

Chú ý: Tạo 1 Project đặt tên là "TestingSystem_Assignment_5", tách mỗi exercise thành 1 class riêng, mỗi Question tách thành 1 method

Exercise 1: Abstraction

Question 1: Interface

Tạo một interface có tên **INews** bao gồm method **void Display()**, **float Calculate()**.

```
package com.vti.entity;

public interface INews {
    public void display();

    public float calculate();
}
```

Tạo một class News bao gồm thuộc tính:

ID (int), Title (String), PublishDate (String), Author (String), Content (String) và AverageRate (float).

Tạo các setter và **getter** cho từng thuộc tính, riêng AverageRate thì chỉ có getter.

```
package com.vti.entity;

public class News implements INews {
    private int id;
    private String title;
    private String publishDate;
    private String author;
    private String content;
    private Float averageRate;
    private int[] rate;
    public static int COUNT = 0;

    public News(String title, String publishDate, String author, String content, int[] rate) {
        COUNT++;
        this.id = COUNT;
        this.title = title;
        this.publishDate = publishDate;
        this.author = author;
        this.content = content;
        this.averageRate = averageRate;
        this.rate = rate;
    }

    /**
     * @return the rate
     */
    public int[] getRate() {
        return rate;
    }

    /**
     * @param rate the rate to set
     */
}
```

```

    */
    public void setRate(int[] rate) {
        this.rate = rate;
    }

    public int getId() {
        return id;
    }

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

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }

    public String getPublishDate() {
        return publishDate;
    }

    public void setPublishDate(String publishDate) {
        this.publishDate = publishDate;
    }

    public String getAuthor() {
        return author;
    }

    public void setAuthor(String author) {
        this.author = author;
    }

    public String getContent() {
        return content;
    }

    public void setContent(String content) {
        this.content = content;
    }

    public Float getAverageRate() {
        return averageRate;
    }

    @Override
    public void display() {
        System.out.println("News [title=" + title + ", publishDate=" +
publishDate + ", author=" + author + "]);
    }

    @Override
    public float calculate() {
        averageRate = (float) ((rate[0] + rate[1] + rate[2]) / 3);
        return averageRate;
    }

```

```
}  
  
}
```

Implement các method trong interface INews như sau:

a) Method Display() sẽ in ra Title, PublishDate, Author, Content và AverageRate của tin tức ra console.

```
@Override  
public void display() {  
    System.out.println("News [title=" + title + ", publishDate=" +  
publishDate + ", author=" + author + "]);  
}
```

b) Method có tên Calculate() để thiết đặt thuộc tính Khai báo một array có tên Rates kiểu int gồm 3 phần tử AverageRate là trung bình cộng của 3 phần tử của array Rates.

```
@Override  
public float calculate() {  
    averageRate = (float) ((rate[0] + rate[1] + rate[2]) / 3);  
    return averageRate;  
}
```

c) Tạo chương trình demo có tên là MyNews và tạo một menu lựa chọn gồm các mục sau:

- Insert news
- View list news
- Average rate
- Exit

(→ Tạo class **Exercise1** để thao tác trên các list các đối tượng của class New đã tạo bên trên)

```
package com.vti.backend;  
  
import java.util.ArrayList;  
import java.util.Scanner;  
  
import com.vti.entity.News;  
  
public class Exercise1 {  
    private Scanner sc;  
    private ArrayList<News> listNews;  
  
    public Exercise1() {  
        sc = new Scanner(System.in);  
        listNews = new ArrayList<News>();  
    }  
  
    public void question1() {  
        loadMenu();  
    }  
}
```

```

        private void loadMenu() {
            while (true) {

                System.out.println("=====
=====");
                System.out.println("=====Lựa chọn chức năng bạn
muốn sử dụng=====");
                System.out.println("===          1. Insert news.
===");
                System.out.println("===          2. View list news.
===");
                System.out.println("===          3. Average rate .
===");
                System.out.println("===          4. Exit.
===");

                System.out.println("=====
=====");
                int menuChoose = sc.nextInt();
                switch (menuChoose) {
                    case 1:
                        System.out.println("Nhập vào Title");
                        String title = sc.next();
                        System.out.println("Nhập vào PublishDate");
                        String publishDate = sc.next();
                        System.out.println("Nhập vào Author");
                        String author = sc.next();
                        System.out.println("Nhập vào Content");
                        String content = sc.next();

                        System.out.println("Nhập vào đánh giá 1");
                        int rate1 = sc.nextInt();
                        System.out.println("Nhập vào đánh giá 2");
                        int rate2 = sc.nextInt();
                        System.out.println("Nhập vào đánh giá 3");
                        int rate3 = sc.nextInt();
                        int[] rates = { rate1, rate2, rate3 };
                        News news = new News(title, publishDate, author,
content, rates);

                        listNews.add(news);
                        break;

                    case 2:
                        for (News news1 : listNews) {
                            news1.display();
                        }
                        break;
                    case 3:
                        System.out.println("Đánh giá trung bình Rate");
                        for (News news1 : listNews) {

                            System.out.println("Title: " + news1.getTitle()
+ " RateAVG: " + news1.calculate());
                        }
                        break;
                    case 4:
                        return;
                }
            }
        }
    }
}

```

```

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

```

→ Tạo Class **Demo_Exercise1** để demo kết quả:

```

package com.vti.frontend;

import com.vti.backend.Exercise1;

public class Demo_Exercise1 {
    public static void main(String[] args) {
        Exercise1 ex1 = new Exercise1();
        ex1.question1();
    }
}

```

Nếu người dùng **chọn 1** từ bàn phím thì tạo một object của class News và nhập giá trị cho các thuộc tính Title, PublishDate, Author, Content sau đó yêu cầu người dùng nhập vào 3 đánh giá để lưu vào Rates.

Nếu người dùng **chọn 2** từ bàn phím thì thực thi method Display().

Nếu người dùng **chọn 3** từ bàn phím thì thực hiện method Calculate() để tính đánh giá trung bình, sau đó thực thi method Display().

Trường hợp người dùng **chọn 4** thì sẽ thoát khỏi chương trình.

Question 2 (Optional):

Các thí sinh dự thi đại học bao gồm các thí sinh thi khối **A, B, và khối C**. Các thí sinh cần quản lý các thông tin sau: **Số báo danh, họ tên, địa chỉ, mức ưu tiên.**

Thí sinh thi khối A thi các môn: **Toán, Lý, Hoá.**

Thí sinh thi khối B thi các môn: **Toán, Hoá, Sinh.**

Thí sinh thi khối C thi các môn: **Văn, Sử, Địa.**

a) Xây dựng các class để quản lý các thí sinh dự thi đại học.

→ Tạo Class Block khối thi:

```

package com.vti.entity;

public class Block {
    private String name;
    private String subject;
}

```

```

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

    /**
     * @return the name
     */
    public String getName() {
        return name;
    }

    public String getSubject() {
        switch (this.name) {
            case "A":
                this.subject = "Toán, Lý, Hóa";
                break;
            case "B":
                this.subject = "Toán, Hóa, Sinh";
                break;
            case "C":
                this.subject = "Văn, Sử, Địa";
                break;
        }
        return this.subject;
    }
}

```

=>Tạo **class Contestant**: lưu thông tin thí sinh

```

package com.vti.entity;

public class Contestant {
    public static int COUNT = 0;
    private int idNum;
    private String name;
    private String address;
    private int priority;
    private Block block;

    /**
     * @return the idNum
     */
    public int getIdNum() {
        return idNum;
    }

    public Contestant(String name, String address, int priority, Block block)
    {
        COUNT++;
        this.idNum = COUNT;
        this.name = name;
        this.address = address;
        this.priority = priority;
        this.block = block;
    }

    @Override

```

```

        public String toString() {
            return "Contestant [idNum=" + idNum + ", name=" + name + ",
address=" + address + ", priority=" + priority
                + ", block=" + block.getName() + ", Subject =" +
block.getSubject() + "];"
        }
    }
}

```

b) Xây dựng interface ITuyen và class TuyenSinh có các chức năng:

=> **Tạo Interface ItuyenSinh**

```

package com.vti.entity;

public interface ITuyen_Sinh {
    public void addContestant();

    public void showInfoContestant();

    public void findByIdNum(int id);
}

```

=> **Tạo class Tuyen_Sinh**

```

package com.vti.entity;

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

public class Tuyen_Sinh implements ITuyen_Sinh {
    private ArrayList<Contestant> listContestant;
    private Scanner sc;

    public Tuyen_Sinh() {
        listContestant = new ArrayList<Contestant>();
        sc = new Scanner(System.in);
    }

    @Override
    public void addContestant() {
        System.out.println("Nhập vào tên thí sinh:");
        String name = sc.next();
        System.out.println("Nhập vào địa chỉ thí sinh:");
        String address = sc.next();
        System.out.println("Nhập vào mức ưu tiên thí sinh:");
        int priority = sc.nextInt();
        System.out.println("Nhập vào khối thí sinh chọn các khối A, B, C:");

        String block = sc.next();
        Contestant contestant = new Contestant(name, address, priority,
new Block(block));
        listContestant.add(contestant);
    }

    @Override
    public void showInfoContestant() {

```

```

        for (Contestant contestant : listContestant) {
            System.out.println(contestant);
        }
    }

    @Override
    public void findByIdNum(int id) {
        for (Contestant contestant : listContestant) {
            if (contestant.getIdNum() == id) {
                System.out.println(contestant);
            }
        }
    }
}

```

a. Thêm mới thí sinh.

```

@ @Override
    public void addContestant() {
        System.out.println("Nhập vào tên thí sinh:");
        String name = sc.next();
        System.out.println("Nhập vào địa chỉ thí sinh:");
        String address = sc.next();
        System.out.println("Nhập vào mức ưu tiên thí sinh:");
        int priority = sc.nextInt();
        System.out.println("Nhập vào khối thí sinh chọn các khối A, B, C:");
        String block = sc.next();
        Contestant contestant = new Contestant(name, address, priority,
        new Block(block));
        listContestant.add(contestant);
    }

```

b. Hiện thị thông tin của thí sinh và khối thi của thí sinh.

```

@Override
    public void showInfoContestant() {
        for (Contestant contestant : listContestant) {
            System.out.println(contestant);
        }
    }

```

c. Tìm kiếm theo số báo danh.

```

@Override
    public void findByIdNum(int id) {
        for (Contestant contestant : listContestant) {
            if (contestant.getIdNum() == id) {
                System.out.println(contestant);
            }
        }
    }

```

d. Thoát khỏi chương trình.

=>Tạo chương trình chạy demo:

```
package com.vti.backend;

import java.util.Scanner;

import com.vti.entity.News;
import com.vti.entity.Tuyen_Sinh;

public class Exercise1_Ques2 {
    private Tuyen_Sinh tuyenSinh;
    private Scanner sc;

    public Exercise1_Ques2() {
        sc = new Scanner(System.in);
        tuyenSinh = new Tuyen_Sinh();
    }

    public void question2() {
        loadMenu();
    }

    private void loadMenu() {
        while (true) {

            System.out.println("=====
=====");
            System.out.println("=====Lựa chọn chức năng bạn
muốn sử dụng=====");
            System.out.println("===                1. Insert Contestant.
===");
            System.out.println("===                2. View Contestant.
===");
            System.out.println("===                3. Find by idNum
Contestant .            ===");
            System.out.println("===                4. Exit.
===");

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

                case 2:
                    tuyenSinh.showInfoContestant();
                    break;
                case 3:
                    System.out.println("Nhập vào ID cần tìm kiếm: ");
                    int id = sc.nextInt();
                    tuyenSinh.findByIdNum(id);
                    break;
                case 4:
                    return;

                default:
```

```

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

```

→ Main

```

package com.vti.frontend;

import com.vti.backend.Exercise1_Ques2;

public class Demo_Exercise2_Ques2 {
    public static void main(String[] args) {
        Exercise1_Ques2 ex1Ques2 = new Exercise1_Ques2();
        ex1Ques2.question2();
    }
}

```

Exercise 2: Polymorphism

Question 1: Interface Management

Tạo 1 class Student gồm các property id, name, group(int)

```

package com.vti.entity;

public class Student implements IStudent {
    public static int COUNT;
    private int id;
    private String name;
    private int group;

    @Override
    public String toString() {
        return "Student [id=" + id + ", name=" + name + ", group=" + group
+ " ]";
    }

    /**
     * @return the group
     */
    public int getGroup() {
        return group;
    }

    @Override
    public void diemDanh() {
        System.out.println(name + " điểm danh.");
    }

    @Override
    public void hocBai() {

```

```

        System.out.println(name + " đang học bài.");
    }

    @Override
    public void donVeSinh() {
        System.out.println(name + " đi dọn vệ sinh.");
    }

    public Student(String name, int group) {
        COUNT++;
        this.id = COUNT;
        this.name = name;
        this.group = group;
    }
}

```

Tạo 1 **interface IStudent** bao gồm các method : **điểmDanh()**, **họcBài()**, **đi dọn vệ sinh()**

```

package com.vti.entity;

public interface IStudent {
    public void diemDanh();

    public void hocBai();

    public void donVeSinh();
}

```

Class Student sẽ implement interface như sau:
Method điểm danh() sẽ in ra nội dung như sau:

```

@Override
    public void diemDanh() {
        System.out.println(name + " điểm danh.");
    }

```

"Nguyễn Văn A điểm danh"
 "Nguyễn Văn B điểm danh"
 "Nguyễn Văn C điểm danh"

Method học Bài () sẽ in ra nội dung như sau:

```

@Override
    public void hocBai() {
        System.out.println(name + " đang học bài.");
    }

```

"Nguyễn Văn A đang học bài"
 "Nguyễn Văn B đang học bài "

"Nguyễn Văn C đang học bài "

...

Tương tự với các method còn lại

```
@Override
public void donVeSinh() {
    System.out.println(name + " đi dọn vệ sinh.");
}
```

Hãy viết chương trình thực hiện các lệnh sau:

=> Tạo class Exercise2:

```
package com.vti.backend;

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

import com.vti.entity.Student;

public class Exercise2 {
    private ArrayList<Student> listStudent;
    private Random random;
    private Scanner sc;

    public Exercise2() {
        listStudent = new ArrayList<Student>();
        random = new Random();
        sc = new Scanner(System.in);
    }

    public void question1() {
        while (true) {

            System.out.println("=====
=====");
            System.out.println("=====Lựa chọn chức năng bạn
muốn sử dụng=====");
            System.out.println("===                               1. Tạo danh sách sinh
viên.                               ===");
            System.out.println("===                               2. Hiển thị ds sv trong
lớp.                               ===");
            System.out.println("===                               3. Điểm danh lớp
===");
            System.out.println("===                               4. Gọi nhóm 1 đi học
bài                               ===");
            System.out.println("===                               5. Gọi nhóm 2 đi dọn vệ
sinh                               ===");
            System.out.println("===                               6. Exit.
=====");

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

```

        initStudent();
        System.out.println("Tạo thành công 10 sinh viên.");
        break;

    case 2:
        for (Student student : listStudent) {
            System.out.println(student);
        }
        break;
    case 3:
        System.out.println("Cả lớp điểm danh: ");
        for (Student student : listStudent) {
            student.diemDanh();
        }
        break;
    case 4:
        System.out.println("Nhóm 1 đi học bài: ");
        for (Student student : listStudent) {
            if (student.getGroup() == 1) {
                student.hocBai();
            }
        }
        break;
    case 5:
        System.out.println("Nhóm 2 đi dọn vệ sinh: ");
        for (Student student : listStudent) {
            if (student.getGroup() == 2) {
                student.donVeSinh();
            }
        }
        break;
    case 6:
        return;
    default:
        System.out.println("Alarm: Lựa chọn đúng số trên
menu");
        break;
    }
}

private void initStudent() {
    for (int i = 0; i < 10; i++) {
        Student st = new Student("Student " + (i + 1),
random.nextInt(3)+1);
        listStudent.add(st);
    }
}
}

```

a) Tạo 10 học sinh, chia thành 3 nhóm

```

private void initStudent() {
    for (int i = 0; i < 10; i++) {

```

```

        Student st = new Student("Student " + (i + 1),
random.nextInt(3)+1);
        listStudent.add(st);
    }
}

```

b) Kêu gọi cả lớp điểm danh.

```

case 3:

    System.out.println("Cả lớp điểm danh: ");
    for (Student student : listStudent) {
        student.diemDanh();
    }
    break;

```

c) Gọi nhóm 1 đi học bài

```

case 4:

    System.out.println("Nhóm 1 đi học bài: ");
    for (Student student : listStudent) {
        if (student.getGroup() == 1) {
            student.hocBai();
        }
    }
    break;

```

d) Gọi nhóm 2 đi dọn vệ sinh

```

case 5:

    System.out.println("Nhóm 2 đi dọn vệ sinh: ");
    for (Student student : listStudent) {
        if (student.getGroup() == 2) {
            student.donVeSinh();
        }
    }
    break;

```

Question 2 (Optional): Abstract Management

Tạo 1 class **Person** chứa các property sau: **tên, giới tính, ngày sinh, địa chỉ** với đầy đủ getter setter, constructor không tham số, constructor đầy đủ tham số

→ Tạo Class Person

```

package com.vti.entity;

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

public class Person {
    private String name;
    private Gender gender;
    private LocalDate birthDate;
    private String address;

    public enum Gender {
        MALE, FEMALE, UNKNOWN;
    }
}

```

```

    }

    public Person(String name, Gender gender, LocalDate birthDate, String
address) {
        super();
        this.name = name;
        this.gender = gender;
        this.birthDate = birthDate;
        this.address = address;
    }

    public Person() {
        super();
    }

    public void inputInfo() {
        Scanner sc = new Scanner(System.in);
        System.out.println("Nhập thông tin Person từ bàn phím: ");
        System.out.println("Name: ");
        this.name = sc.next();
        System.out.println("Gender 1.Male, 2.Female, 3.Unknown: ");
        int flagGender = sc.nextInt();
        switch (flagGender) {
            case 1:
                this.gender = Gender.MALE;
                break;
            case 2:
                this.gender = Gender.FEMALE;
                break;
            case 3:
                this.gender = Gender.UNKNOWN;
                break;
        }
        System.out.println("BirthDate nhập theo định dạng yyyy-MM-dd: ");
        this.birthDate = LocalDate.parse(sc.next());
        System.out.println("Address: ");
        this.address = sc.next();
    }

    public String showInfo() {
        return "Person [name=" + name + ", gender=" + gender + ",
        birthDate=" + birthDate + ", address="
            + address + "];"
    }
}

```

a) Viết phương thức inputInfo(), nhập thông tin Person từ bàn phím

```

public void inputInfo() {
    Scanner sc = new Scanner(System.in);
    System.out.println("Nhập thông tin Person từ bàn phím: ");
    System.out.println("Name: ");
    this.name = sc.next();
}

```

```

        System.out.println("Gender 1.Male, 2.Female, 3.Unknown: ");
        int flagGender = sc.nextInt();
        switch (flagGender) {
            case 1:
                this.gender = Gender.MALE;
                break;
            case 2:
                this.gender = Gender.FEMALE;
                break;
            case 3:
                this.gender = Gender.UNKNOWN;
                break;
        }
        System.out.println("BirthDate nhập theo định dạng yyyy-MM-dd: ");
        this.birthDate = LocalDate.parse(sc.next());
        System.out.println("Address: ");
        this.address = sc.next();
    }
}

```

b) Viết phương thức showInfo(), hiển thị tất cả thông tin Person

```

public String showInfo() {
    return "Person [name=" + name + ", gender=" + gender + ",
    birthDate=" + birthDate + ", address="
        + address + "]\n";
}

```

Tạo class Student_Person thừa kế Person, lưu trữ các thông tin một sinh viên:
Mã sinh viên, Điểm trung bình, Email

```

package com.vti.entity;

import java.util.Scanner;

public class Student_Person extends Person {

    private int id;
    private Float gradeAvg;
    private String email;

    @Override
    public void inputInfo() {
        Scanner sc = new Scanner(System.in);
        super.inputInfo();
        System.out.println("ID: ");
        this.id = sc.nextInt();
        System.out.println("GradeAvg: ");
        this.gradeAvg = sc.nextFloat();
        System.out.println("email: ");
        this.email = sc.next();
    }

    @Override
    public String showInfo() {
        return super.showInfo() + " id= " + id + ", gradeAvg=" + gradeAvg
        + ", email=" + email + "]\n";
    }
}

```



```

    }

    public boolean Scholarship() {
        return gradeAvg > 8.0 ? true : false;
    }
}

```

a) Override phương thức inputInfo(), nhập thông tin Student từ bàn phím

```

@Override
public void inputInfo() {
    Scanner sc = new Scanner(System.in);
    super.inputInfo();
    System.out.println("ID: ");
    this.id = sc.nextInt();
    System.out.println("GradeAvg: ");
    this.gradeAvg = sc.nextFloat();
    System.out.println("email: ");
    this.email = sc.next();
}

```

b) Override phương thức showInfo(), hiển thị tất cả thông tin Student

```

@Override
public String showInfo() {
    return super.showInfo() + " id= " + id + ", gradeAvg=" + gradeAvg
+ ", email=" + email + "]";
}

```

c) Viết phương thức xét xem Student có được học bổng không?

Điểm trung bình từ 8.0 trở lên là được học bổng → chưa hiểu kiểu return này

```

public boolean Scholarship() {
    return gradeAvg > 8.0 ? true : false;
}

```

→ Tạo chương trình demo:

Tạo Class Exercise2_Ques2

```

package com.vti.backend;

import java.util.Scanner;

import com.vti.entity.Person;
import com.vti.entity.Student_Person;

public class Exercise2_Ques2 {
    private Scanner sc;

    public Exercise2_Ques2() {
        super();
        sc = new Scanner(System.in);
    }
}

```

```

public void question2() {

    while (true) {

        System.out.println("=====
=====");
        System.out.println("=====Lựa chọn chức năng bạn
muốn sử dụng=====");
        System.out.println("===          1. Demo Person.
===");
        System.out.println("===          2. Demo Stusent.
===");
        System.out.println("===          3. Exit.
===");

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

            case 2:
                demoStudent();
                break;
            case 3:
                return;

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

}

private void demoStudent() {
    Student_Person st = new Student_Person();
    st.inputInfo();
    System.out.println("Thông tin Student vừa nhập vào: ");
    System.out.println(st.showInfo());
    if (st.Scholarship()) {
        System.out.println("Sinh viên này đạt học bổng");
    } else {
        System.out.println("Sinh viên này không đạt học bổng");
    }
}

private void demoPerson() {
    Person person = new Person();
    person.inputInfo();
    System.out.println("Thông tin person vừa nhập vào: ");
    System.out.println(person.showInfo());
}
}

```

```
}
```

Tạo Class Demo_Exercise2_Ques2 có chứa hàm main để demo chương trình

```
package com.vti.frontend;

import com.vti.backend.Exercise2_Ques2;

public class Demo_Exercise2_Ques2 {
    public static void main(String[] args) {
        Exercise2_Ques2 ex2_ques2 = new Exercise2_Ques2();
        ex2_ques2.question2();
    }
}
```

Question 3: This & Super, Overriding

Tạo 1 class HìnhChuNhat có 2 method: tính chu vi và tính diện tích.
Hãy implement 2 method này.

→ Tạo class Rectangle

```
package com.vti.entity;

public class Rectangle {
    private Float a;
    private Float b;

    public Float tinhChuvi() {
        return 2 * (a + b);
    }

    public Float dienTich() {
        return a * b;
    }

    public Rectangle(Float a, Float b) {
        super();
        this.a = a;
        this.b = b;
    }
}
```

Tiếp theo hãy tạo 1 class HìnhVuong extends HìnhChuNhat, có 2 method tính chu vi và tính diện tích

→ Tạo class Square

```
package com.vti.entity;

public class Square extends Rectangle {
```

```

    public Square(Float a) {
        super(a, a);
        // TODO Auto-generated constructor stub
    }
}

```

Chú ý:

Không implement method trong class HìnhVuong theo công thức tính diện tích, chu vi hình vuông mà sẽ gọi theo

HìnhChuNhat.tinhChuVi(), HìnhChuNhat.tinhDienTich())

Và trong mỗi method print thêm dòng "Tính diện tích/ chu vi theo Hình Vuông/ Hình Chữ Nhật"

➔ Viết hàm main demo chương trình

```

package com.vti.backend;

import java.util.Scanner;

import com.vti.entity.Square;

public class Exercise2_Ques3 {
    private Square sq;
    private Scanner sc;

    public Exercise2_Ques3() {
        sc = new Scanner(System.in);
    }

    public void question() {

        while (true) {

            System.out.println("=====
=====");
            System.out.println("=====Lựa chọn chức năng bạn
muốn sử dụng=====");
            System.out.println("===                                1. Create Square.
===");
            System.out.println("===                                2. Tính chu vi hình
vuông.                                ===");
            System.out.println("===                                3. Tính diện tích hình
vuông.                                ===");
            System.out.println("===                                4. Exit.
===");

            System.out.println("=====
=====");
            int menuChoose = sc.nextInt();
            switch (menuChoose) {
                case 1:
                    System.out.println("Nhập vào cạnh của hình vuông: ");
                    Float a = sc.nextFloat();
                    sq = new Square(a);
                    System.out.println("Bạn vừa tạo thành công hình vuông
có cạnh " + a);

```

```

        break;
    case 2:
        System.out.println("Chu vi của hình vuông là: " +
sq.tinhChuvi());
        break;
    case 3:
        System.out.println("Diện tích của hình vuông là: " +
sq.diệnTich());
        break;
    case 4:
        return;
    default:
        System.out.println("Alarm: Lựa chọn đúng số trên
menu");
        break;
    }
}
}
}

```

Hàm main:

```

package com.vti.frontend;

import com.vti.backend.Exercise2_Ques3;

public class Demo_Exercise2_Ques3 {
    public static void main(String[] args) {
        Exercise2_Ques3 ex2_ques3 = new Exercise2_Ques3();
        ex2_ques3.question();
    }
}

```

Question 4 (Optional):

Tạo 1 class MyMath có method sum có 2 parameter (có thể là int, có thể là byte, có thể là float), sau đó trả về tổng của 2 số đó. Hãy viết chương trình demo.

→ Tạo class MyMath

```

package com.vti.entity;

public class MyMath {
    public int getSum(int par1, int par2) {
        return (par1 + par2);
    }

    public byte getSum(Byte par1, Byte par2) {
        return (byte) (par1 + par2);
    }

    public Float getSum(Float par1, Float par2) {
        return (par1 + par2);
    }
}

```

```

    public Float getSum(Byte par1, Float par2) {
        return (par1 + par2);
    }

    public int getSum(Byte par1, int par2) {
        return (par1 + par2);
    }

    public Float getSum(Float par1, int par2) {
        return (par1 + par2);
    }
}

```

→ Tạo Chương trình Demo

```

package com.vti.frontend;

import java.util.Scanner;

import com.vti.entity.MyMath;
import com.vti.entity.Square;

public class Demo_Exercise2_Ques4 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        MyMath myMath = new MyMath();
        while (true) {

            System.out.println("=====
            =====");
            System.out.println("=====Lựa chọn chức năng bạn
            muốn sử dụng=====");
            int.           System.out.println("===           1. Tính tổng 2 số kiểu
            byte.           ===");
            System.out.println("===           2. Tính tổng 2 số kiểu
            Float.          ===");
            System.out.println("===           3. Tính tổng 2 số kiểu
            int và float.    ===");
            System.out.println("===           4. Tính tổng 2 số kiểu
            Byte và float.   ===");
            System.out.println("===           5. Tính tổng 2 số kiểu
            ===");
            System.out.println("===           6. Exit.

            =====");
            System.out.println("=====
            =====");
            int menuChoose = sc.nextInt();
            switch (menuChoose) {
                case 1:
                    System.out.println("Nhập vào số int 1: ");
                    int int1 = sc.nextInt();
                    System.out.println("Nhập vào số int 2: ");
                    int int2 = sc.nextInt();
                    System.out.println("Tổng 2 số là: " +
                    myMath.getSum(int1, int2));
                    break;
            }
        }
    }
}

```

```

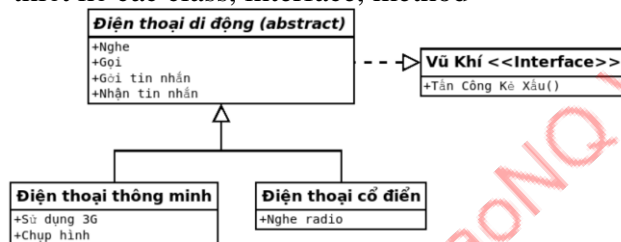
        case 2:
            System.out.println("Nhập vào số byte 1: ");
            byte byte1 = sc.nextByte();
            System.out.println("Nhập vào số byte 2: ");
            byte byte2 = sc.nextByte();
            System.out.println("Tổng 2 số là: " +
myMath.getSum(byte1, byte2));
            break;
        case 3:
            System.out.println("Nhập vào số float 1: ");
            Float float1 = sc.nextFloat();
            System.out.println("Nhập vào số float 2: ");
            Float float2 = sc.nextFloat();
            System.out.println("Tổng 2 số là: " +
myMath.getSum(float1, float2));
            break;
        case 4:
            System.out.println("Nhập vào số int 1: ");
            int int4 = sc.nextInt();
            System.out.println("Nhập vào số float 2: ");
            Float float4 = sc.nextFloat();
            System.out.println("Tổng 2 số là: " +
myMath.getSum(float4, int4));
            break;
        case 5:
            System.out.println("Nhập vào số byte 1: ");
            byte byte5 = sc.nextByte();
            System.out.println("Nhập vào số float 2: ");
            Float float5 = sc.nextFloat();
            System.out.println("Tổng 2 số là: " +
myMath.getSum(byte5, float5));
            break;
        case 6:
            return;
    }
}
}
}
}

```

Question 5 (Optional):

Điện thoại di động được chia thành 2 loại: **điện thoại cổ điển và điện thoại thông minh**. Cả 2 loại điện thoại này đều có 4 chức năng cơ bản: **nghe và gọi điện thoại, gửi và nhận tin nhắn văn bản**. Riêng điện thoại thông minh thì có thêm các chức năng: **sử dụng 3G, chụp hình**. Điện thoại cổ điển có sẵn chức năng **nghe đài radio** (điện thoại thông minh không có sẵn chức năng này). Cả điện thoại cổ điển và điện thoại thông minh đều có thể sử dụng làm vũ khí (trong tình huống người dùng cần sử dụng để tấn công kẻ xấu).

Hãy thiết kế các class, interface, method



Gợi ý (à không, đáp án mới đúng)

→ Tạo class iTanCong

```
package com.vti.entity.polymorphism.question5;

public interface iTanCong {
    void tanCong();
}
```

→ Tạo class DienThoaiDiDong

```
package com.vti.entity.polymorphism.question5;

public abstract class DienThoaiDiDong implements iTanCong {

    public void nghe() {
        System.out.println("Đang nghe...");
    }

    public void gọi() {
        System.out.println("Đang gọi...");
    }

    public void guiTinNhan() {
        System.out.println("Đang gửi tin nhắn...");
    }

    public void nhanTinNhan() {
        System.out.println("Đã nhận tin nhắn...");
    }
}
```

→ Tạo Class DienThoaiThongMinh

```
package com.vti.entity.polymorphism.question5;

public class DienThoaiThongMinh extends DienThoaiDiDong {

    @Override
    public void tanCong() {
        System.out.println("Ném là vỡ màn ...");
    }

    public void suDung3G() {
        System.out.println("Sử dụng 3G ...");
    }
}
```



```
    }  
  
    public void chupHinh() {  
        System.out.println("Chup hình ...");  
    }  
}
```

→ **Tạo class DienThoaiCoDien**

```
package com.vti.entity.polymorphism.question5;  
  
public class DienThoaiCoDien extends DienThoaiDiDong {  
  
    @Override  
    public void tanCong() {  
        System.out.println("Ném là chết chó...");  
    }  
  
    public void ngheDaiRadio() {  
        System.out.println("Nghe Đài radio...");  
    }  
}
```