng hàm Main để kiểm nghiệm

BÀI TẬP

3. Sở giao thông cần theo dõi việc đăng ký xe của người dân. Dựa vào thông tin trị giá xe và dung tích xylanh, sở giao thông tính mức phí đóng trước bạ khi mua xe như sau:

- Dưới 100cc, 1% giá trị xe
- Từ 100 đến 200cc, 3% giá trị xe
- Trên 200cc, 5% giá trị xe

Yêu cầu:

- Viết các lớp Xe, DanhSachXe với các thuộc tính, phương thức, constructor phù hợp. Thoát

BÀI TẬP

- Xây dựng lớp chứa hàm Main thực hiện các công việc sau:
 - Thêm xe
 - Sửa xe
 - Xóa xe
 - Xuất bảng kê khai mức phí phải đóng cho từng loại xe
 - Thoát

CYBERSOFT
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

BÀI TẬP

4. Một công ty du lịch quản lý thông tin về các chuyến xe. Các chuyến xe có 2 loại:

- Chuyến xe nội thành: mã, họ tên tài xế, số xe, số tuyến, số km đi được, doanh thu ($= \text{số km} * 20.000 \text{ VNĐ}$)
- Chuyến xe ngoại thành: mã, họ tên tài xế, số xe, nơi đến, số ngày đi được, doanh thu ($= \text{số ngày} * 500.000 \text{ VNĐ}$)

Thực hiện yêu cầu sau:

- Xây dựng các lớp, sử dụng kế thừa
- Xây dựng chương trình với các tính năng sau:
 - Nhập, xuất danh sách các chuyến xe
 - Tính tổng doanh thu cho tất cả chuyến xe và tổng doanh thu cho từng loại xe

BÀI TẬP

5. Một thư viện quản lý danh sách các loại sách. Thông tin về các loại sách:

- Sách giáo khoa: mã sách, tên sách, ngày nhập (ngày, tháng, năm), đơn giá, số lượng, nhà xuất bản, tình trạng (mới, cũ)
Nếu tình trạng mới: thành tiền = số lượng * đơn giá
Nếu tình trạng cũ: thành tiền = số lượng * đơn giá * 50%
- Sách tham khảo: mã sách, tên sách, ngày nhập (ngày, tháng, năm), đơn giá, số lượng, nhà xuất bản, thuế
Thành tiền = số lượng * đơn giá + thuế

BÀI TẬP

Yêu cầu:

- Xây dựng các lớp với chức năng thừa kế
- Nhập sách mới
- Tính tổng thành tiền cho tất cả sách và cho từng loại sách
- Xuất danh sách sách giáo khoa/sách tham khảo/ tất cả

BÀI TẬP

6. Xây dựng chương trình quản lý danh sách các loại giao dịch. Hệ thống quản lý có 2 loại giao dịch:

- Giao dịch vàng: mã giao dịch, ngày giao dịch (ngày, tháng, năm), đơn giá, số lượng, loại vàng. Thành tiền = số lượng * đơn giá
- Giao dịch tiền tệ: mã giao dịch, ngày giao dịch (ngày, tháng, năm), đơn giá, số lượng, tỉ giá, loại tiền tệ (tiền Việt Nam, tiền USD, tiền Euro). Thành tiền được tính như sau:
 - Nếu USD hoặc Euro: thành tiền = số lượng * đơn giá * tỉ giá
 - Nếu VN: thành tiền = số lượng * đơn giá

BÀI TẬP

- Yêu cầu:

- Xây dựng các lớp với chức năng kế thừa
- Nhập xuất danh sách các loại giao dịch
- Tính tổng số lượng cho từng loại giao dịch
- Tính trung bình thành tiền của giao dịch tiền tệ
- Xuất ra các giao dịch có đơn giá > 1 tỷ