

```

public class Main {
    public static void main(String[] args) {
        Sim sim = new Sim();
        char sc;
        do {
            menu();
            sc = Input.leggiChar('a','g',"Scelta: ");
            switch (sc) {
                case 'a': {
                    attivaSim(sim);
                    break;
                }
                case 'b': {
                    ricaricaSim(sim);
                    break;
                }
                case 'c': {
                    chiama(sim);
                    break;
                }
                case 'd': {
                    showOre(sim);
                    break;
                }
                case 'e': {
                    disattivaSim(sim);
                    break;
                }
                case 'f':{
                    statoSim(sim);
                    break;
                }
            }
        }while(sc!='g');
    }
    private static void statoSim(Sim sim) {
        String status = sim.toString();
        System.out.println(status);
    }
    private static void showOre(Sim sim) {
        String ore = sim.oreChiamata();
        System.out.println(ore);
    }
    private static void disattivaSim(Sim sim) {
        sim.setActive(false);
        System.out.println("Sim disattivata con successo");
    }
    public static void menu(){
        System.out.println("a)Attiva SIM");
        System.out.println("b)Ricarica SIM");
        System.out.println("c)Chiama");
        System.out.println("d)Mostra minuti");
        System.out.println("e)Disattiva SIM");
        System.out.println("f)Mostra status SIM");
    }
}

```

```

        System.out.println("g) Termina programma");
    }
    public static void attivaSim(Sim sim){
        sim.setActive(true);
        System.out.println("Sim attivata con successo");
    }
    public static void ricaricaSim(Sim sim){
        char sc;
        System.out.println("Scegliere il taglio della ricarica");
        System.out.println("a) 5 €");
        System.out.println("b) 10 €");
        System.out.println("c) 20 €");
        sc = Input.leggiChar('a','c',"Scelta:");
        switch (sc){
            case 'a':{
                sim.addCredito(5f);
                break;
            }
            case 'b':{
                sim.addCredito(10f);
                break;
            }
            case 'c':{
                sim.addCredito(20f);
                break;
            }
        }
    }
}
    public static void chiama(Sim sim){
        int minuti = Input.leggiInt(0,Integer.MAX_VALUE,"Per quanti minuti si
desidera chiamare");
        int status = sim.call(minuti);
        switch (status){
            case 0:{
                System.out.println("Impossibile chiamare, la SIM non è
attiva");
            }
            case 1:{
                System.out.println("La chiamata è andata a buon fine");
            }
            case 2:{
                System.out.println("Impossibile chiamare, credito
insufficiente");
            }
        }
    }
}
}
////////////////////////////////////
public class Sim implements Comparable<Sim> {
    private String nome;
    private String operatore;
    private String iccId;
    private String numero;
    private float credito;

```

```

private int minuti;
private boolean isActive;

//COSTRUTTORI
public Sim(){
    setActive(false);
    setOperatore("Tim");
    setCredito(5);
    setNome("NULL");
    setNumero("1235468790");
    setIccId("12345678901234567890");
    setMinuti(0);
}

public Sim(String nome,String operatore,String numero,String iccId){
    setOperatore("Postemobile");
    setActive(false);
    setCredito(0);
    setNome(nome);
    setNumero(numero);
    setIccId(iccId);
    setMinuti(0);
}

public Sim(Sim a){
    setActive(a.isActive());
    setCredito(a.getCredito());
    setNome(a.getNome());
    setNumero(a.getNumero());
    setIccId(a.getIccId());
    setMinuti(a.getMinuti());
}

//SETTER

public void setOperatore(String operatore) {
    if (!operatore.trim().isEmpty()) {
        this.operatore = operatore;
    } else
        this.operatore = "NULL";
}

public void setActive(boolean active) {
    this.isActive = active;
}

public void setCredito(float credito) {
    if(credito>=0){
        this.credito = credito;
    }
    else this.credito = 5;
}

public void setNome(String nome) {
    if (!nome.trim().isEmpty()) {

```

```

        this.nome = nome;
    } else
        throw new IllegalArgumentException("Errore - nome intestatario
errato - SIM NON ATTIVA");
    }
    public void setNumero(String numero) {
        if (!(numero.trim().isEmpty())) {
            this.numero = numero;
        } else
            this.numero = "1234567890";
    }

    public void setMinuti(int minuti) {
        if(minuti>=0){
            this.minuti = minuti;
        }
        else this.minuti = 0;
    }

    public void setIccId(String iccId) {
        if(!(iccId.trim().isEmpty())){
            this.iccId = iccId;
        }else{
            this.iccId = "12354678901234568790";
        }
    }

    //GETTER
    public float getCredito() {
        return credito;
    }

    public String getOperatore() { return operatore; }

    public boolean isActive() {
        return isActive;
    }

    public int getMinuti() {
        return minuti;
    }

    public String getIccId() {
        return iccId;
    }

    public String getNome() {
        return nome;
    }
    public String getNumero() {
        return numero;
    }
    //METODI
    public void addCredito(float credito){

```

```

        if(credito>0){
            this.credito+=credito;
        }
    }
    public int call(int minuti){
        int n;
        float creditoEnough;
        creditoEnough = this.credito * 0.32f;
        if(isActive()){
            if(creditoEnough>this.credito){
                n=1;
                this.credito=-creditoEnough;
            }else{
                n=2;
            }
        }else{
            n=0;
        }
        return n;
    }

    public String oreChiamata() {
        float ore = minuti / 60;
        float minutiCall = minuti % 60;
        String str = "Ore :" + ore + "Minuti :" + minutiCall;
        return str;
    }

    @Override
    public String toString() {
        return "Sim{" + "nome='" + nome + '\'' +
            ", iccId='" + iccId + '\'' +
            ", numero='" + numero + '\'' +
            ", credito=" + credito +
            ", minuti=" + minuti +
            ", isActive=" + isActive +
            '}';
    }

    @Override
    public int compareTo(Sim o) {
        int n;
        if(o.getIccId().compareTo(this.iccId)==0){
            n = 0;
        }
        if(o.getIccId().compareTo(this.iccId)>0){
            n = 1;
        }else{
            n = -1;
        }
        return n;
    }
}

////////////////////////////////////
import java.util.InputMismatchException;
import java.util.Scanner;

```

```

public class Input {
    public static int leggiInt (int vmin, int vmax,String msg){
        int n=0;
        boolean err;
        do{
            System.out.println(msg);
            Scanner input = new Scanner(System.in);
            err = false;
            try {
                n = input.nextInt();
                input.nextLine();
            }catch (InputMismatchException e){
                System.out.println("Ciò che hai inserito non è un numero");
                err = true;
            }
            if(n<vmin || n>vmax)
                System.out.println("Errore");
        }while((n<vmin || n>vmax) || err);
        return n;
    }

    public static char leggiChar (char vmin, char vmax,String msg){
        char c='.';
        boolean err;
        do{
            System.out.println(msg);
            Scanner input = new Scanner(System.in);
            err = false;
            try {
                c = input.nextLine().charAt(0);
                input.nextLine();
            }catch (InputMismatchException e){
                System.out.println("Ciò che hai inserito non è un carattere");
                err = true;
            }
            if(c<vmin || c>vmax)
                System.out.println("Errore");
        }while((c<vmin || c>vmax) || err);
        return c;
    }

    public static float leggiFloat (float vmin, float vmax,String msg){
        float n=0;
        boolean err;
        do{
            System.out.println(msg);
            Scanner input = new Scanner(System.in);
            err = false;
            try {
                n = input.nextFloat();
                input.nextLine();
            }catch (InputMismatchException e){
                System.out.println("Ciò che hai inserito non è un numero
decimale");
            }
        }while((n<vmin || n>vmax) || err);
        return n;
    }
}

```

```

        err = true;
    }
    if(n<vmin || n>vmax)
        System.out.println("Errore");
    }while((n<vmin || n>vmax) || err);
    return n;
}

public static Double leggiDouble (double vmin, double vmax,String msg){
    double n=0;
    boolean err;
    do{
        System.out.println(msg);
        Scanner input = new Scanner(System.in);
        err = false;
        try {
            n = input.nextFloat();
            input.nextLine();
        }catch (InputMismatchException e){
            System.out.println("Ciò che hai inserito non è un numero
decimale");
            err = true;
        }
        if(n<vmin || n>vmax)
            System.out.println("Errore");
    }while((n<vmin || n>vmax) || err);
    return n;
}

public static String leggiStr(String msg) {
    String s="";
    boolean err;
    do{
        System.out.println(msg);
        Scanner input = new Scanner(System.in);
        err = false;
        try {
            s = input.nextLine();
            input.nextLine();
        }catch (InputMismatchException e){
            System.out.println("Ciò che hai inserito non è una Stringa");
            err = true;
        }
        if(s.trim().isEmpty())
            System.out.println("Errore");
    }while(s.trim().isEmpty() || err);
    return s;
}

}

```