Correction Exercices POO (Java)

Préambule

1°)

	Classe1	Classe2	Classe3	Classe4	Classe5
a	X	X			
b	X	X	X		
c		X			
d	X	X	X	X	X
e	X	X	X	X	
f	X	X	X		

2°)

cas a:

Une base de travail pourrait être :

```
package esgi.chapter.two;

public class Employee {
   private String name;
   private Service service;
   //fonctions habituelles

   public Employee(Service service) {    }
   public Service getService() {    }
   public boolean setService(Service newService) {     }
   public void delete() {    }
}

package esgi.chapter.one;

public class Service {
   private String name;
   private List<Employee> employees;
   //fonctions habituelles

   public Service() {    }
   public Employee getEmployee(int index) {    }
   public List<Employee> getEmployees() {    }
   public int numberOfEmployees() {    }
   public boolean hasEmployees() {    }
   public int indexOfEmployee(Employee employee) {    }
   public static int minimumNumberOfEmployees() {    }
   public Employee addEmployee() {    }
   public boolean addEmployee(Employee newEmployee) {    }
   public boolean removeEmployee(Employee aEmployee) {    }
   public boolean addEmployeeAt(Employee aEmployee, int index) {    }
   public void delete() {
   }
```

cas b:

```
package esgi.chapter.two;

public class Person {
   private List<Person> persons;
   private List<Person> parents;

   public Person() {
      persons = new ArrayList<Person>();
      parents = new ArrayList<Person>();
   }

   public Person getPerson(int index) {     }
   public List<Person> getPersons() {    }
   public int numberOfPersons() {    }
   public boolean hasPersons() {    }
   public int indexOfPerson(Person person) {    }
   public Person getParent(int index) {    }
   public List<Person> getParents() {    }
   public int numberOfParents() {    }
   public boolean hasParents() {    }
   public int indexOfParent(Person parent) {    }
   public static int minimumNumberOfPersons() {    }
   public boolean addPerson(Person person) {    }
   public boolean removePerson(Person person) {    }
   public boolean addPersonAt(Person person, int index) {    }
   public static int minimumNumberOfParents() {
      return 0;
}
  public static int maximumNumberOfParents() {
      return 2;
   }
   public boolean addParent(Person parent) {    }
   public boolean removeParent(Person parent) {    }
   public boolean setParents(Person... newParents) {    }
   public boolean addParentAt(Person parent, int index) {    }
   public void delete() {    }
}
```

cas c:

```
package esgi.chapter.two;

public class Student {
   private String name;
   private List<Result> results;

   public Student(String name) {
      this.name = name;
      results = new ArrayList<Result>();
   }

   public boolean setName(String name) {    }
   public String getName() {    }
   public Result getResult(int index) {    }
   public List<Result> getResults() {    }
   public int numberOfResults() {    }
   public boolean hasResults() {    }
   public int indexOfResult(Result result) {    }
   public static int minimumNumberOfResults() {
      return 0;
   }
   public Result addResult(double mark, Course course) {    }
   public boolean addResult(Result result) {    }
   public boolean removeResult(Result result) {    }
   public boolean addResultAt(Result result, int index) {    }
   public void delete() {    }
   public String toString() {    }
}
```

```
package esgi.chapter.two;

public class Result {
   private double mark;
   private Student student;
   private Course course;

   public Result(double mark, Student student, Course course) {    }
   public boolean setMark(double mark) {    }
   public double getMark() {    }
   public Student getStudent() {    }
   public Course getCourse() {    }
   public boolean setStudent(Student student) {    }
   public boolean setCourse(Course course) {    }
   public void delete() {    }
   public String toString() {    }
}
```

```
package esgi.chapter.two;

public class Course {
   private String subject;

   public Course(String aSubject) {    }
   public boolean setSubject(String aSubject) {    }
   public String getSubject() {    }
   public String toString() {
   }
}
```

Exercice 1

a) et b)

fichier Dice.java

```
package esgi.chapter.two;

import java.util.Random;

public class Dice {
   private byte value;
   private static Random generator = new Random();

   public Dice(byte value) {
      setValue(value);
   }

   public boolean setValue(byte value) {
      if (value >= 1 && value <= 6) {
         this.value = value;
         return true;
      }
      return false;
   }

   public byte getValue() {
      return value;
   }

   public byte rollDice() {
      value = (byte) (generator.nextInt(6) + 1);
      return value;
   }
}
```

fichier DiceTest.java

```
package esgi.chapter.two;

public class DiceTest {
   public static void main(String args[]) {
      if (args.length < 1) {
         System.out.println("Il faut fournir un paramètre");
         System.exit(-1);
      }
      int counter = Integer.parseInt(args[0]);
      int j;
      Dice dice = new Dice((byte) 1);
      for (j = 0; j < counter; j++)
         System.out.println("rollDice : " + dice.rollDice());
   }
}
```

c) Par exemple

fichier DiceTest.java

```
package esgi.chapter.two;

class DiceTest {
   public static void main(String args[]) {
      if (args.length < 1) {
         System.out.println("Il faut fournir un paramètre");
         System.exit(-1);
      }
      Dice diceA = new Dice((byte) 1);
      Dice diceB = new Dice((byte) 1);
      byte valueA, valueB;
      int counter = Integer.parseInt(args[0]), i;
      for (i = 1; i <= counter; i++) {
         valueA = diceA.rollDice();
         valueB = diceB.rollDice();
         System.out.println("lancer : " + i + ", des 1: " + valueA + ", des 2: " + valueB);
         if (valueA == valueB)
            System.out.println("\tlancer gagne !");
      }
   }
}
```

Exercice 2

a) Correction partie programme

Fichier Mark.java

```
package esgi.chapter.two;

public class Mark {
   private double value;
   private String subject;

   public Mark() {
   }

   public Mark(double value, String subject) {
      if (value >= 0 && value <= 20)
         this.value = value;
      this.subject = subject;
   }

   public double getValue() {
      return value;
   }

   public void setValue(double value) {
      if (value >= 0 && value <= 20)
         this.value = value;
   }
   public String getSubject() {
      return subject;
   }

   public void setSubject(String subject) {
      this.subject = subject;
   }

   @Override
   public String toString() {
      return "Mark{" +
            "value=" + value +
            ", subject='" + subject + '\'' +
            '}';
   }
}
```

```
}
```

Fichier Application.java (solution donnée avec des tableaux)

```
package esgi.chapter.two;
import java.util.Arrays;

public class Application {
  private Mark[][] array;

  public Application(Mark[][] array) {
    this.array = array;
  }

  public void print() {
    if (array == null) {
      System.out.println("Tableau vide");
      return;
    }
    System.out.println();
    for (int i = 0; i < array.length; i++) {
      System.out.println("Etudiant : " + i);
      if (array[i] == null)
        continue;
      for (int j = 0; j < array[i].length; j++) {
        System.out.println(array[i][j]);
      }
    }
  }

  public double average(int position) {
   if(array==null || position < 0 || position >= array.length || array[position]==null)
      return -1;
    double s = 0.;
    for (int i = 0; i < array[position].length; i++)
      s += array[position][i].getValue();
    return s / array[position].length;
  }

  public void add(int position, String subject, double value) {
    if (array == null || position < 0 || position >= array.length)
      return;
    int size = (array[position] == null ? 1 : array[position].length + 1);
    Mark[] inter = new Mark[size];
    int i;
    for (i = 0; i < array[position].length; i++)
      inter[i] = array[position][i];
    inter[i] = new Mark(value, subject);
    array[position] = inter;
  }


  public int compute() {
    if (array == null)
      return 0;
    int max = 0;
    for (int i = 0; i < array.length; i++) {
      if (array[i] != null && max < array[i].length)
        max = array[i].length;
    }
    return max;
  }

  public double[] avgs() {
    double[] result = null;
    if (array == null) return result;
```

```
        int i, counter = 0, k =  0;
        for (i = 0; i < array.length; i++)
            if (array[i] != null) counter++;
        result = new double[counter];
        for (i = 0; i < array.length; i++) {
            if (array[i] == null) continue;
            result[k] = average(i);
            k++;
        }
        Arrays.sort(result);
        return result;
    }

    public static void main(String[] args) {
        Mark[][] array = new Mark[3][];
        array[0] = new Mark[3];
        array[0][0] = new Mark();
        array[0][0].setSubject("Java");
        array[0][0].setValue(12.25);
        array[0][1] = new Mark(2., "C");
        array[0][2] = new Mark(15, "C#");
        array[2] = new Mark[1];
        array[2][0] = new Mark();
        array[2][0].setSubject("Java");
        array[2][0].setValue(14);
        Application run = new Application(array);
        run.print();
        System.out.println("Taille max : " + run.compute());
        for (int i = 0; i < array.length; i++)
            System.out.println(run.average(i));
        run.add(0, "PHP", 14.);
        run.print();
        System.out.println("Taille max : " + run.compute());
        double[] res = run.avgs();
        for (int i = 0; i < res.length; i++)
            System.out.println(res[i]);
    }
}
```

b) Par exemple :

```
HashSet<Mark> hashset = new HashSet<Mark>();
Mark n = new Mark (12,"math");
Mark m = new Mark (12,"math");
hashset.add(n);
hashset.add(m);
Iterator<Note> it= hashset.iterator();
while(it.hasNext()) {
        System.out.println(it.next());
}
```

donne deux objets sans la redéfinition de hashCode et un seul avec la redéfinition de hashCode.

Exercice 3

```
package esgi.chapter.two;

import java.util.Objects;

public class Bus {

    private int numBus;
    private String start;
    private String destination;
    private int capacity;
    private int numTravellers;
    private int currentStop;
    private int numStops;
    private boolean returns;

    public Bus(int numBus, String start, String destination, int capacity, int numStops) {
        this.numBus = numBus;
        this.start = start;
        this.destination = destination;
        this.capacity = capacity;
        this.numStops = numStops;
        this.currentStop = 1;
    }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;
        Bus bus = (Bus) o;
        return numBus == bus.numBus &&
                capacity == bus.capacity &&
                numTravellers == bus.numTravellers &&
                currentStop == bus.currentStop &&
                numStops == bus.numStops &&
                returns == bus.returns &&
                Objects.equals(start, bus.start) &&
                Objects.equals(destination, bus.destination);
    }

    @Override
    public int hashCode() {
        return Objects.hash(numBus, start, destination, capacity, numTravellers, currentStop, numStops,
returns);
    }

    public int getNumTravellers() {
        return numTravellers;
    }

    public int getStop() {
        return currentStop;
    }

    public void addTravellers(int number) {
        if (number < 0)
            return;
        if (numTravellers + number > capacity)
            numTravellers = capacity;
        else numTravellers += number;
    }

    public void removeTravellers(int number) {
        if (number < 0)
            return;
        if (numTravellers - number < 0)
            numTravellers = 0;
        else numTravellers -= number;

    }

    public void travel() {
        if (!returns) {
            currentStop += 1;
            if (currentStop == numStops) {
                returns = true;
            }
        } else {
            currentStop -= 1;
            if (currentStop == 1) {
                returns = false;
            }
        }
    }

    public String getDestination() {
        return destination;
    }

    @Override
    public String toString() {
        return "Bus{" +
                "numBus=" + numBus +
                ", start='" + start + '\'' +
                ", destination='" + destination + '\'' +
                ", capacity=" + capacity +
                ", numTravellers=" + numTravellers +
                ", currentStop=" + currentStop +
                ", numStops=" + numStops +
                ", returns=" + returns +
                '}';
    }
}

package esgi.chapter.two;
import java.util.Scanner;


public class Test {

    public static void main(String[] args) {

        int number;
        Scanner scanner = new Scanner(System.in);
        System.out.print("Quel numero ?");
        int numBus = scanner.nextInt();

        scanner.nextLine();
        System.out.print("Quelle station de démarrage ?");
        String start = scanner.nextLine();

        System.out.print("Quelle station d'arrivée ?");
        String destination = scanner.nextLine();

        System.out.print("Combien d'arrets ? ");
        int numStops = scanner.nextInt();
        System.out.print("Quelle capacité ? ");
        int capacity = scanner.nextInt();
        scanner.nextLine();

        Bus bus = new Bus(numBus, start, destination, capacity, numStops);
        System.out.println();
        System.out.println(bus);
        System.out.println();

        char menuChoice;
        do {
            System.out.println("CHOOSE:");
            System.out.println("1 - Avance d'un arret");
            System.out.println("2 - Prend des passagers");
            System.out.println("3 - Depose des passengers");
            System.out.println("4 - Fin");
            menuChoice = scanner.nextLine().toUpperCase().charAt(0);

            switch (menuChoice) {
                case '1':
                  bus.travel();
                  System.out.println(bus);
                  break;
                case '2':
                  System.out.print("Combien de passagers ? ");
                  number = scanner.nextInt();
                  scanner.nextLine();
                  bus.addTravellers(number);
                  System.out.println(bus);
                  break;
                case '3':
                  System.out.print("Combien de passagers ? ");
                  number = scanner.nextInt();
                  scanner.nextLine();
                  bus.removeTravellers(number);
                  System.out.println(bus);
                  break;
                case '4':
                  System.out.println("Fin de travail!");
                  break;
                default:
                  System.out.println("Erreur de saisie");
            }
        } while (menuChoice != '4');
    }
}
```

Exercice 4

fichier Application.java

```
package esgi.chaptertwo;

public class Application {
   public static void main(String[] args) {
      Student student = new Student();
      Representative representative = new Representative();
      Teacher teacher = new Teacher();
      student.init();
      teacher.instruction(representative, student);
   }
}
```

fichier Student.java

```
package esgi.chaptertwo;
import java.util.Scanner;

public class Student {
   private int age;
   private int parentAge;
   private Paper paper = new Paper();
   private static Scanner sc = new Scanner(System.in);

   public int getAge() {
      return age;
   }

   public void setAge(int age) {
      this.age = age;
   }

   public int getSum() {
      return parentAge;
   }

   public void setSum(int sum) {
      this.parentAge = sum;
   }

   public Paper getPaper() {
      return paper;
   }

   public void setPaper(Paper paper) {
      this.paper = paper;
   }

   public void init() {
      System.out.println("saisir age ");
      age = sc.nextInt();
      System.out.println("saisir age parent ");
      parentAge = sc.nextInt();
   }

   public void writePaper() {
      paper.setPAge(age);
      paper.setParentAge(parentAge);
   }
}
```

fichier Representative.java

```
package esgi.chaptertwo;

public class Representative {
   private Paper dp;
   private int result;

   public Paper getDP() {
      return dp;
   }

   public void setDP(Paper paper) {
      this.dp = paper;
   }

   public void compute() {
      result = dp.getPAge() * 2;
      result = result + 5;
      result = result * 50;
      result = result + dp.getParentAge() - 365;
      System.out.println("le nombre est : " + result);
   }


   public int getResult() {
      return result;
   }
}
```

fichier Paper.java

```
package esgi.chaptertwo;

public class Paper {
   private int pAge;
   private int pParentAge;

   public int getPAge() {
      return pAge;
   }

   public void setPAge(int age) {
      this.pAge = age;
   }

   public int getParentAge() {
      return pParentAge;
   }

   public void setParentAge(int sum) {
      this.pParentAge = sum;
   }
}
```

fichier Teacher.java

```
package esgi.chaptertwo;

public class Teacher {
   private int realAge;
   private int realParentAge;

   public void instruction(Representative representative, Student student) {
      int result;
      student.writePaper();
      representative.setDP(student.getPaper());
      representative.compute();
      result = representative.getResult();
      realCompute(result);
      printResult();

   }

   public void realCompute(int result) {
      result = result + 115;
      realAge = result / 100;
      realParentAge = result % 100;
   }


   public void printResult() {
      System.out.println("Votre age est de : " + realAge);
      System.out.println("L'age de votre parent est de : " + realParentAge);
   }

}
```