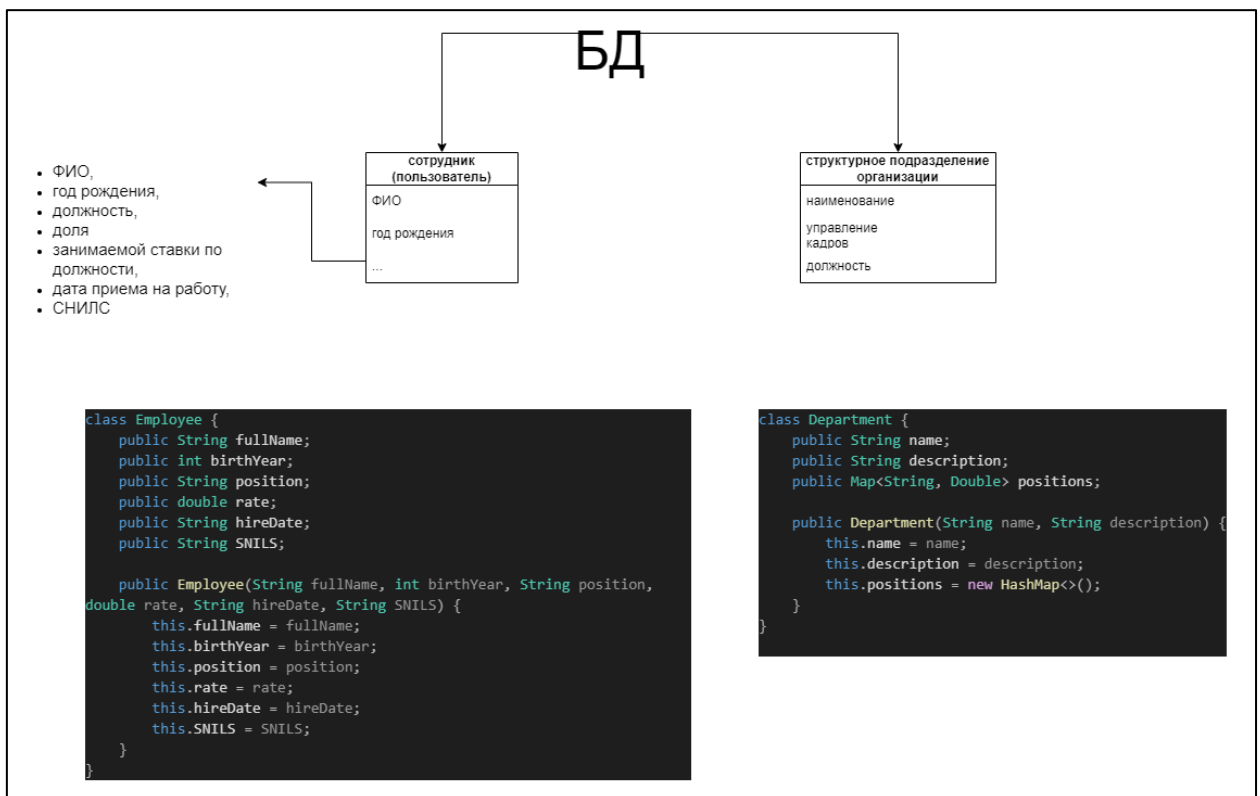


Дополнение

Как устроена программа? Для этого упростим программу до 6-х функций:

- ввод данных о новом сотруднике (полных данных или частичных: ФИО, год рождения и СНИЛС);
- поиск и вывод полных данных о сотруднике (сотрудниках) (поиск по ФИО, поиск по СНИЛС);
- ввод данных о новом структурном подразделении (полностью или частично);
- поиск по названию и удаление подразделения из БД (сотрудники этого подразделения не удаляются, остаются в БД за штатом).
- вывод списка всех сотрудников (краткая информация: ФИО, год рождения, СНИЛС);
- вывод списка всех подразделения (названия);

Схема БД:



КОД:

```
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.Iterator;
```

```

class Department {
    public String name;
    public String description;
    public String positions2;

    public Department(String name, String description, String positions) {
        this.name = name;
        this.description = description;
        this.positions2 = positions;
    }
}

class Employee {
    public String fullName;
    public int birthYear;
    public String position;
    public double rate;
    public String hireDate;
    public String SNILS;

    public Employee(String fullName, int birthYear, String position, double rate,
String hireDate, String SNILS) {
        this.fullName = fullName;
        this.birthYear = birthYear;
        this.position = position;
        this.rate = rate;
        this.hireDate = hireDate;
        this.SNILS = SNILS;
    }
}

class DatabaseManagement {
    private List<Employee> employees;
    private List<Department> departments;

    public DatabaseManagement() {
        employees = new ArrayList<>();
        departments = new ArrayList<>();
    }

    public void addEmployee(Employee employee) throws IllegalArgumentException {
        employees.add(employee);
        System.out.println("сотрудник '" + employee.fullName + "' успешно
добавлено.");
    }

    public void addDepartment(Department department) throws
IllegalArgumentException {
        departments.add(department);
    }
}

```

```

public void searchEmployeeByFullName_DOPOLNENIE(String fullName) {
    for (Employee employee : employees) {
        if (employee.fullName.equals(fullName)) {
            System.out.println("Сотрудник найден: " + employee.fullName);
            System.out.println("position:\n" + employee.position);

            for (Department department : departments) {
                if (department.positions2.equals(employee.position)) {
                    System.out.println("Отдел найден:\n" + department.name);
                    // return;
                }
            }
            return;
        }
    }
    System.out.println("Сотрудник с полным именем '" + fullName + "' не найден.");
}

public void searchEmployeeByFullName(String fullName) {
    for (Employee employee : employees) {
        if (employee.fullName.equals(fullName)) {
            System.out.println("Сотрудник найден:\n" + employee.fullName);
            return;
        }
    }
    System.out.println("Сотрудник с полным именем '" + fullName + "' не найден.");
}

public void searchEmployeeBySnils(String snils) {
    for (Employee employee : employees) {
        if (employee.SNILS.equals(snils)) {
            System.out.println("Сотрудник найден:\n" + employee.fullName);
            return;
        }
    }
    System.out.println("Сотрудник со СНИЛС '" + snils + "' не найден.");
}

public void addDepartment(String departmentName, String departmentName2,
String positions) {
    departments.add(new Department(departmentName, departmentName2,
positions));
    System.out.println("Отделение '" + departmentName + "' успешно
добавлено.");
}

public void searchDepartmentByName(String departmentName) {
    for (Department department : departments) {
        if (department.name.equals(departmentName)) {

```

```

        System.out.println("Отдел найден:\n" + department);
        return;
    }
}
System.out.println("Отдел с названием '" + departmentName + "' не
найден.");
}

public void deleteDepartmentByName(String departmentName) {
    Iterator<Department> iterator = departments.iterator();
    while (iterator.hasNext()) {
        Department department = iterator.next();
        if (department.name.equals(departmentName)) {
            iterator.remove();
            System.out.println("Отделение '" + departmentName + "' успешно
удалено.");
            return;
        }
    }
    System.out.println("Отдел с названием '" + departmentName + "' не
найден.");
}

public void printAllDepartments() {
    System.out.println("\n* Список всех отделов:");
    for (Department department : departments) {
        System.out.println("name: " + department.name);
        System.out.println("description: " + department.description);
        System.out.println("positions: " + department.positions2);
    }
}

public void printEmployees() {
    System.out.println("\nСотрудники:\n");
    for (Employee employee : employees) {
        System.out.println("* Данные *");
        System.out.println("ФИО сотрудника: " + employee.fullName);
        System.out.println("Год рождения: " + employee.birthYear);
        System.out.println("Введите СНИЛС: " + employee.SNILS);
        System.out.println("Введите Должность: " + employee.position);
        System.out.println("Дата приема на работу: " + employee.hireDate);
        System.out.println("Ставка: " + employee.rate);
    }
}

public static void main(String[] args) {
    DatabaseManagement db = new DatabaseManagement();

    String fullName = "KULIKOVA ALYONA VLADIMIROVNA";
    int birthYear = 2000;
    String ssn = "123-456-789 00";

```

```

String position = "PROGRAMMER";
String _hireDate = "01/01/2020";
int rate = 1;
String namedescription = "OTDEL KADROV";
String description = "OTDEL KADROV MGTU";

// ввод данных о новом сотруднике (полных данных или частичных: ФИО, год
// рождения и СНИЛС)
db.addEmployee(new Employee(fullName, birthYear, position, rate,
_hireDate, ssn));
db.addEmployee(new Employee(fullName + "2", birthYear, position, rate,
_hireDate, ssn));
db.addEmployee(new Employee(fullName + "3", birthYear, position, rate,
_hireDate, ssn));
System.out.println("\n");
db.printEmployees();
System.out.println("\n");
// поиск и вывод полных данных о сотруднике (сотрудниках) (поиск по ФИО,
поиск
// по СНИЛС)
db.searchEmployeeByFullName(fullName);
System.out.println("\n");
// вывод списка всех сотрудников (краткая информация: ФИО, год рождения,
СНИЛС)
db.searchEmployeeBySnils(ssn);
System.out.println("\n");
// ввод данных о новом структурном подразделении (полностью или частично)
db.addDepartment(namedescription, description, position);
db.addDepartment(namedescription + "1", description + "3", position +
"2");
db.addDepartment(namedescription + "2", description + "22", position);
db.addDepartment(namedescription + "3", description, position + "3");
System.out.println("\n");
// вывод списка всех подразделения (названия);
db.printAllDepartments();
System.out.println("\n");
// *****
db.searchEmployeeByFullName_DOPOLNENIE(fullName);
System.out.println("\n");
// поиск по названию и удаление подразделения из БД (сотрудники этого
// подразделения не удаляются, остаются в БД за штатом)
db.deleteDepartmentByName(namedescription);
}
}

```

Результат:

```
PS C:\javasing\lab4> c::; cd 'c:\javasing\lab4'; & 'C:\Program
'-agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=
ew' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\U
rkspaceStorage\47ed0770e7826ddd73c1489ec19f7f56\redhat.java\jdt
Management'
сотрудник 'KULIKOVA ALYONA VLADIMIROVNA' успешно добавлено.
сотрудник 'KULIKOVA ALYONA VLADIMIROVNA2' успешно добавлено.
сотрудник 'KULIKOVA ALYONA VLADIMIROVNA3' успешно добавлено.
```

Сотрудники:

* Данные *

ФИО сотрудника: KULIKOVA ALYONA VLADIMIROVNA

Год рождения: 2000

Введите СНИЛС: 123-456-789 00

Введите Должность: PROGRAMMER

Дата приема на работу: 01/01/2020

Ставка: 1.0

* Данные *

ФИО сотрудника: KULIKOVA ALYONA VLADIMIROVNA2

Год рождения: 2000

Введите СНИЛС: 123-456-789 00

Введите Должность: PROGRAMMER

Дата приема на работу: 01/01/2020

Ставка: 1.0

* Данные *

ФИО сотрудника: KULIKOVA ALYONA VLADIMIROVNA3

Год рождения: 2000

Введите СНИЛС: 123-456-789 00

Введите Должность: PROGRAMMER

Дата приема на работу: 01/01/2020

Ставка: 1.0

Сотрудник найден:
KULIKOVA ALYONA VLADIMIROVNA

Сотрудник найден:
KULIKOVA ALYONA VLADIMIROVNA

Отделение 'OTDEL KADROV' успешно добавлено.
Отделение 'OTDEL KADROV1' успешно добавлено.
Отделение 'OTDEL KADROV2' успешно добавлено.
Отделение 'OTDEL KADROV3' успешно добавлено.

* Список всех отделов:
name: OTDEL KADROV
description: OTDEL KADROV MGTU
positions: PROGRAMMER
name: OTDEL KADROV1
description: OTDEL KADROV MGTU3
positions: PROGRAMMER2
name: OTDEL KADROV2
description: OTDEL KADROV MGTU22
positions: PROGRAMMER
name: OTDEL KADROV3
description: OTDEL KADROV MGTU
positions: PROGRAMMER3

Сотрудник найден: KULIKOVA ALYONA VLADIMIROVNA
position:
PROGRAMMER
Отдел найден:
OTDEL KADROV
Отдел найден:
OTDEL KADROV2

Отделение 'OTDEL KADROV' успешно удалено.
PS C:\javasing\lab4> █

Дополненная функция по связке двух БД:

```
public void searchEmployeeByFullName_DOPOLNENIE(String fullName) {  
    for (Employee employee : employees) {  
        if (employee.fullName.equals(fullName)) {  
            System.out.println("Сотрудник найден: " + employee.fullName);  
            System.out.println("position:\n" + employee.position);  
  
            for (Department department : departments) {  
                if (department.positions2.equals(employee.position)) {  
                    System.out.println("Отдел найден:\n" + department.name);  
                    // return;  
                }  
            }  
            return;  
        }  
    }  
    System.out.println("Сотрудник с полным именем '" + fullName + "' не  
найдено.");  
}
```

РЕЗУЛЬТАТ:

```
Сотрудник найден: KULIKOVA ALYONA VLADIMIROVNA  
position:  
PROGRAMMER  
Отдел найден:  
OTDEL KADROV  
Отдел найден:  
OTDEL KADROV2
```