Exercise 1 (Optional): Constructor

Question 1:

Tạo constructor cho department:

a) Không có parameters

b) Có 1 parameter là nameDepartment và default id của Department = 0

Khởi tạo 1 Object với mỗi constructor ở trên

```
	public Department() {

	}

	public Department(String name) {
		this.id = 0;
		this.name = name;
	}
```

- Question 1:

```
	private static void question1() {
		Department dep = new Department();
		Department dep1 = new Department("Dep1");

	}
```

Question 2:

Tạo constructor cho Account:

a) Không có parameters

```
public Account() {
	}
```

b) Có các parameter là id, Email, Username, FirstName, LastName (với FullName = FirstName + LastName)

```
	public Account(int id, String email, String userName, String fullName) {
		this.id = id;
		this.email = email;
		this.userName = userName;
		this.fullName = fullName;
	}
```

c) Có các parameter là id, Email, Username, FirstName, LastName (với FullName = FirstName + LastName) và Position của User, default createDate = now

```
public Account(int id, String email, String userName, String fullName, Position
position) {
		this.id = id;
		this.email = email;
		this.userName = userName;
		this.fullName = fullName;
		this.position = position;
		this.createDate = LocalDate.now();
	}
```

d) Có các parameter là id, Email, Username, FirstName,

Created By DaoNQ- VTI Academy

LastName (với FullName = FirstName + LastName) và
Position của User, createDate

```
public Account(int id, String email, String userName, String fullName, Position
position, LocalDate createDate) {
		super();
		this.id = id;
		this.email = email;
		this.userName = userName;
		this.fullName = fullName;
		this.position = position;
		this.createDate = createDate;
	}
```

Khởi tạo 1 Object với mỗi constructor ở trên

```
private static void question2() {
		Account acc1 = new Account();
		Account acc2 = new Account(2, "email2", "username2", "fullname2");
		Position pos3 = new Position();
		Account acc3 = new Account(3, "email3", "username3", "fullname3",
pos3);
		System.out.println(acc3.createDate);

		Position pos4 = new Position();
		Account acc4 = new Account(3, "email3", "username3", "fullname3",
pos4, LocalDate.of(2021, 03, 17));
	}
```

Question 3:

a) không có parameters

```
public Group() {
	}
```

b) Có các parameter là GroupName, Creator, array Account[]
accounts, CreateDate

```
public Group(int id, String name, Account creator, LocalDate createDate,
Account[] accounts) {
		this.id = id;
		this.name = name;
		this.creator = creator;
		this.createDate = createDate;
		this.accounts = accounts;
	}
```

c) Có các parameter là GroupName, Creator, array String[]
usernames , CreateDate
Với mỗi username thì sẽ khởi tạo 1 Account (chỉ có thông tin
username, các thông tin còn lại = null).
Khởi tạo 1 Object với mỗi constructor ở trên ➔ Đoạn này chưa hiểu, chưa có lời giải, đây là đáp án

```
public Group(int id, String name, Account creator, String[] usernames, LocalDate createDate) {
		this.id = id;
		this.name = name;
		this.creator = creator;
```

Created By DaoNQ- VTI Academy

```
                // accounts
                Account[] accounts = new Account[usernames.length];
                for (int i = 0; i < usernames.length; i++) {
                        accounts[i] = new Account(usernames[i]);
                }
                this.createDate = createDate;
        }
```

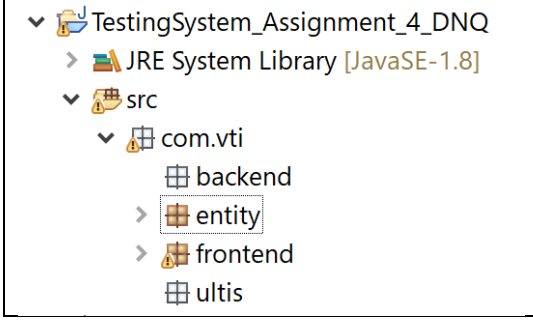
Exercise 2: Package

Question 1:

Tạo các package có tên như sau:

- com.vti.entity
- com.vti.frontend
- com.vti.backend

Thực hiện tạo trên giao diện

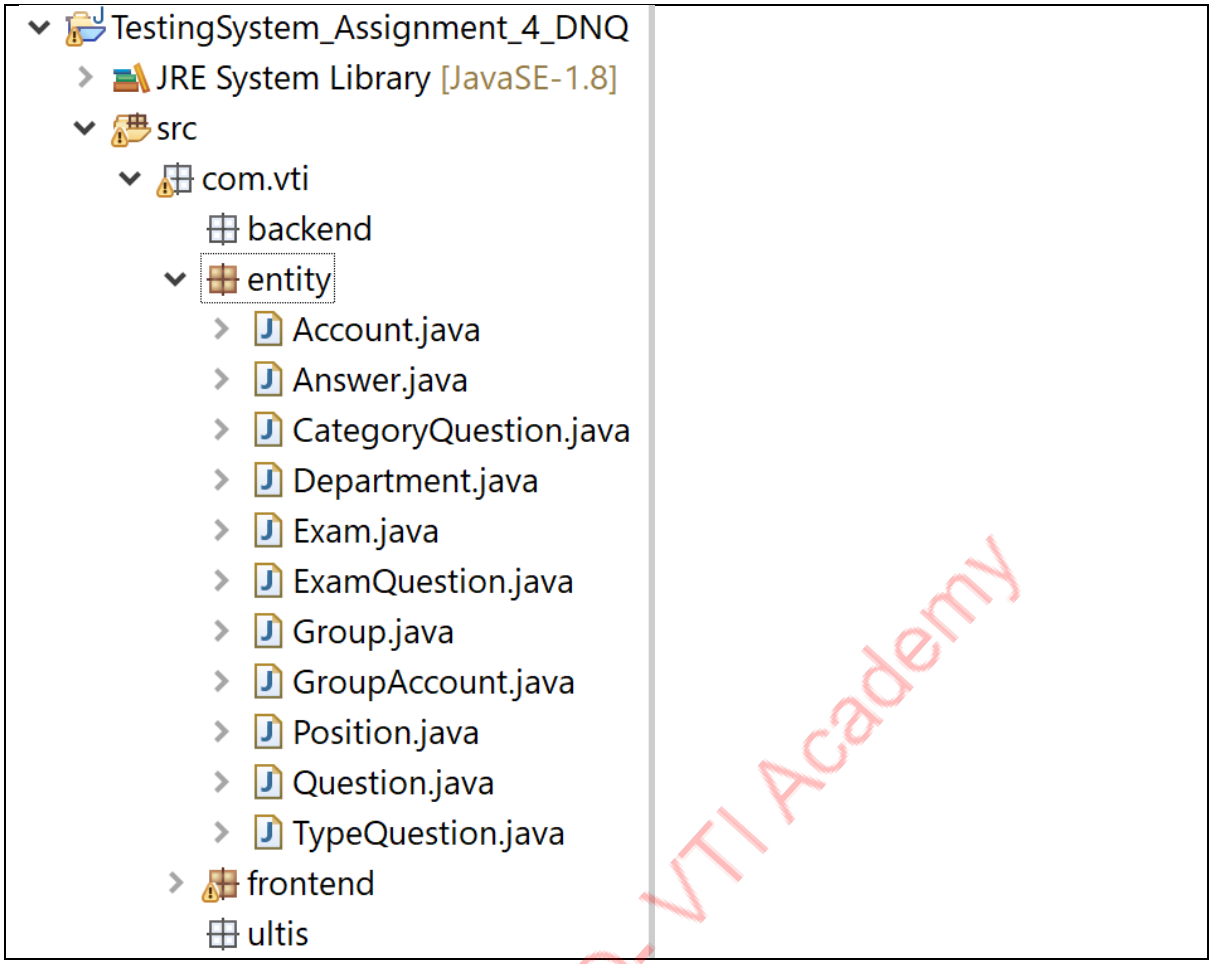


Question 2:

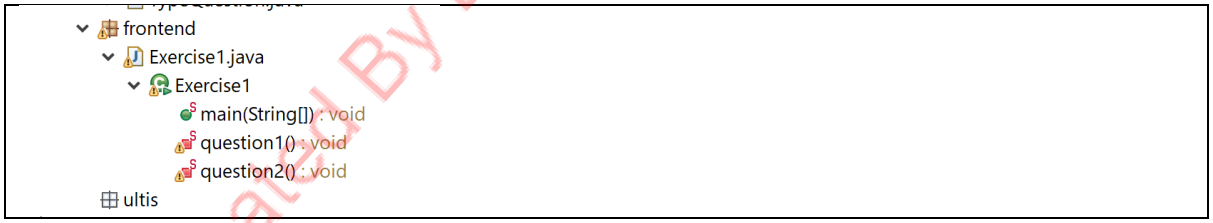
Trong package entity ta sẽ copy các Object đã tạo từ bài hôm trước như: Department, Account, Group, …

Thực hiện copy trên giao diện

Created By DaoNQ- VTI Academy



Trong phần backend tạo các class Excercise1, Excercise2, Excercise3, … Mỗi method trong mỗi Exercise là 1 question.
Trong front-end tạo các class Program1, Program2, Program3, … để demo kết quả của các Excercise1, Excercise2, Excercise3, …



Created By DaoNQ - VTI Academy

Exercise 3: Access Modifier

Question 1: private access modifier
Thay đổi access modifier và tạo getter/ setter của những class trong package entity, frontend, backend cho phù hợp
Gợi ý:
- Các class trong package entity thì để access modifier của property là private và tạo getter, setter cho từng property.
- Các method là các question ở package back-end sẽ để là public để các class ở frontend có thể gọi được (không để static).

=>Tạo 1 class Account

```java
public int getId() {
		return id;
	}

	public void setId(int id) {
		this.id = id;
	}

	public String getEmail() {
		return email;
	}

	public void setEmail(String email) {
		this.email = email;
	}

	public String getUserName() {
		return userName;
	}

	public void setUserName(String userName) {
		this.userName = userName;
	}

	public String getFullName() {
		return fullName;
	}

	public void setFullName(String fullName) {
		this.fullName = fullName;
	}

	public Department getDepartment() {
		return department;
	}

	public void setDepartment(Department department) {
		this.department = department;
	}

	public Position getPosition() {
		return position;
	}

	public void setPosition(Position position) {
		this.position = position;
	}

	public LocalDate getCreateDate() {
		return createDate;
	}

	public void setCreateDate(LocalDate createDate) {
		this.createDate = createDate;
	}

	public Group[] getGroups() {
		return groups;
	}
```

Created By DaoNQ- VTI Academy

```
	public void setGroups(Group[] groups) {
		this.groups = groups;
	}
```

➔ Tạo các Question dạng Public:

```
package com.vti.frontend;

import java.time.LocalDate;

import com.vti.entity.Account;
import com.vti.entity.Department;
import com.vti.entity.Position;

public class Exercise1 {

	public static void main(String[] args) {
		Exercise1 ex1 = new Exercise1();
		ex1.question1();
		ex1.question2();
	}

	public void question2() {
		Account acc1 = new Account();
		Account acc2 = new Account(2, "email2", "username2", "fullname2");
		Position pos3 = new Position();
		Account acc3 = new Account(3, "email3", "username3", "fullname3",
pos3);
		System.out.println(acc3.getCreateDate());

		Position pos4 = new Position();
		Account acc4 = new Account(3, "email3", "username3", "fullname3",
pos4, LocalDate.of(2021, 03, 17));
	}

	public void question1() {
		Department dep = new Department();
		Department dep1 = new Department("Dep1");

	}

}
```

Created By DaoNQ - VTI Academy

Exercise 4 : Encapsulation

Question 1:

Tạo Object Student có các property id, name, hometown, điểm học lực

a) Tất cả các property sẽ để là private để các class khác không chỉnh sửa hay nhìn thấy

Tạo Class Student trong tab Entity

```
package com.vti.entity;

public class Student {
	private int id;
```

```
	private String name;
	private String hometown;
	private Float score;

}
```

b) Tạo constructor cho phép khi khởi tạo mỗi student thì người dùng sẽ nhập vào tên, hometown và có điểm học lực = 0

```
		public Student(String name, String hometown) {
			this.name = name;
			this.hometown = hometown;
			this.score = (float) 0;
		}
```

c) Tạo 1 method cho phép set điểm vào

```
	public void setScore(Float score) {
		this.score = score;
	}
```

d) Tạo 1 method cho phép cộng thêm điểm

```
	public void plusScore(Float score) {
		this.score = this.score + score;
	}
```

e) Tạo 1 method để in ra thông tin của sinh viên bao gồm có tên, điểm học lực (nếu điểm <4.0 thì sẽ in ra là Yếu, nếu điểm > 4.0 và < 6.0 thì sẽ in ra là trung bình, nếu điểm > 6.0 và < 8.0 thì sẽ in ra là khá, nếu > 8.0 thì in ra là Giỏi)

```
	@Override
	public String toString() {
		String rank = null;
		if (this.score < 4.0) {
			rank = "Yếu";
		} else if (this.score < 6.0) {
			rank = "Trung Bình";
		} else if (this.score < 8.0) {
			rank = "Khá";
		}else {
			rank = "Giỏi";
		}

		return "Student [id=" + id + ", name=" + name + ", hometown=" +
hometown + ", score=" + score + ", Xếp loại=" + rank + "]";
	}
```

Demo các chức năng trên bằng class ở front-end.

Tạo Class Encapsulation trong package để Demo

```
	package com.vti.backend;

import com.vti.entity.Student;
```

Created By DaoNQ - VTI Academy

```java
public class Exercise4 {
	public void printStudent() {
		Student st1 = new Student("name1", "H1");
		st1.plusScore(1f);
		Student st2 = new Student("name2", "H2");
		st2.plusScore(5f);
		Student st3 = new Student("name3", "H3");
		st3.plusScore(9f);


		System.out.println(st1);
		System.out.println(st2);
		System.out.println(st3);
	}
}
```

Tạo Class EncapsulationProgram trong package để Demo

```java
package com.vti.frontend;

import com.vti.backend.Exercise4;

public class Demo_Exercise4 {

	public static void main(String[] args) {
		Exercise4 ex4 = new Exercise4();
		ex4.printStudent();

	}

}
```

Finally Output:

```
Picked up _JAVA_OPTIONS: -Djava.net.preferIPv4Stack=true
Student [id=0, name=name1, hometown=H1, score=1.0, Xếp loại=Yếu]
Student [id=0, name=name2, hometown=H2, score=5.0, Xếp loại=Trung Bình]
Student [id=0, name=name3, hometown=H3, score=9.0, Xếp loại=Giỏi]
```

Question 2 (Optional):

Tạo class phù hợp cho thiết kế sau:

Circle	Account	Date
- radius: double = 1.0 - color: String = 'red'	- id: String, - name: String, - balance: int	- day: int - month: int - year: int
+ Circle() + Circle(radius: double) + Circle(radius: double, color: String) + getRadius(): double + setRadius(radius: double) + getColor(): String + setColor(color: String) + getRadius(): double + getArea(): double + toString(): String	+ Account(id: String, name: String, balance: int) + getID(): String + getName: String, + getBalance(): int + credit(amount: int) + debit(amount: int) + tranferTo(account: Account, int amount)	+ Date(int day, int month, int year) + getDay(): int + getMonth(): int + getYear(): int + setDay(day: int) + setMonth(month: int) + setYear(year: int) + toString(): String + isLeapYear(): boolean

Class Circle

```java
package com.vti.entity;
```

Created By DaoNQ- VTI Academy

```java
public class Circle {
	private Double radius;
	private String color;

	public static final Double PI = 3.14;

	public Circle() {
	}

	public Circle(Double radius) {
		this.radius = radius;
	}

	public Circle(Double radius, String color) {
		this.radius = radius;
		this.color = color;
	}

	public Double getRadius() {
		return radius;
	}

	public void setRadius(Double radius) {
		this.radius = radius;
	}

	public String getColor() {
		return color;
	}

	public void setColor(String color) {
		this.color = color;
	}

	public Double getArea() {
		return PI * Math.pow(this.radius, 2.0);
	}

	@Override
	public String toString() {
		return "Circle [radius=" + radius + ", color=" + color + "]";
	}

}
```

Created By DaoNQ- VTI Academy

Class Account

```java
package com.vti.entity;

public class Account_Ques2 {
	private String id;
	private String name;
	private int balance;

	public Account_Ques2(String id, String name, int balance) {
		this.id = id;
		this.name = name;
		this.balance = balance;
```

```
	}

	public String getId() {
		return id;
	}

	public String getName() {
		return name;
	}

	public int getBalance() {
		return balance;
	}

	public int credit(int amount) {
		return this.balance += amount;
	}

	public int debit(int amount) {
		return this.balance -= amount;
	}

	public void tranfer(Account_Ques2 acc, int amount) {
		this.balance -= amount;
		acc.balance += amount;
	}

}
```

Class Date

```
	package com.vti.entity;

public class Date {
	private int day;
	private int month;
	private int year;

	public Date(int day, int month, int year) {
		this.day = day;
		this.month = month;
		this.year = year;
	}

	public int getDay() {
		return day;
	}

	public int getMonth() {
		return month;
	}

	public int getYear() {
		return year;
	}

	public void setDay(int day) {
		this.day = day;
	}
```

Created By DaoNQ- VTI Academy

```
	public void setMonth(int month) {
		this.month = month;
	}

	public void setYear(int year) {
		this.year = year;
	}

	public boolean isLeapYear() {

		year = this.year;
		boolean isLeap = false;

		if (year % 4 == 0) // chia hết cho 4 là năm nhuận
		{
			if (year % 100 == 0)
			// nếu vừa chia hết cho 4 mà vừa chia hết cho 100 thì k phải
là năm nhuận
			{
				if (year % 400 == 0)// chia hết cho 400 là năm nhuận
					return true;
				else
					return false;// không chia hết cho 400 thì
không phải năm nhuận
			} else// chia hết cho 4 nhưng không chia hết cho 100 là năm
nhuận
				return true;
		} else {
			return false;
		}

	}

	@Override
	public String toString() {
		return "Date [day=" + day + ", month=" + month + ", year=" + year
+ "]";
	}

}
```

Chương trình demo:

```
package com.vti.backend;

import com.vti.entity.Account_Ques2;
import com.vti.entity.Circle;
import com.vti.entity.Date;
import com.vti.entity.Student;

public class Exercise4 {
	public void question1() {
		Student st1 = new Student("name1", "H1");
		st1.plusScore(1f);
		Student st2 = new Student("name2", "H2");
		st2.plusScore(5f);
```

Created By DaoNQ- VTI Academy

```
            Student st3 = new Student("name3", "H3");
            st3.plusScore(9f);

            System.out.println(st1);
            System.out.println(st2);
            System.out.println(st3);
      }

      public void question2() {
            System.out.println("---Demo Circle ----");
            Circle circle = new Circle(2.0, "red");
            System.out.println("Diện tích hình tròn là: " + circle.getArea());

            System.out.println("---- Demo AccountQues2 -----");
            Account_Ques2 accQues1 = new Account_Ques2("1", "accQues1", 10);
            Account_Ques2 accQues2 = new Account_Ques2("2", "accQues2", 20);

            System.out.println(
                        "Số tiền ban đầu của: accQues1: " +
accQues1.getBalance() + " accQues2: " + accQues2.getBalance());

            accQues1.credit(50);
            System.out.println("Balace của accQues1 sau khi Credit 50: " +
accQues1.getBalance());

            accQues1.debit(20);
            System.out.println("Balace của accQues1 sau khi debit 20: " +
accQues1.getBalance());

            System.out.println("accQues1 chuyển 20 cho accQues2: ");
            accQues1.tranfer(accQues2, 20);

            System.out.println("Sau khi chuyển số tiền của accQues1: " +
accQues1.getBalance() + " accQues2: "
                        + accQues2.getBalance());

            System.out.println("---- Demo Date -----");
            Date date1 = new Date(25, 04, 1988);
            System.out.println("Bạn vừa tạo ngày: " + date1);

            System.out.println("Check năm nhuận: ");
            if (date1.isLeapYear()) {
                  System.out.println("Đây là năm nhuận");
            } else {
                  System.out.println("Đây không phải là năm nhuận");
            }
      }
}
```

Created By DatNQ - VTI Academy

Exercise 5: Inheritance

Chú ý: áp dụng cả encapsulation cho các question ở dưới

Question 1: inheritance

Một đơn vị sản xuất gồm có các cán bộ là công nhân, kỹ sư, nhân viên.

Mỗi cán bộ cần quản lý các dữ liệu: Họ tên, tuổi, giới tính(name, nữ, khác), địa chỉ.

Cấp công nhân sẽ có thêm các thuộc tính riêng: Bậc (1 đến 10).

Cấp kỹ sư có thuộc tính riêng: Nghành đào tạo.

Các nhân viên có thuộc tính riêng: công việc.

Hãy xây dựng các lớp CongNhan, KySu, NhanVien kế thừa từ lớp CanBo.

Class Staff

```
package com.vti.entity;

public class Staff {

	private String name;
	private int age;
	private Gender gender;
	private String address;

	public enum Gender {
		MALE, FEMALE, UNKNOWN

	}

	public Staff(String name, int age, Gender gender, String address) {
		this.name = name;
		this.age = age;
		this.gender = gender;
		this.address = address;
	}


	public String getName() {
		return name;
	}


	@Override
	public String toString() {
		return "Staff [name=" + name + ", age=" + age + ", gender=" +
gender + ", address=" + address + "]";
	}

}
```

Class Worker

```
package com.vti.entity;

public class Worker extends Staff {
	private int rank;
```

Created By DaoNQ- VTI Academy

```
	public Worker(String name, int age, Gender gender, String address, int
rank) {
		super(name, age, gender, address);
		this.rank = rank;
	}

	@Override
	public String toString() {

		return super.toString() + "Position: Worker [rank=" + rank + "]";
	}

}
```

Class Engineer

```
package com.vti.entity;

public class Engineer extends Staff {
	private String specialized;

	public Engineer(String name, int age, Gender gender, String address,
String specialized) {
		super(name, age, gender, address);
		this.specialized = specialized;
	}

	@Override
	public String toString() {
		return super.toString() + "Position: Engineer [specialized=" +
specialized + "]";
	}


}
```

Class Employee

```
package com.vti.entity;

public class Employee extends Staff {
	private String task;

	public Employee(String name, int age, Gender gender, String address,
String task) {
		super(name, age, gender, address);
		this.task = task;
	}

	@Override
	public String toString() {
		return super.toString() + "Position: Employee [task=" + task +
"]";
	}

}
```

Created By DaoNQ - VTI Academy

Question 2: Tiếp tục Question 1
Xây dựng lớp QLCB(quản lý cán bộ) cài đặt các phương thức thực hiện các chức năng sau:

```
package com.vti.backend;

import java.util.ArrayList;
import java.util.Scanner;

import com.vti.entity.Engineer;
import com.vti.entity.Staff;
import com.vti.entity.Worker;
import com.vti.entity.Staff.Gender;

public class Exercise5 {
	private ArrayList<Staff> staffList;
	private Scanner sc;

	public Exercise5() {
		staffList = new ArrayList<Staff>();
		sc = new Scanner(System.in);
	}

	public void question1() {

	}

	public void question2() {
		menu();
	}

	public void menu() {

//		ArrayList<Staff> staffList = null;
		while (true) {

	System.out.println("==========================================================
================");
			System.out.println("=================Lựa chọn chức năng bạn
muốn sử dụng==================");
			System.out.println("===              1. Thêm mới cán bộ.
===");
			System.out.println("===              2. Tìm kiếm theo họ
tên.                    ===");
			System.out.println("===              3. Hiện thị thông tin
về danh sách các cán bộ.    ===");
			System.out.println("===              4. Nhập vào tên của cán
bộ và delete cán bộ đó    ===");
			System.out.println("===              5. Thoát khỏi chương
trình.                  ===");

	System.out.println("==========================================================
================");
			int menuChoose = sc.nextInt();
			switch (menuChoose) {
			case 1:
				addStaff();
				break;
```

Created By DaoNQ- VTI Academy

```
                        case 2:
                                findByName();
                                break;
                        case 3:
                                printListStaff();
                                break;
                        case 4:
                                deleteByName();
                                break;
                        case 5:
                                return;

                        default:
                                System.out.println("Lựa chọn đúng số trên menu");
                                break;
                        }
                }

        }

        private void deleteByName() {
                System.out.println("Nhập tên cần xóa thông tin: ");
                String delName = sc.next();
                staffList.removeIf(staff -> staff.getName().equals(delName));
                printListStaff();
//              for (Staff staff : staffList) {
//                      if (staff.getName().equals(delName)) {
//                              staffList.remove(staff);
//                      }
//              }  => không thể sử dụng vòng lặp này để xóa do dữ liệu sau khi xóa
không tường minh, nên k chạy tiếp được vòng lặp để kiểm tra sau khi xóa
        }

        private void findByName() {
                System.out.println("Nhập vào tên muốn tìm kiếm: ");
                String findName = sc.next();
                for (Staff staff : staffList) {
                        if (staff.getName().equals(findName)) {
                                System.out.println(staff);
                        }
                }

        }

        private void printListStaff() {
                for (Staff staff : staffList) {
                        System.out.println(staff);
                }

        }

        private void addStaff() {
                System.out.println("----------------------------------------------
------");
                System.out.println("--------Lựa chọn chức năng bạn muốn sử dụng---
------");
                System.out.println("---     1. Thêm Engineer
---");
```

Created By ĐạoNC - VTI Academy

```
            System.out.println("---     2. Thêm Worker
---");
            System.out.println("---     3. Thêm Employeee
---");
            System.out.println("-----------------------------------------
------");
            int choose1 = sc.nextInt();
            switch (choose1) {
            case 1:
                  System.out.println("Nhập vào tên Engineer: ");
                  String nameEngineer = sc.next();
                  System.out.println("Nhập vào tuổi Engineer: ");
                  int ageEngineer = sc.nextInt();
                  System.out.println("Nhập vào giới tính Engineer 1.Male,
2.Female, 3.Unknown: ");
                  int flagEngineer = sc.nextInt();
                  Gender genderName1 = null;
                  switch (flagEngineer) {
                  case 1:
                        genderName1 = Gender.MALE;
                        break;
                  case 2:
                        genderName1 = Gender.FEMALE;
                        break;
                  case 3:
                        genderName1 = Gender.UNKNOWN;
                        break;
                  }
                  System.out.println("Nhập vào địa chỉ Engineer: ");
                  String addEngineer = sc.next();
                  System.out.println("Nhập vào chuyên ngành Engineer: ");
                  String specializedEngineer = sc.next();
                  Staff engineer = new Engineer(nameEngineer, ageEngineer,
genderName1, addEngineer, specializedEngineer);
                  staffList.add(engineer);
                  break;
            case 2:
                  System.out.println("Nhập vào tên Worker: ");
                  String nameWorker = sc.next();
                  System.out.println("Nhập vào tuổi Worker: ");
                  int ageWorker = sc.nextInt();
                  System.out.println("Nhập vào giới tính Worker 1.Male,
2.Female, 3.Unknown: ");
                  int flagGender = sc.nextInt();
                  Gender genderName = null;
                  switch (flagGender) {
                  case 1:
                        genderName = Gender.MALE;
                        break;
                  case 2:
                        genderName = Gender.FEMALE;
                        break;
                  case 3:
                        genderName = Gender.UNKNOWN;
                        break;
                  }
                  System.out.println("Nhập vào địa chỉ Worker: ");
                  String addWorker = sc.next();
                  System.out.println("Nhập vào bậc Worker: ");
```

Created By Dang Ngọc VTI Academy

```
                    int rankWorker = sc.nextInt();
                    Staff worker1 = new Worker(nameWorker, ageWorker,
genderName, addWorker, rankWorker);
                    staffList.add(worker1);
                    break;
             case 3:
                    System.out.println("Nhập vào tên Employee: ");
                    String nameEmployee = sc.next();
                    System.out.println("Nhập vào tuổi Employee: ");
                    int ageEmployee = sc.nextInt();
                    System.out.println("Nhập vào giới tính Employee 1.Male,
2.Female, 3.Unknown: ");
                    int flagEmployee = sc.nextInt();
                    Gender genderEmployee = null;
                    switch (flagEmployee) {
                    case 1:
                           genderEmployee = Gender.MALE;
                           break;
                    case 2:
                           genderEmployee = Gender.FEMALE;
                           break;
                    case 3:
                           genderEmployee = Gender.UNKNOWN;
                           break;
                    }
                    System.out.println("Nhập vào địa chỉ Employee: ");
                    String addEmployee = sc.next();
                    System.out.println("Nhập vào nhiệm vụ Employee: ");
                    String taskEmployee = sc.next();
                    Staff Employee = new com.vti.entity.Employee(nameEmployee,
ageEmployee, genderEmployee, addEmployee,
                                  taskEmployee);
                    staffList.add(Employee);
                    break;
             default:
                    break;
             }

      }
}
```

a) Thêm mới cán bộ.

```
             System.out.println("-----------------------------------------------
------");
             System.out.println("--------Lựa chọn chức năng bạn muốn sử dụng---
------");
             System.out.println("---     1. Thêm Engineer
---");
             System.out.println("---     2. Thêm Worker
---");
             System.out.println("---     3. Thêm Employeee
---");
             System.out.println("-----------------------------------------------
------");
             int choose1 = sc.nextInt();
             switch (choose1) {
```

Created By Daonq - VTI Academy

```
            case 1:
                  System.out.println("Nhập vào tên Engineer: ");
                  String nameEngineer = sc.next();
                  System.out.println("Nhập vào tuổi Engineer: ");
                  int ageEngineer = sc.nextInt();
                  System.out.println("Nhập vào giới tính Engineer 1.Male,
2.Female, 3.Unknown: ");
                  int flagEngineer = sc.nextInt();
                  Gender genderName1 = null;
                  switch (flagEngineer) {
                  case 1:
                        genderName1 = Gender.MALE;
                        break;
                  case 2:
                        genderName1 = Gender.FEMALE;
                        break;
                  case 3:
                        genderName1 = Gender.UNKNOWN;
                        break;
                  }
                  System.out.println("Nhập vào địa chỉ Engineer: ");
                  String addEngineer = sc.next();
                  System.out.println("Nhập vào chuyên ngành Engineer: ");
                  String specializedEngineer = sc.next();
                  Staff engineer = new Engineer(nameEngineer, ageEngineer,
genderName1, addEngineer, specializedEngineer);
                  staffList.add(engineer);
                  break;
            case 2:
                  System.out.println("Nhập vào tên Worker: ");
                  String nameWorker = sc.next();
                  System.out.println("Nhập vào tuổi Worker: ");
                  int ageWorker = sc.nextInt();
                  System.out.println("Nhập vào giới tính Worker 1.Male,
2.Female, 3.Unknown: ");
                  int flagGender = sc.nextInt();
                  Gender genderName = null;
                  switch (flagGender) {
                  case 1:
                        genderName = Gender.MALE;
                        break;
                  case 2:
                        genderName = Gender.FEMALE;
                        break;
                  case 3:
                        genderName = Gender.UNKNOWN;
                        break;
                  }
                  System.out.println("Nhập vào địa chỉ Worker: ");
                  String addWorker = sc.next();
                  System.out.println("Nhập vào bậc Worker: ");
                  int rankWorker = sc.nextInt();
                  Staff worker1 = new Worker(nameWorker, ageWorker,
genderName, addWorker, rankWorker);
                  staffList.add(worker1);
                  break;
            case 3:
                  System.out.println("Nhập vào tên Employee: ");
                  String nameEmployee = sc.next();
```

Created By Dao Ngọc VTI Academy

```
                    System.out.println("Nhập vào tuổi Employee: ");
                    int ageEmployee = sc.nextInt();
                    System.out.println("Nhập vào giới tính Employee 1.Male,
2.Female, 3.Unknown: ");
                    int flagEmployee = sc.nextInt();
                    Gender genderEmployee = null;
                    switch (flagEmployee) {
                    case 1:
                           genderEmployee = Gender.MALE;
                           break;
                    case 2:
                           genderEmployee = Gender.FEMALE;
                           break;
                    case 3:
                           genderEmployee = Gender.UNKNOWN;
                           break;
                    }
                    System.out.println("Nhập vào địa chỉ Employee: ");
                    String addEmployee = sc.next();
                    System.out.println("Nhập vào nhiệm vụ Employee: ");
                    String taskEmployee = sc.next();
                    Staff Employee = new com.vti.entity.Employee(nameEmployee,
ageEmployee, genderEmployee, addEmployee,
                                  taskEmployee);
                    staffList.add(Employee);
                    break;
             default:
                    break;
             }

      }
```

b) Tìm kiếm theo họ tên.

```
private void findByName() {
             System.out.println("Nhập vào tên muốn tìm kiếm: ");
             String findName = sc.next();
             for (Staff staff : staffList) {
                    if (staff.getName().equals(findName)) {
                           System.out.println(staff);
                    }
             }

      }
```

c) Hiện thị thông tin về danh sách các cán bộ.

```
      private void printListStaff() {
             for (Staff staff : staffList) {
                    System.out.println(staff);
             }

      }
```

d) Nhập vào tên của cán bộ và delete cán bộ đó

```
private void deleteByName() {
```

Created By TraoNQ- VTI Academy

```
                System.out.println("Nhập tên cần xóa thông tin: ");
                String delName = sc.next();
                staffList.removeIf(staff -> staff.getName().equals(delName));
                printListStaff();
//              for (Staff staff : staffList) {
//                      if (staff.getName().equals(delName)) {
//                              staffList.remove(staff);
//                      }
//              }  => không thể sử dụng vòng lặp này để xóa do dữ liệu sau khi xóa không tường minh, nên k chạy tiếp được vòng lặp để kiểm tra sau khi xóa
        }
```

e) Thoát khỏi chương trình.

Sử return trong while True

Question 3 (Optional): constructor inheritance

Tạo class abstract Person gồm các property name và tạo constructor có 1 parameter name

```
package com.vti.entity.inheritance.question3;

public abstract class Person {

        private String name;

        public Person(String name) {
                this.name = name;
        }

        public String getName() {
                return name;
        }

        public void setName(String name) {
                this.name = name;
        }
}
```

Tạo class abstract Student kế thừa Person gồm các property id, name và tạo constructor có 2 parameter id, name

```
package com.vti.entity.inheritance.question3;

public abstract class Student extends Person {
        private int id;

        public Student(String name, int id) {
                super(name);
                this.id = id;
        }

        public int getId() {
                return id;
        }

        public void setId(int id) {
                this.id = id;
        }
}
```

Tạo class HighSchoolStudent kế thừa Student bao gồm các property id, name, clazz (Lớp đang học), desiredUniversity (trường đại học mong muốn vào) và tạo constructor có 4 parameter id, name, clazz,

Created By DaoNQ- VTI Academy

desiredUniversity.
Hãy khởi tạo Object HighSchoolStudent với các giá trị: id = 1, name = "Nam", clazz = "Chuyên Văn", desiredUniversity = "Đại học công nghệ"

```
package com.vti.entity.inheritance.question3;

public class HighSchoolStudent extends Student {

	private String clazz;
	private String desiredUniversity;

	public HighSchoolStudent(String name, int id, String clazz, String desiredUniversity) {
		super(name, id);
		this.clazz = clazz;
		this.desiredUniversity = desiredUniversity;
	}

	@Override
	public String toString() {
		return "HighSchoolStudent{" + "id='" + super.getId() + '\'' + "name='" +
super.getName() + '\'' + "clazz='"
				+ clazz + '\'' + ", desiredUniversity='" + desiredUniversity + '\''
+ '}';
	}
}
```

```
	public void question3() {
		HighSchoolStudent highSchoolStudent = new HighSchoolStudent("Nam", 1, "Chuyên Văn",
"Đại học công nghệ");
		System.out.println(highSchoolStudent);
	}
```

Question 4 (Optional):
Một thư viện cần quản lý các tài liệu bao gồm Sách, Tạp chí, Báo. Mỗi tài liệu gồm có các thuộc tính sau: Mã tài liệu(Mã tài liệu là duy nhất), Tên nhà xuất bản, số bản phát hành.
Tạo class Document

```
package com.vti.entity.ex4_ques4;

public class Document {
	private int id;
	private String publisher;
	private int numRelease;

	public Document(int id, String publisher, int numRelease) {
		this.id = id;
		this.publisher = publisher;
		this.numRelease = numRelease;
	}

	public int getId() {
		return id;
	}

	@Override
	public String toString() {
		return "Document [id=" + id + ", publisher=" + publisher + ",
numRelease=" + numRelease + "]";
	}
```

Created By Daohq- VTI Academy

```
}
```

Các loại sách cần quản lý thêm các thuộc tính: tên tác giả, số trang.

Tạo Class Book

```
package com.vti.entity.ex4_ques4;

public class Book extends Document {
	private String authorName;
	private int numPage;
	public Book(int id, String publisher, int numRelease, String authorName,
int numPage) {
		super(id, publisher, numRelease);
		this.authorName = authorName;
		this.numPage = numPage;
	}
	@Override
	public String toString() {
		return super.toString() + " Category: Book [authorName=" +
authorName + ", numPage=" + numPage + "]";
	}

}
```

Các tạp chí cần quản lý thêm: Số phát hành, tháng phát hành.

Tạo Class Magazine

```
package com.vti.entity.ex4_ques4;

import java.time.LocalDate;

public class Magazine extends Document {
	private int idRelease;
	private LocalDate monthRelease;
	public Magazine(int id, String publisher, int numRelease, int idRelease,
LocalDate monthRelease) {
		super(id, publisher, numRelease);
		this.idRelease = idRelease;
		this.monthRelease = monthRelease;
	}
	@Override
	public String toString() {
		return super.toString() + " Category: Magazine [idRelease=" +
idRelease + ", monthRelease=" + monthRelease + "]";
	}

}
```

Các báo cần quản lý thêm: Ngày phát hành.

Tạo Class Article

Created by DaoNQ - VTI Academy

```java
package com.vti.entity.ex4_ques4;

import java.time.LocalDate;

public class Article extends Document {
	private LocalDate dayRelease;

	public Article(int id, String publisher, int numRelease, LocalDate
dayRelease) {
		super(id, publisher, numRelease);
		this.dayRelease = dayRelease;
	}

	@Override
	public String toString() {
		return super.toString() + " Category: Article [dayRelease=" +
dayRelease + "]";
	}

}
```

Xây dựng chương trình để quản lý tài liệu (QLTV) cho thư viện một cách hiệu quả.
Xây dựng lớp QuanLySach có các chức năng sau

```java
package com.vti.backend;

import java.time.LocalDate;
import java.util.ArrayList;
import java.util.Scanner;

import com.vti.entity.Engineer;
import com.vti.entity.Staff;
import com.vti.entity.Worker;
import com.vti.entity.ex4_ques4.Article;
import com.vti.entity.ex4_ques4.Book;
import com.vti.entity.ex4_ques4.Document;
import com.vti.entity.ex4_ques4.Magazine;
import com.vti.entity.Staff.Gender;

public class Exercise5_Ques4 {
	public static int ID = 0;
	private ArrayList<Document> documentList;
	private Scanner sc;

	public Exercise5_Ques4() {
		sc = new Scanner(System.in);
		documentList = new ArrayList<Document>();
	}

	public void question4() {
		loadMenu();
	}

	public void loadMenu() {
		while (true) {
```

Created By DaoNQ- VTI Academy

```
        System.out.println("==========================================================
================");
                        System.out.println("================Lựa chọn chức năng bạn
muốn sử dụng==================");
                        System.out.println("===              1. Thêm mới tài liệu.
===");
                        System.out.println("===              2. Xóa tài liệu.
===");
                        System.out.println("===              3. Hiện thị thông tin
tài liệu.                    ===");
                        System.out.println("===              4. Tìm kiếm tài liệu
theo loại                                ===");
                        System.out.println("===              5. Thoát khỏi chương
trình.                       ===");

        System.out.println("==========================================================
================");
                        int menuChoose = sc.nextInt();
                        switch (menuChoose) {
                        case 1:
                                addDocument();
                                break;

                        case 2:
                                delDocument();
                                break;
                        case 3:
                                printListDocument();
                                break;
                        case 4:
                                findByCategory();
                                break;
                        case 5:
                                return;

                        default:
                                System.out.println("Alarm: Lựa chọn đúng số trên
menu");
                                break;
                        }
                }
        }

        private void findByCategory() {
                System.out.println("Nhập vào loại tài liệu cần tìm kiếm 1.Sách,
2.Báo, 3.Tạp chí: ");
                int chooseCategory = sc.nextInt();
                switch (chooseCategory) {
                case 1:
                        for (Document document : documentList) {
                                if (document instanceof Book) {
                                        System.out.println(document);
                                }
                        }
                        break;
                case 2:
                        for (Document document : documentList) {
                                if (document instanceof Article) {
```

Created By DaoNQ- VTI Academy

```
                                System.out.println(document);
                        }
                    }
                    break;
                case 3:
                    for (Document document : documentList) {
                        if (document instanceof Magazine) {
                            System.out.println(document);
                        }
                    }
                    break;
                default:
                    System.out.println("Nhập không đúng.!!");
                    break;
                }
            }

            private void printListDocument() {
                for (Document document : documentList) {
                    System.out.println(document);
                }

            }

            private void delDocument() {
                System.out.println("Nhập vào mã tài liệu muốn xóa: ");
                int idDel = sc.nextInt();
                Boolean flagCheckExists = false;
                for (Document document : documentList) {
                    if (document.getId() == idDel) {
                        flagCheckExists = true;
                    }
                }
                if (flagCheckExists) {
                    documentList.removeIf(doc -> doc.getId() == idDel);
                    for (Document document : documentList) {
                        System.out.println("Xóa tài liệu thành công, danh
sách tài liệu còn lại:");
                        System.out.println(document);
                    }
                } else {
                    System.out.println("Mã tài liệu bạn nhập không có trên hệ
thống.");
                }

            }

            private void addDocument() {

                System.out.println("----------------------------------------------
------");
                System.out.println("--------Lựa chọn chức năng bạn muốn sử dụng---
------");
                System.out.println("---     1. Thêm Sách
---");
                System.out.println("---     2. Thêm Báo
---");
                System.out.println("---     3. Thêm Tạp chí
---");
```

Created By DaoNQ - VTI Academy

```
            System.out.println("-----------------------------------------------
------");
            int choose1 = sc.nextInt();
            switch (choose1) {
            case 1:
                  System.out.println("Nhập vào tên nhà xuất bản: ");
                  String publisher = sc.next();
                  System.out.println("Nhập vào số bản phát hành: ");
                  int numRelease = sc.nextInt();
                  System.out.println("Nhập vào tên tác giả: ");
                  String authorName = sc.next();
                  System.out.println("Nhập vào số trang: ");
                  int numPage = sc.nextInt();
                  ID++;
                  Document book = new Book(ID, publisher, numRelease,
authorName, numPage);
                  documentList.add(book);
                  break;
            case 2:
                  System.out.println("Nhập vào tên nhà xuất bản: ");
                  String publisherArticle = sc.next();
                  System.out.println("Nhập vào số bản phát hành: ");
                  int numReleaseArticle = sc.nextInt();
                  System.out.println("Nhập vào ngày phát hành: ");
                  int day = sc.nextInt();
                  System.out.println("Nhập vào tháng phát hành: ");
                  int month = sc.nextInt();
                  System.out.println("Nhập vào năm phát hành: ");
                  int year = sc.nextInt();
                  LocalDate dayReleaseArticle = LocalDate.of(year, month,
day);
                  ID++;
                  Document artice = new Article(ID, publisherArticle,
numReleaseArticle, dayReleaseArticle);
                  documentList.add(artice);
                  break;

            case 3:
                  System.out.println("Nhập vào tên nhà xuất bản: ");
                  String publisherMagazine = sc.next();
                  System.out.println("Nhập vào số bản phát hành: ");
                  int numReleaseMagazine = sc.nextInt();
                  System.out.println("Nhập vào số phát hành: ");
                  int idReleaseMagazine = sc.nextInt();
                  System.out.println("Nhập vào ngày phát hành: ");
                  int day1 = sc.nextInt();
                  System.out.println("Nhập vào tháng phát hành: ");
                  int month1 = sc.nextInt();
                  System.out.println("Nhập vào năm phát hành: ");
                  int year1 = sc.nextInt();
                  LocalDate monthReleaseMagazine = LocalDate.of(year1, month1,
day1);
                  ID++;
                  Document magazine = new Magazine(ID, publisherMagazine,
numReleaseMagazine, idReleaseMagazine,
                              monthReleaseMagazine);
                  documentList.add(magazine);

                  break;
```

Created By DatNQ-VTI Academy

```
            default:
                  break;
            }

      }
}
```

a) Thêm mới tài liêu: Sách, tạp chí, báo.

```
private void addDocument() {

            System.out.println("-----------------------------------------------
------");
            System.out.println("--------Lựa chọn chức năng bạn muốn sử dụng---
------");
            System.out.println("---     1. Thêm Sách
---");
            System.out.println("---     2. Thêm Báo
---");
            System.out.println("---     3. Thêm Tạp chí
---");
            System.out.println("-----------------------------------------------
------");
            int choose1 = sc.nextInt();
            switch (choose1) {
            case 1:
                  System.out.println("Nhập vào tên nhà xuất bản: ");
                  String publisher = sc.next();
                  System.out.println("Nhập vào số bản phát hành: ");
                  int numRelease = sc.nextInt();
                  System.out.println("Nhập vào tên tác giả: ");
                  String authorName = sc.next();
                  System.out.println("Nhập vào số trang: ");
                  int numPage = sc.nextInt();
                  ID++;
                  Document book = new Book(ID, publisher, numRelease,
authorName, numPage);
                  documentList.add(book);
                  break;
            case 2:
                  System.out.println("Nhập vào tên nhà xuất bản: ");
                  String publisherArticle = sc.next();
                  System.out.println("Nhập vào số bản phát hành: ");
                  int numReleaseArticle = sc.nextInt();
                  System.out.println("Nhập vào ngày phát hành: ");
                  int day = sc.nextInt();
                  System.out.println("Nhập vào tháng phát hành: ");
                  int month = sc.nextInt();
                  System.out.println("Nhập vào năm phát hành: ");
                  int year = sc.nextInt();
                  LocalDate dayReleaseArticle = LocalDate.of(year, month,
day);
                  ID++;
                  Document artice = new Article(ID, publisherArticle,
numReleaseArticle, dayReleaseArticle);
                  documentList.add(artice);
                  break;
```

Created By Dao Duc - VTI Academy

```
            case 3:
                  System.out.println("Nhập vào tên nhà xuất bản: ");
                  String publisherMagazine = sc.next();
                  System.out.println("Nhập vào số bản phát hành: ");
                  int numReleaseMagazine = sc.nextInt();
                  System.out.println("Nhập vào số phát hành: ");
                  int idReleaseMagazine = sc.nextInt();
                  System.out.println("Nhập vào ngày phát hành: ");
                  int day1 = sc.nextInt();
                  System.out.println("Nhập vào tháng phát hành: ");
                  int month1 = sc.nextInt();
                  System.out.println("Nhập vào năm phát hành: ");
                  int year1 = sc.nextInt();
                  LocalDate monthReleaseMagazine = LocalDate.of(year1, month1,
day1);
                  ID++;
                  Document magazine = new Magazine(ID, publisherMagazine,
numReleaseMagazine, idReleaseMagazine,
                              monthReleaseMagazine);
                  documentList.add(magazine);

                  break;
            default:
                  break;
            }

      }
```

b) Xoá tài liệu theo mã tài liệu.

```
      private void delDocument() {
            System.out.println("Nhập vào mã tài liệu muốn xóa: ");
            int idDel = sc.nextInt();
            Boolean flagCheckExists = false;
            for (Document document : documentList) {
                  if (document.getId() == idDel) {
                        flagCheckExists = true;
                  }
            }
            if (flagCheckExists) {
                  documentList.removeIf(doc -> doc.getId() == idDel);
                  for (Document document : documentList) {
                        System.out.println("Xóa tài liệu thành công, danh
sách tài liệu còn lại:");
                        System.out.println(document);
                  }
            } else {
                  System.out.println("Mã tài liệu bạn nhập không có trên hệ
thống.");
            }

      }
```

c) Hiện thị thông tin về tài liệu.

```
private void printListDocument() {
            for (Document document : documentList) {
                  System.out.println(document);
            }
```

Created By Đạo Q- VTI Academy

```
	}
```

d) Tìm kiếm tài liệu theo loại: Sách, tạp chí, báo.

```
private void findByCategory() {
		System.out.println("Nhập vào loại tài liệu cần tìm kiếm 1.Sách,
2.Báo, 3.Tạp chí: ");
		int chooseCategory = sc.nextInt();
		switch (chooseCategory) {
		case 1:
			for (Document document : documentList) {
				if (document instanceof Book) {
					System.out.println(document);
				}
			}
			break;
		case 2:
			for (Document document : documentList) {
				if (document instanceof Article) {
					System.out.println(document);
				}
			}
			break;
		case 3:
			for (Document document : documentList) {
				if (document instanceof Magazine) {
					System.out.println(document);
				}
			}
			break;
		default:
			System.out.println("Nhập không đúng.!!");
			break;
		}
	}

	private void printListDocument() {
		for (Document document : documentList) {
			System.out.println(document);
		}

	}
```

e) Thoát khỏi chương trình.

```
Sử dụng lệnh return để thoát khỏi vòng true
```

Created By DaoHQ- VTI Academy

Exercise 6: Abstraction

Question 1: abstract class & abstract method

Tạo 1 class Phone để lưu thông tin liên lạc, trong Phone có chứa 1 property có kiểu dữ liệu là Array contacts, trong từng contact có lưu thông tin number, name

a) Tạo các abstract method

Tạo class Phone:

```java
package com.vti.entity.ex6;

public abstract class Phone {

	public abstract void insertContact(String name, String phone);

	public abstract void removeContact(String name);

	public abstract void updateContact(String name, String newPhone);

	public abstract void searchContact(String name);

}
```

Tạo Class Contact:

```java
package com.vti.entity.ex6;

public class Contact {
	private String name;
	private String phone;

	public Contact(String name, String phone) {
		this.name = name;
		this.phone = phone;
	}

	public String getName() {
		return name;
	}

	public void setName(String name) {
		this.name = name;
	}

	public String getPhone() {
		return phone;
	}

	public void setPhone(String phone) {
		this.phone = phone;
	}

	@Override
	public String toString() {
		return "Contact [name=" + name + ", phone=" + phone + "]";
	}


}
```

Created By DaoNQ- VTI Academy

Tạo class VietnamPhone

```java
package com.vti.entity.ex6;

import java.util.ArrayList;

public class VietnamesePhone extends Phone {
```

```
	private ArrayList<Contact> contacts;

	public VietnamesePhone() {
		contacts = new ArrayList<Contact>();
	}

	@Override
	public void insertContact(String name, String phone) {
		Contact contact = new Contact(name, phone);
		contacts.add(contact);
	}

	@Override
	public void removeContact(String name) {
		contacts.removeIf(contact -> contact.getName().equals(name));
	}

	@Override
	public void updateContact(String name, String newPhone) {
		for (Contact contact : contacts) {
			if (contact.getName().equals(name)) {
				contact.setPhone(newPhone);
			}
		}

	}

	@Override
	public void searchContact(String name) {
		for (Contact contact : contacts) {
			if (contact.getName().equals(name)) {
				System.out.println(contact);
			}
		}

	}

	public void printContact() {
		for (Contact contact : contacts) {
			System.out.println(contact);
		}
	}

}
```

Tạo Class Exercise6 trong backend

```
package com.vti.backend;

import java.util.Scanner;

import com.vti.entity.ex6.Phone;
import com.vti.entity.ex6.VietnamesePhone;

public class Exercise6 {
	Scanner sc;
```

Created By DaoNQ - VTI Academy

```
	public Exercise6() {
		sc = new Scanner(System.in);
	}

	public void question1() {
		loadmenuAbstract();

	}

	private void loadmenuAbstract() {
		VietnamesePhone vnPhone = new VietnamesePhone();
		while (true) {

	System.out.println("==========================================================
================");
			System.out.println("=================Lựa chọn chức năng bạn
muốn sử dụng==================");
			System.out.println("===              1. InsertContact.
===");
			System.out.println("===              2. RemoveContact.
===");
			System.out.println("===              3. UpdateContact.
				===");
			System.out.println("===                 4. SearchContact
				                ===");
			System.out.println("===                 5. ShowContact
				            ===");
			System.out.println("===              6. Thoát khỏi chương
trình.                      ===");

	System.out.println("==========================================================
================");
			int menuChoose = sc.nextInt();
			switch (menuChoose) {
			case 1:
				System.out.println("Nhập vào tên Contact: ");
				String name = sc.next();
				System.out.println("Nhập vào tên số Phone: ");
				String phone = sc.next();
				vnPhone.insertContact(name, phone);
				break;
			case 2:
				System.out.println("Nhập vào tên Contact cần remove:
");
				String removeName = sc.next();
				vnPhone.removeContact(removeName);
				break;
			case 3:
				System.out.println("Nhập tên cần Update: ");
				String name1 = sc.next();
				System.out.println("Nhập số Phone mới: ");
				String newPhone = sc.next();
				vnPhone.updateContact(name1, newPhone);
				System.out.println("Kết quả: ");
				vnPhone.searchContact(name1);
				break;
			case 4:
				System.out.println("Nhập vào tên Contact cần tìm
kiếm: ");
```

Created By DaoNQ - VTI Academy

```
                        String searchName = sc.next();
                        vnPhone.searchContact(searchName);
                        break;
                case 5:
                        vnPhone.printContact();
                        break;
                case 6:
                        return;

                default:
                        System.out.println("Alarm: Lựa chọn đúng số trên
menu");
                        break;
                }
        }

    }
}
```

a. void insertContact(String name, String phone)

```
    @Override
    public void insertContact(String name, String phone) {
        Contact contact = new Contact(name, phone);
        contacts.add(contact);
    }
```

b. void removeContact(String name)

```
    @Override
    public void removeContact(String name) {
        contacts.removeIf(contact -> contact.getName().equals(name));
    }
```

c. void updateContact(String name, String newPhone)

```
@Override
    public void updateContact(String name, String newPhone) {
        for (Contact contact : contacts) {
            if (contact.getName().equals(name)) {
                contact.setPhone(newPhone);
            }
        }

    }
```

d. void searchContact(String name)

```
    @Override
    public void searchContact(String name) {
        for (Contact contact : contacts) {
            if (contact.getName().equals(name)) {
                System.out.println(contact);
            }
        }

    }
```

Created By DaoNQ- VTI Academy

b) Tạo class VietnamesePhone kế thừa Phone và triển khai các method abstract
Viết chương trình demo

```
package com.vti.frontend;

import com.vti.backend.Exercise6;

public class Demo_Exercise6 {
	public static void main(String[] args) {
		Exercise6 ex6 = new Exercise6();
		ex6.question1();
	}
}
```

Question 2 (Optional):
Hãy xây dựng 1 class User với các thuộc tính name kiểu String và salary ratio kiểu double.
a) Viết các hàm cho phép nhập và trả về name, salary ratio.
b) Viết hàm calculatePay() dạng abstract trả về thu nhập của nhân viên, kiểu double.
c) Viết hàm displayInfor()
Tạo Class User:

```
package com.vti.entity.ex6Ques2_3;

public abstract class User {
	private String name;

	protected void name() {

	}

	Double salaryRatio;

	public User(String name, Double salaryRatio) {
		this.name = name;
		this.salaryRatio = salaryRatio;
	}

	public abstract Double calculatePay();

	public void displayInfor() {
		System.out.println("Name: " + name);
		System.out.println("Ration Salary: " + salaryRatio);
		System.out.println("Lương chi trả: " + calculatePay());
	}
}
```

Tạo Class Employee:

```
package com.vti.entity.ex6Ques2_3;
```

Created By DaoNQ- VTI Academy

```java
public class Employee extends User {

	public Employee(String name, Double salaryRatio) {
		super(name, salaryRatio);
	}

	@Override
	public Double calculatePay() {
		return salaryRatio * 420;
	}

}
```

Tạo Class Manager:

```java
package com.vti.entity.ex6Ques2_3;

public class Manager extends User {

	public Manager(String name, Double salaryRatio) {
		super(name, salaryRatio);

	}

	@Override
	public Double calculatePay() {
		return salaryRatio * 520;
	}

}
```

Tạo Class Waiter:

```java
package com.vti.entity.ex6Ques2_3;

public class Waiter extends User {

	public Waiter(String name, Double salaryRatio) {
		super(name, salaryRatio);
	}

	@Override
	public Double calculatePay() {
		return salaryRatio * 220;
	}

}
```

Created By DaoNQ- VTI Academy

Question 3 (Optional): Tiếp tục Question 2
Viết class Employee, Manager, Waiter kế thừa User như sau:
Implement method calculatePay() như sau:
Đối với Employee sẽ được tính = salary ratio * 420
Đối với Manager sẽ được tính = salary ratio * 520
Đối với Waiter sẽ được tính = salary ratio * 220

```
public void question23() {
             while (true) {

      System.out.println("==========================================================
=================");
                    System.out.println("=================Lựa chọn chức năng bạn
muốn sử dụng==================");
                    System.out.println("===              1. Thêm Employee.
===");
                    System.out.println("===              2. Thêm Manager.
===");
                    System.out.println("===              3. Thêm Waiter .
===");
                    System.out.println("===              4. Thoát khỏi chương
trình.                    ===");

      System.out.println("==========================================================
=================");
                    int menuChoose = sc.nextInt();
                    switch (menuChoose) {
                    case 1:
                           System.out.println("Nhập vào tên Employee: ");
                           String emName = sc.next();
                           System.out.println("Nhập vào SalaryRatio: ");
                           Double emSalaryRatio = sc.nextDouble();
                           com.vti.entity.ex6Ques2_3.Employee em = new
com.vti.entity.ex6Ques2_3.Employee(emName, emSalaryRatio);
                           em.displayInfor();
                           break;

                    case 2:
                           System.out.println("Nhập vào tên Manager: ");
                           String managerName = sc.next();
                           System.out.println("Nhập vào SalaryRatio: ");
                           Double managerSalaryRatio = sc.nextDouble();
                           Manager manager = new Manager(managerName,
managerSalaryRatio);
                           manager.displayInfor();
                           break;
                    case 3:
                           System.out.println("Nhập vào tên Waiter: ");
                           String waiterName = sc.next();
                           System.out.println("Nhập vào SalaryRatio: ");
                           Double waiterSalaryRatio = sc.nextDouble();
                           Waiter waiter = new Waiter(waiterName,
waiterSalaryRatio);
                           waiter.displayInfor();
                           break;
                    case 4:
                           return;
```

Created By Đạo NQ, VTI Academy

```
                    default:
                        System.out.println("Alarm: Lựa chọn đúng số trên
menu");
                        break;
                    }
                }
            }
```

Created By DaoNQ- VTI Academy