Mehdi Bouchachi

THE COMPLETE JAVA

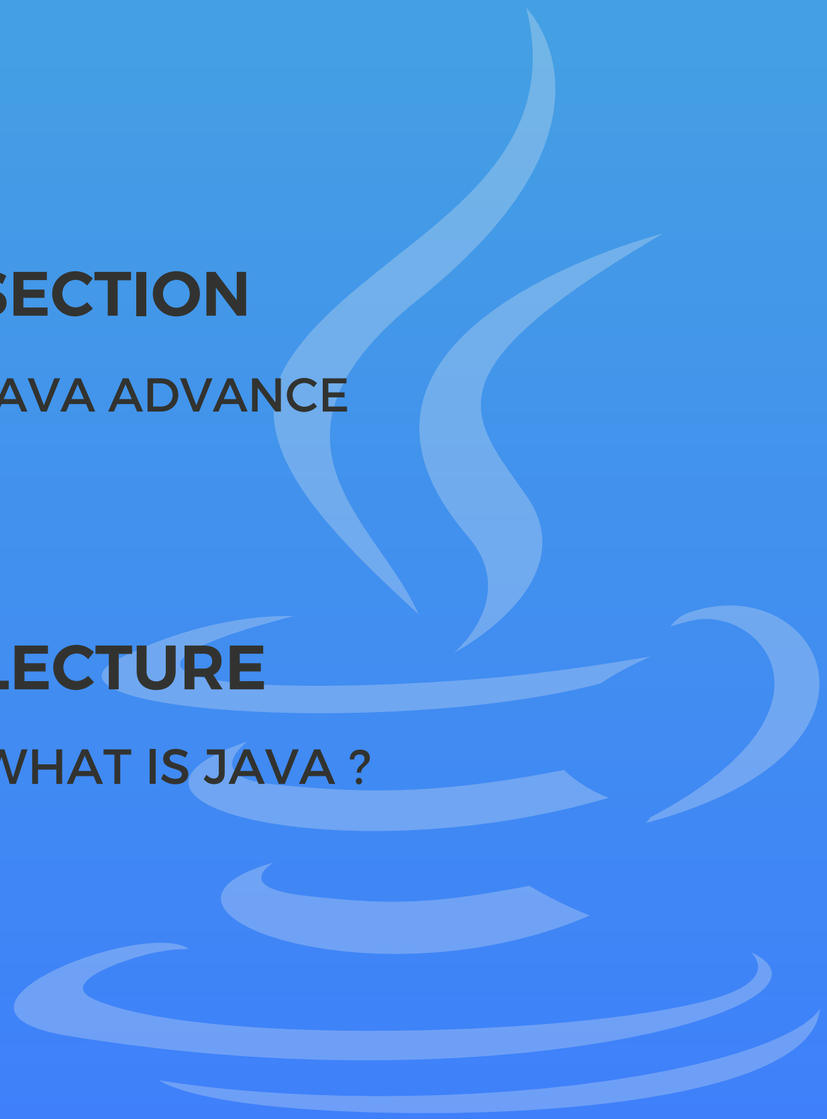
MehdiBouchachi

SECTION

JAVA ADVANCE

LECTURE

WHAT IS JAVA ?



NOTE

NOTE

👈 ماتفوتوا حتى page خطرش كلش متسلسل انا نضمنلك اذا كملت كامل الPDF و كنت قاري جافا من1, 2 PDF راح تخرج ماستر في JAVA مور ماتكمل هذا الPDF (اقل شيء قدر المجهود لي درتوا باش بسطتلك و وضحتلك كلش خطوة بخطوة و تفصيلة بتفصيلة :)).

👈 امشي في غرضك, ونتا تقرا فيه اكتب الكود و جرب تفهم شوية منو وحدك منبعد شوف الشرح تاع الكود و اكتبو في ورقة باش تفهم اكثر, و تولي تحل و تفهم اي كود مكتوب بالجافا :) .

👈 مور هذو الثلاثة PDFs روح تحل الExams لي بقاو (شوف الحل افهموا و ادي الفكرة من الكود و عاود حلوا وحدك).

👈 موفقين ان شاء الله :).

QUICK REVEIW

ARRAYLIST

- هو نوع **متطور من** الArray ياخذ بالعادة مجموعة من **ال**Objects **بداخله.**
- **يمتاز بديناميكية** اي ليس علينا تحديد **طوله او عدد العناصر** التي سوف تكون بداخله.
- يمكن **ادخال عناصر جديدة** بداخله حتى **بعد تعريفيه بسبب ال**Methods التي يوفرها لنا الArraylist.
- **اهم ال**ArrayList Methods **: هي** size **(طول)** , get , isEmpty, clear , remove , add
- **لصناعة** ArrayList :

```
ArrayList <Object> arrayName = new ArrayList <Object> ();
```

QUICK REVEIW

FINAL KEYWORD

👈 تستخدم حتى تحصر المتغير او الميثود و تمنعها من التغيير.(يصبح **CONSTANT**)

👈اذا قمت بصناعة **دالة final** داخل **SuperClass**, لا يمكنك التعديل عليها في **SubClass**, اي لا يمكن عمل **override** لها.

EXAM

```java
import java.util.ArrayList;

public class Rectangle{

    private double longueur, largeur;
    private static ArrayList <Rectangle> instances = new ArrayList <Rectangle>();

    public Rectangle (double lo, double la){
        longueur=lo;
        largeur=la;
        instances.add(this);
    }

    public void affiche () {
        System.out.println("Rectangle la longueur : "+ longueur + " et la largeur : "+largeur );
    }

    public static ArrayList <Rectangle> getInstances(){
        return instances;
    }

    public double surface (){
        return longueur*largeur;
    }

    public double getLongueur() {
        return longueur;
    }

    public double getLargeur() {
        return largeur;
    }

    public void setLongueur (double lon) {
        longueur = lon;
    }
    public void setLargeur (double larg) {
        largeur = larg;
    }
}
```

```java
public class Carré extends Rectangle {

    public Carré (double cote) {
        super(cote, cote);
    }

    public void affiche () {
        System.out.println(" Carré la longueur du cote: "+ getLargeur () );
    }

    public static ArrayList <Rectangle> getInstances(){

        ArrayList <Rectangle> instancesCarre =new ArrayList <Rectangle>();

        ArrayList <Rectangle> instances = Rectangle.getInstances();

        for (int i=0;i<instances.size();i++){

            if(instances.get(i) instanceof Carré){

                instancesCarre.add(instances.get(i));

            }
        }
        return instancesCarre;
    }
```

EXAM

MAIN

```java
public class Main {
    public static void main(String[] args) {

        Rectangle myInstances[]=new Rectangle [4];

        for (int i=0; i < myInstances.length ; i++){

            if(i % 2 == 0){

                myInstances[i]= new Rectangle (Math.random() ,Math.random());

            }else {

                myInstances[i]= new Carré (Math.random());

            }
        }
        System.out.println("Les Carrés: "+Carré.getInstances().size());
        System.out.println("Les Rectangles: "+Rectangle.getInstances().size());

    }
}
```

EXAM

CODE EXPLANATION

👈 قمنا بانشاء **كلاس** Rectangle الذي يحتوي على **متغيرين** longueur **و** largeur.

👈 قمنا بانشاء ايضا ArrayList instances الذي سوف يحتوي بداخله على عدد **من** Rectangle instance (شرحنا في ال2 PDF واش هو الinstance ارجع ليه)

```java
public class Rectangle{

    private double longueur, largeur;
    private static ArrayList <Rectangle> instances = new ArrayList <Rectangle>();

    public Rectangle (double lo, double la){
        longueur = lo;
        largeur = la;
        instances.add(this);
    }

    public void affiche () {
        System.out.println("Rectangle la longueur : "+ longueur + " et la largueur : "+largeur );
    }

    public static ArrayList <Rectangle> getInstances(){
        return instances;
    }
```

EXAM

CODE EXPLANATION

👈 قمنا بانشاء Constructor Rectangle.

👈 نريد **حفظ جميع ال**instances **التي نصنعها في ال**ArrayList instances و قلنا في الدروس السابقة ان عند صناعة instance (صناعة Object) **لكلاس ما دائما ينفذ ال**Constructor **الخاص بالكلاس مباشرة** لذلك قمنا بوضع ال(this)instances.add داخل ال**Construtor**

و this تشير الى هذا ال instance الذي **صنعته, اي ضف** هذا **ال**instance الى داخل ال ArrayList instances.

```java
public class Rectangle{

    private double longueur, largeur;
    private static ArrayList <Rectangle> instances = new ArrayList <Rectangle>();

    public Rectangle (double lo, double la){
        longueur = lo;
        largeur = la;
        instances.add(this);
    }

    public void affiche () {
        System.out.println("Rectangle la longueur : "+ longueur + " et la largueur : "+largeur );
    }

    public static ArrayList <Rectangle> getInstances(){
        return instances;
    }
```

EXAM

CODE EXPLANATION

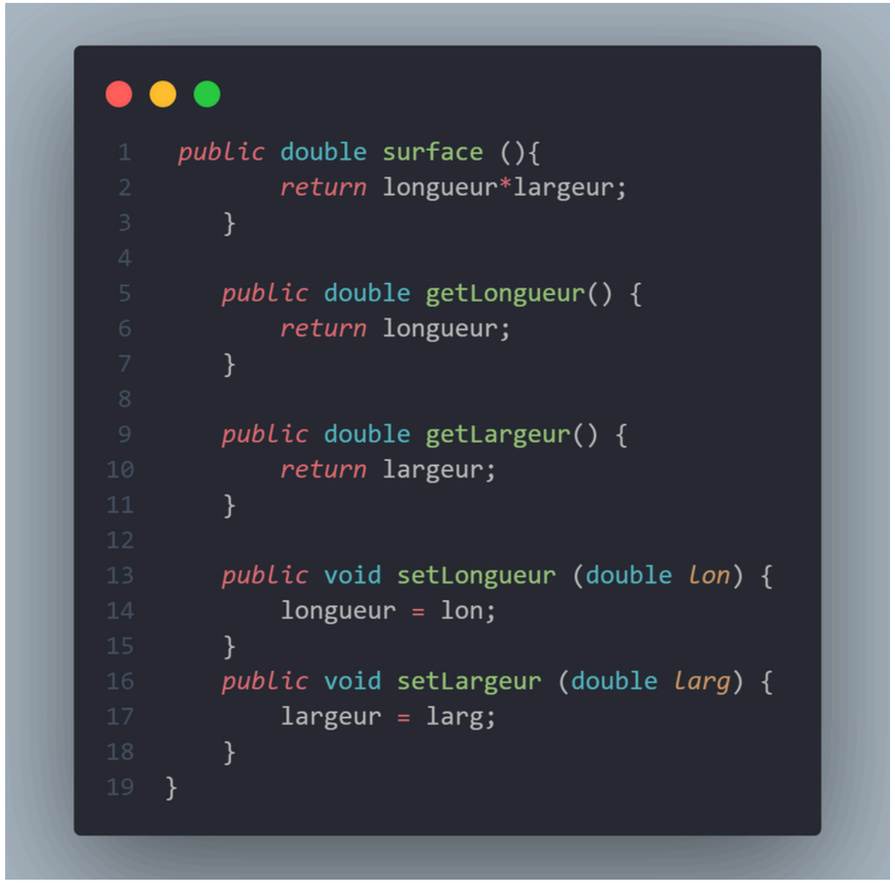
👈 قمنا بانشاء الدالة **affiche** التي تقوم بالطباعة.

👈 قمنا بانشاء الدالة **getInstances** التي تقوم **بارجاع ال ArrayList المحمل ب Rectangle instances.**

```java
public class Rectangle{

    private double longueur, largeur;
    private static ArrayList <Rectangle> instances = new ArrayList <Rectangle>();

    public Rectangle (double lo, double la){
        longueur = lo;
        largeur = la;
        instances.add(this);
    }

    public void affiche () {
        System.out.println("Rectangle la longueur : "+ longueur + " et la largeur : "+largeur );
    }

    public static ArrayList <Rectangle> getInstances(){
        return instances;
    }
```

EXAM

CODE EXPLANATION



```java
public double surface (){
        return longueur*largeur;
    }

    public double getLongueur() {
        return longueur;
    }

    public double getLargeur() {
        return largeur;
    }

    public void setLongueur (double lon) {
        longueur = lon;
    }
    public void setLargeur (double larg) {
        largeur = larg;
    }
}
```

👈 قمنا بانشاء الدالة **surface التي تقوم بحساب المساحة.**

👈 قمنا بانشاء الدوال **Getter و Setter** للمتغيرات.

EXAM

CODE EXPLANATION

👈 قمنا بانشاء الكلاس Carre الذي يرث جميع الMethods و المتغيرات الموجودة (public / protected) بداخل الRectangle class

👈 قمنا بانشاء الConstructor الخاص بCarre class.

```java
public class Carré extends Rectangle {

    public Carré (double cote) {
        super(cote, cote);
    }

    public void affiche () {
        System.out.println(" Carré la longueur du cote: "+ getLargeur () );
    }

    public static ArrayList <Rectangle> getInstances(){

        ArrayList <Rectangle> instancesCarre =new ArrayList <Rectangle>();

        ArrayList <Rectangle> instances = Rectangle.getInstances();

        for (int i=0;i<instances.size();i++){

            if(instances.get(i) instanceof Carré){

                instancesCarre.add(instances.get(i));

            }
        }
        return instancesCarre;
    }

```

EXAM

CODE EXPLANATION

👈 **Super(cote ,cote) تقوم بنداء** ال**Constructor** الخاص ب**Rectangle class**, بمعنى اخر **عند صناعة** **Carre instance** (صناعة **Carre**ل **Object**) **يتم تنفيذ ال****Constructor** **الخاص ب** **Carre class** **مباشرة** و يقوم هذا الاخير **بتنفيذ ال****Constructor** **الخاص ب** **Rectangle class** بواسطة **Super** من **اجل اضافة** **هذا ال****Carre instance** **الجديد** الى داخل **ال****ArrayList instances** **وحفظه.**

```java
public class Carré extends Rectangle {

    public Carré (double cote) {
        super(cote, cote);
    }

    public void affiche () {
        System.out.println(" Carré la longueur du cote: "+ getLargeur () );
    }

    public static ArrayList <Rectangle> getInstances(){

        ArrayList <Rectangle> instancesCarre =new ArrayList <Rectangle>();

        ArrayList <Rectangle> instances = Rectangle.getInstances();

        for (int i=0;i<instances.size();i++){

            if(instances.get(i) instanceof Carré){

                instancesCarre.add(instances.get(i));

            }
        }
        return instancesCarre;
    }

```

EXAM

CODE EXPLANATION

👈 قمنا بانشاء الدالة **getInstances** التي تقوم بارجاع ال **ArrayList** المحمل ب **Carre instances**.

```java
public class Carré extends Rectangle {

    public Carré (double cote) {
        super(cote, cote);
    }

    public void affiche () {
        System.out.println(" Carré la longueur du cote: "+ getLargeur () );
    }

    public static ArrayList <Rectangle> getInstances(){

        ArrayList <Rectangle> instancesCarre =new ArrayList <Rectangle>();

        ArrayList <Rectangle> instances = Rectangle.getInstances();

        for (int i=0;i<instances.size();i++){

            if(instances.get(i) instanceof Carré){

                instancesCarre.add(instances.get(i));

            }
        }
        return instancesCarre;
    }

```

EXAM

CODE EXPLANATION

👈 قمنا بانشاء ال**ArrayList instancesCarre** من نوع ال**Rectangle** الذي سيحمل **Carre instances**.

👈 قمنا بانشاء **ArrayList instances** من النوع **Rectangle** و قمنا **بجلب جميع ال**instances **الموجودة في**

Rectangle ArrayList instances بواسطة ال Method Static getInstance

التي تم انشائها في ال Rectangle class ووضعها **بداخله**.

```java
public class Carré extends Rectangle {

    public Carré (double cote) {
        super(cote, cote);
    }

    public void affiche () {
        System.out.println(" Carré la longueur du cote: "+ getLargeur () );
    }

    public static ArrayList <Rectangle> getInstances(){

        ArrayList <Rectangle> instancesCarre =new ArrayList <Rectangle>();

        ArrayList <Rectangle> instances = Rectangle.getInstances();

        for (int i=0;i<instances.size();i++){

            if(instances.get(i) instanceof Carré){

                instancesCarre.add(instances.get(i));

            }
        }
        return instancesCarre;
    }

```

EXAM

CODE EXPLANATION

👈 هنا **تكمن فائدة ال**static Methods / Variables**,** لو لم تكن ال**Method getInstance** من نوع **static** **سيتوجب علينا** حينها **صناعة** Object خاص **بكلاس** Rectangle ا(Rectangle instance) **داخل كلاس** Carre من اجل **استخدام هذه ال**Method و كما قلنا **عند صناعة** Rectangle instance **سيتوجب علينا اعطاء قيم لل**Constructor ا(new Rectangle **(1,4)**) مما يؤدي الى **تنفيذ ال**Constructor الخاص ب Rectangle وبالتالي **اضافة** Rectangle instance **جديد** داخل ال**ArrayList instances** و هذا ال**Rectangle instance** مصنوع من طرف ال**Carre class** وليس ال**Main class.**

👈 **اي كلما استخدمنا ال**Method getInstance الخاصة **بكلاس** Carre **سيتم صناعة** Rectangle instance **جديد تلقائيا و هذا ما يسبب** ارتفاع ضغط المبرمج و قلقه **بسبب زيادة عدد ال**Rectangle instance **بدون معرفة السبب :) .**

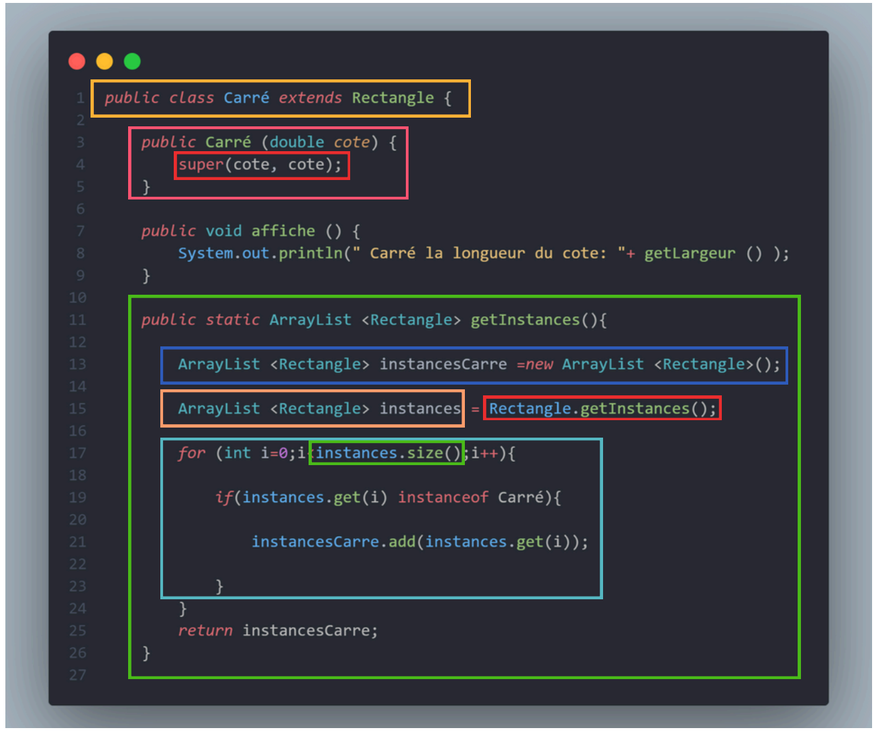
```java
public class Carré extends Rectangle {

    public Carré (double cote) {
        super(cote, cote);
    }

    public void affiche () {
        System.out.println(" Carré la longueur du cote: "+ getLargeur () );
    }

    public static ArrayList <Rectangle> getInstances(){

        ArrayList <Rectangle> instancesCarre =new ArrayList <Rectangle>();

        ArrayList <Rectangle> instances = Rectangle.getInstances();

        for (int i=0;i<instances.size();i++){

            if(instances.get(i) instanceof Carré){

                instancesCarre.add(instances.get(i));

            }
        }
        return instancesCarre;
    }

```

EXAM

CODE EXPLANATION

👈 انشائنا ال**For loop** من اجل **فصل** ال **Rectangle instances** عن ال **Carre instances** الموجودين بداخل ال **ArryList instances**.

👈 **instances.size()** من اجل معرفة **طول او عدد ال instances** الموجودين بداخل ال **ArrayList instances**.



```java
public class Carré extends Rectangle {

    public Carré (double cote) {
        super(cote, cote);
    }

    public void affiche () {
        System.out.println(" Carré la longueur du cote: "+ getLargeur () );
    }

    public static ArrayList <Rectangle> getInstances(){

        ArrayList <Rectangle> instancesCarre =new ArrayList <Rectangle>();

        ArrayList <Rectangle> instances = Rectangle.getInstances();

        for (int i=0;i<instances.size();i++){

            if(instances.get(i) instanceof Carré){

                instancesCarre.add(instances.get(i));

            }
        }
        return instancesCarre;
    }

```

EXAM

CODE EXPLANATION

👈 instance.get(i) تقوم بجلب ال(instance) Object ذو ال **index i**.

👈 instanceof هي **كلمة محجوزة في جافا** تقوم **بالتحقق من نوعية ال**Object **او ال**instance ما اذا **كان تابعا لل Class** ام لا.

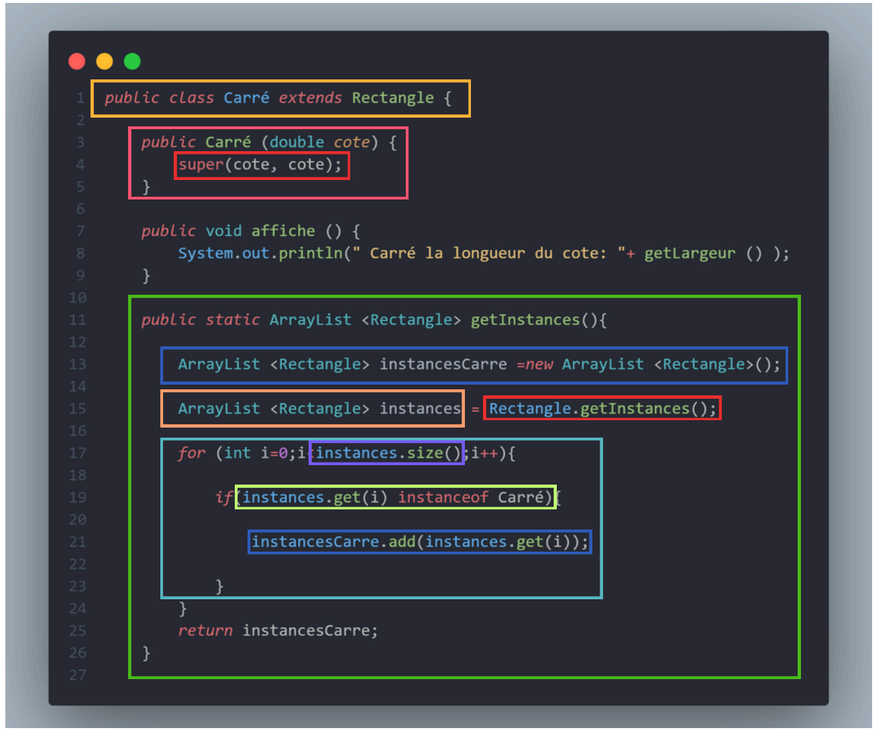
👈 instance.get(i) instanceof Carre تقوم بالتحقق من **ال**instance ذو ال **index i** الذي بداخل **ال**ArrayList instances تابع ل Carre class ام لا.

```java
public class Carré extends Rectangle {

    public Carré (double cote) {
        super(cote, cote);
    }

    public void affiche () {
        System.out.println(" Carré la longueur du cote: "+ getLargeur () );
    }

    public static ArrayList <Rectangle> getInstances(){

        ArrayList <Rectangle> instancesCarre =new ArrayList <Rectangle>();

        ArrayList <Rectangle> instances = Rectangle.getInstances();

        for (int i=0;i<instances.size();i++){

            if(instances.get(i) instanceof Carré){

                instancesCarre.add(instances.get(i));

            }
        }
        return instancesCarre;
    }

}
```

EXAM

CODE EXPLANATION

👈 اذا كان ال**instance** ذو ال **index i** الذي بداخل ال**ArrayList instances** تابع ل **Carre class** فاننا نقوم بوضعه بداخل ال**ArrayList instancesCarre**.



```java
public class Carré extends Rectangle {

    public Carré (double cote) {
        super(cote, cote);
    }

    public void affiche () {
        System.out.println(" Carré la longueur du cote: "+ getLargeur () );
    }

    public static ArrayList <Rectangle> getInstances(){

        ArrayList <Rectangle> instancesCarre =new ArrayList <Rectangle>();

        ArrayList <Rectangle> instances = Rectangle.getInstances();

        for (int i=0;i<instances.size();i++){

            if(instances.get(i) instanceof Carré){

                instancesCarre.add(instances.get(i));

            }
        }
        return instancesCarre;
    }

}
```

EXAM

CODE EXPLANATION

👈 في الاخير نقوم بارجاع ال ArrayList instancesCarre لان الMethod من نوع get.

```java
public class Carré extends Rectangle {

    public Carré (double cote) {
        super(cote, cote);
    }

    public void affiche () {
        System.out.println(" Carré la longueur du cote: "+ getLargeur () );
    }

    public static ArrayList <Rectangle> getInstances(){

        ArrayList <Rectangle> instancesCarre =new ArrayList <Rectangle>();

        ArrayList <Rectangle> instances = Rectangle.getInstances();

        for (int i=0;i<instances.size();i++){

            if(instances.get(i) instanceof Carré){

                instancesCarre.add(instances.get(i));

            }
        }
        return instancesCarre;
    }

}
```

EXAM

CODE EXPLANATION

👈 قمنا بانشاء Array myInstances ذو النوع Rectangle اي سيحمل بداخل الinstances **وحددنا طوله ب 4.**

👈 قمنا بانشاء For loop التي تقوم التحقق ما اذا كان **ال i فردي ام زوجي.**

```java
public class Main {
    public static void main(String[] args) {

        Rectangle myInstances[]=new Rectangle [4];

        for (int i=0; i < myInstances.length ; i++){

            if(i % 2 == 0){

                myInstances[i]= new Rectangle (Math.random() ,Math.random());

            }else {

                myInstances[i]= new Carré (Math.random());

            }
        }
        System.out.println("Les Carrés: "+Carré.getInstances().size());
        System.out.println("Les Rectangles: "+Rectangle.getInstances().size());

    }
}
```

EXAM

CODE EXPLANATION

👈 في حالة ما كان **ال i زوجي** نقوم **بصناعة ال** Rectangle instance اما في حالة **الفردي** نقوم **بصناعة ال** Carre instance.

```java
public class Main {
    public static void main(String[] args) {

        Rectangle myInstances[]=new Rectangle [4];

        for (int i=0; i < myInstances.length ; i++){

            if(i % 2 == 0){

                myInstances[i]= new Rectangle (Math.random() ,Math.random());

            }else {

                myInstances[i]= new Carré (Math.random());

            }
        }
        System.out.println("Les Carrés: "+Carré.getInstances().size());
        System.out.println("Les Rectangles: "+Rectangle.getInstances().size());

    }
}
```

EXAM

CODE EXPLANATION

👈 هنا ايضا **تكمن فائدة ال**static , كما قلنا لو لم تكن الMethod getInstances **من نوع** static سيتوجب علينا **صناعة Object l(instance) للاستخدامها** و عند **صناعة** instance **سينفذ ال**Constructor و سيتم **اضافة هذا ال**instance الى داخل الArrayList instances و في هذه الحالة سيكون لدينا **6 instances** و**ليس 4** (1 Rectangle instance و 1 Carre instance).

```java
public class Main {
    public static void main(String[] args) {

        Rectangle myInstances[]=new Rectangle [4];

        for (int i=0; i < myInstances.length ; i++){

            if(i % 2 == 0){

                myInstances[i]= new Rectangle (Math.random() ,Math.random());

            }else {

                myInstances[i]= new Carré (Math.random());

            }
        }
        System.out.println("Les Carrés: "+Carré.getInstances().size());
        System.out.println("Les Rectangles: "+Rectangle.getInstances().size());

    }
}
```

EXAM

CODE EXPLANATION

👈 سيتم طباعة عدد الinstances الموجودة داخل **ArrayList instancesCarre** و داخل **ArrayList instances**.

```java
public class Main {
    public static void main(String[] args) {

        Rectangle myInstances[]=new Rectangle [4];

        for (int i=0; i < myInstances.length ; i++){

            if(i % 2 == 0){

                myInstances[i]= new Rectangle (Math.random() ,Math.random());

            }else {

                myInstances[i]= new Carré (Math.random());

            }
        }
        System.out.println("Les Carrés: "+Carré.getInstances().size());
        System.out.println("Les Rectangles: "+Rectangle.getInstances().size());

    }
}
```

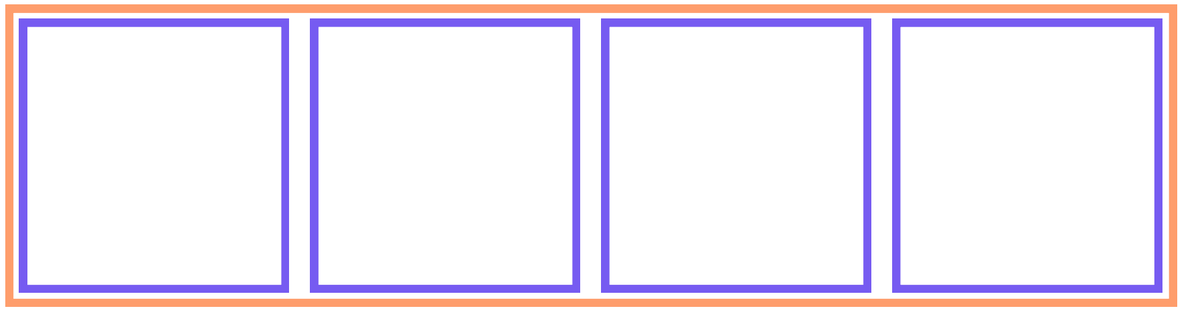
EXAM

BEHIND THE SCENES

👈 قمنا بانشاء **Array myInstances** ذو النوع **Rectangle** اي سيحمل بداخل ال**instances** و **حددنا طوله ب 4.**

```
Rectangle myInstances [ ] = new Rectangle [4];
```

myInstances



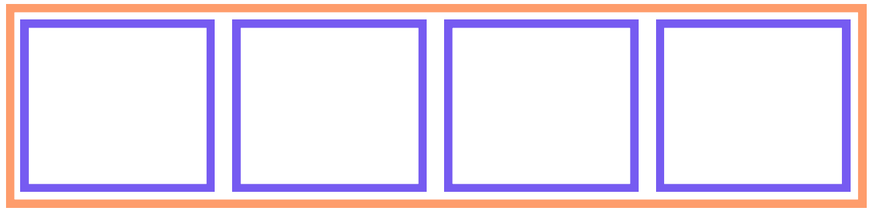
EXAM

BEHIND THE SCENES

👈 **طول ال**Array myInstances هو **4** اذا سيتم تنفيذ الكود الذي بداخل ال**For loop** اربع مرات.

```
for (int i = 0; i < myInstances.length ; i++){
      if(i % 2 == 0){
        myInstances[ i ]= new Rectangle (2 ,5);
      }else {
        myInstances[ i ] = new Carré (4);
     }
   }
```

myInstances



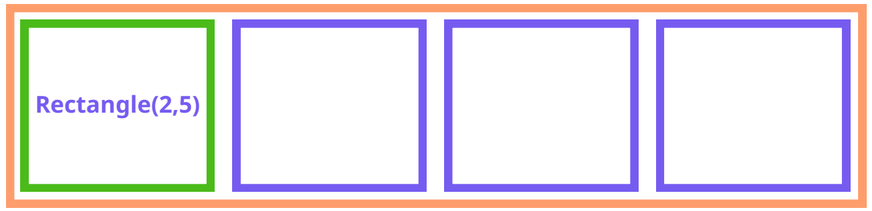
EXAM

BEHIND THE SCENES

EXECUTION OF FOR LOOP 01 : **i = 0**

```
for (int i = 0; i < myInstances.length ; i++){
      if(i % 2 == 0){
        myInstances[ 0 ]= new Rectangle (2 ,5);
      }else {
        myInstances[ i ] = new Carré (4);
     }
   }
```

myInstances



EXAM

BEHIND THE SCENES

👈 عند صناعة Rectangle instance سيتم تنفيذ الConstructor الخاص بRectangle مما يؤدي الى اضافة هذا الinstance الى ArrayList instances الموجود بالكلاس Rectangle.

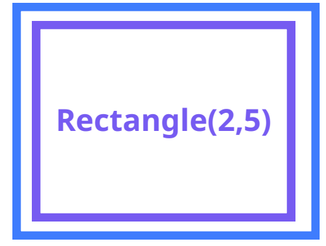
```
myInstances[ 0 ]= new Rectangle (2 ,5);
```

Rectangle(2,5)

```
public class Rectangle{
  private static ArrayList <Rectangle> instances = new ArrayList <Rectangle>();
  public Rectangle (2, 5){
    longueur = 2;
    largeur = 5;
    instances.add(this);
  }
```

instances

instances



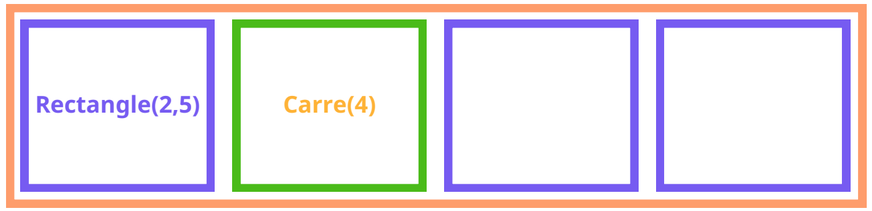
EXAM

BEHIND THE SCENES

EXECUTION OF FOR LOOP 02 : **i = 1**

```
for (int i = 0; i < myInstances.length ; i++){
      if(i % 2 == 0){
        myInstances[ i ]= new Rectangle (2 ,5);
      }else {
        myInstances[ 1 ] = new Carré (4);
     }
   }
```

myInstances



EXAM

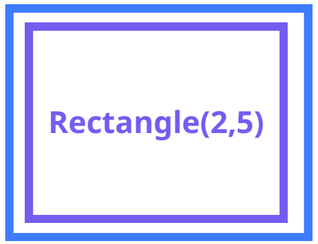
BEHIND THE SCENES

👈 عند صناعة Caree instance سيتم تنفيذ الConstructor الخاص بCarre مما يؤدي الى تنفيذ ايضا الConstructor الخاص بكلاس Rectangle بسبب Super(4,4) مما يؤدي الى اضافة هذا الinstance الى ArrayList instances الموجود بالكلاس Rectangle.

```
myInstances[ 1 ]= new Carre(4);
```


```
public class Carré extends Rectangle {

  public Carré (4) {
    super(4, 4);
  }
```

instances



EXAM

BEHIND THE SCENES

👈 يتم اضافة هذا الinstance الى ArrayList instances الموجود بالكلاس Rectangle.

```
myInstances[ 1 ]= new Carre(4);
```

```
public class Rectangle{
  private static ArrayList <Rectangle> instances = new ArrayList <Rectangle>();
  public Rectangle (4, 4){
    longueur = 4;
    largeur = 4;
    instances.add(this);
  }
```

instances

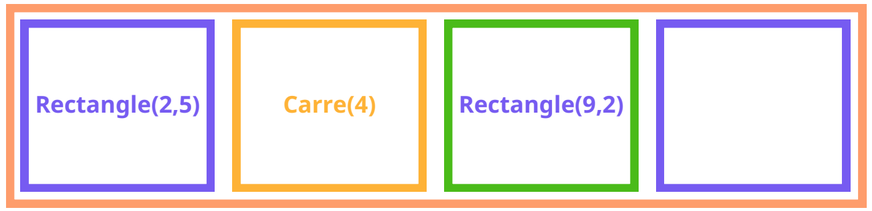
EXAM

BEHIND THE SCENES

EXECUTION OF FOR LOOP 03 : **i = 2**

```
for (int i = 0; i < myInstances.length ; i++){
      if(i % 2 == 0){
        myInstances[ 2 ]= new Rectangle (9 ,2);
      }else {
        myInstances[ i ] = new Carré (4);
    }
  }
```

myInstances



EXAM

BEHIND THE SCENES

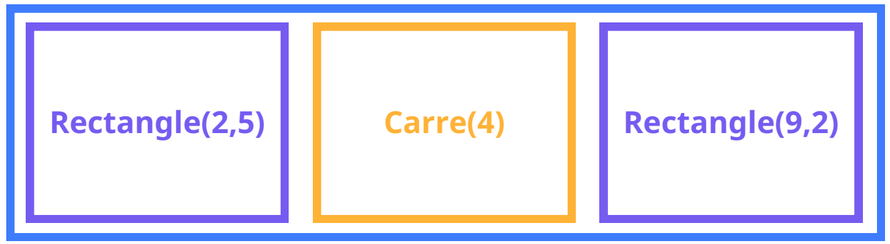
👈 عند صناعة Rectangle instance سيتم تنفيذ الConstructor الخاص بRectangle مما يؤدي الى اضافة هذا الinstance الى ArrayList instances الموجود بالكلاس Rectangle.

```
myInstances[ 2 ]= new Rectangle (9 ,2);
```

Rectangle(9,2)

```
public class Rectangle{
  private static ArrayList <Rectangle> instances = new ArrayList <Rectangle>();
  public Rectangle (9, 2){
    longueur = 9;
    largeur = 2;
    instances.add(this);
  }
```

instances



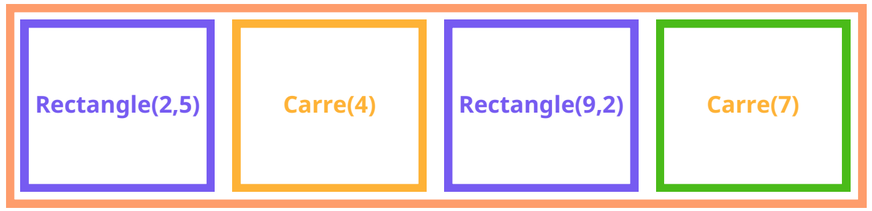
EXAM

BEHIND THE SCENES

EXECUTION OF FOR LOOP 04 : **i = 3**

```
for (int i = 0; i < myInstances.length ; i++){
        if(i % 2 == 0){
          myInstances[ i ]= new Rectangle (9 ,2);
        }else {
          myInstances[ 3 ] = new Carré (7);
      }
    }
```

myInstances



EXAM

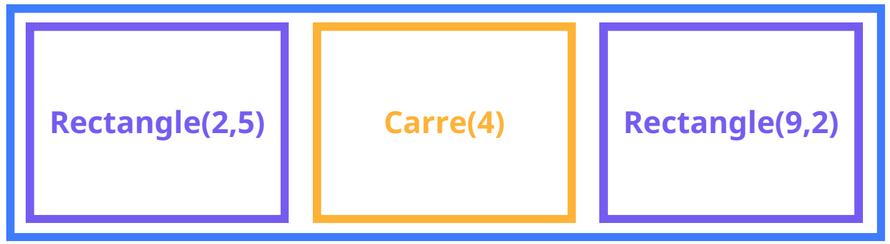
BEHIND THE SCENES

👈 عند صناعة Caree instance سيتم تنفيذ الConstructor الخاص بCarre مما يؤدي الى تنفيذ ايضا الConstructor الخاص بكلاس Rectangle بسبب Super(7,7) مما يؤدي الى اضافة هذا الinstance الى ArrayList instances الموجود بالكلاس Rectangle.

```
myInstances[ 3 ]= new Carre(7);
```


```
public class Carré extends Rectangle {

  public Carré (7) {
    super(7, 7);
  }
```

instances



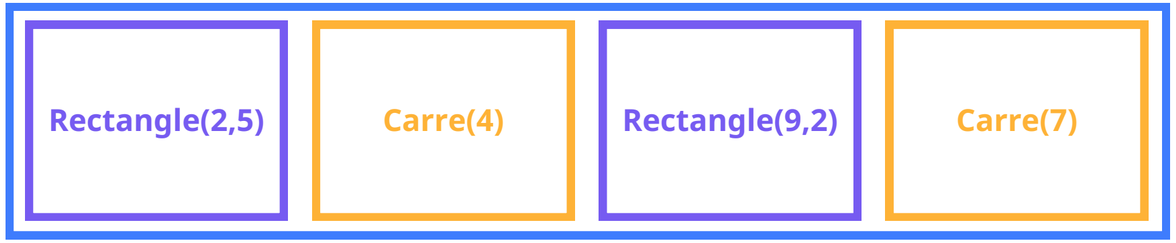
EXAM

BEHIND THE SCENES

👈 يتم اضافة هذا الinstance الى ArrayList instances الموجود بالكلاس Rectangle.

```
myInstances[ 1 ]= new Carre(7);


public class Rectangle{
  private static ArrayList <Rectangle> instances = new ArrayList <Rectangle>();
  public Rectangle (7, 7){
    longueur = 7;
    largeur = 7;
    instances.add(this);
  }
```

instances



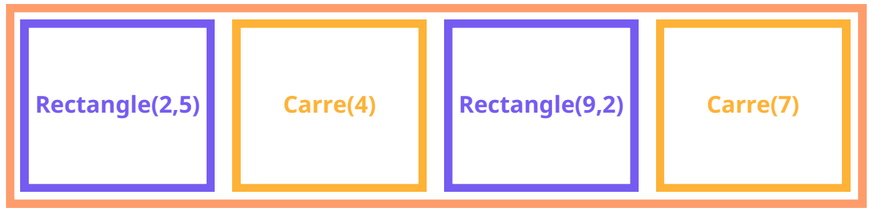
EXAM

BEHIND THE SCENES

EXECUTION OF FOR LOOP 05 : **i = 4 ? impossible -> 4 < 4**

```
for (int i = 0; i < myInstances.length ; i++){
      if(i % 2 == 0){
        myInstances[ i ]= new Rectangle (2 ,5);
      }else {
        myInstances[ 3 ] = new Carré (4);
     }
   }
```

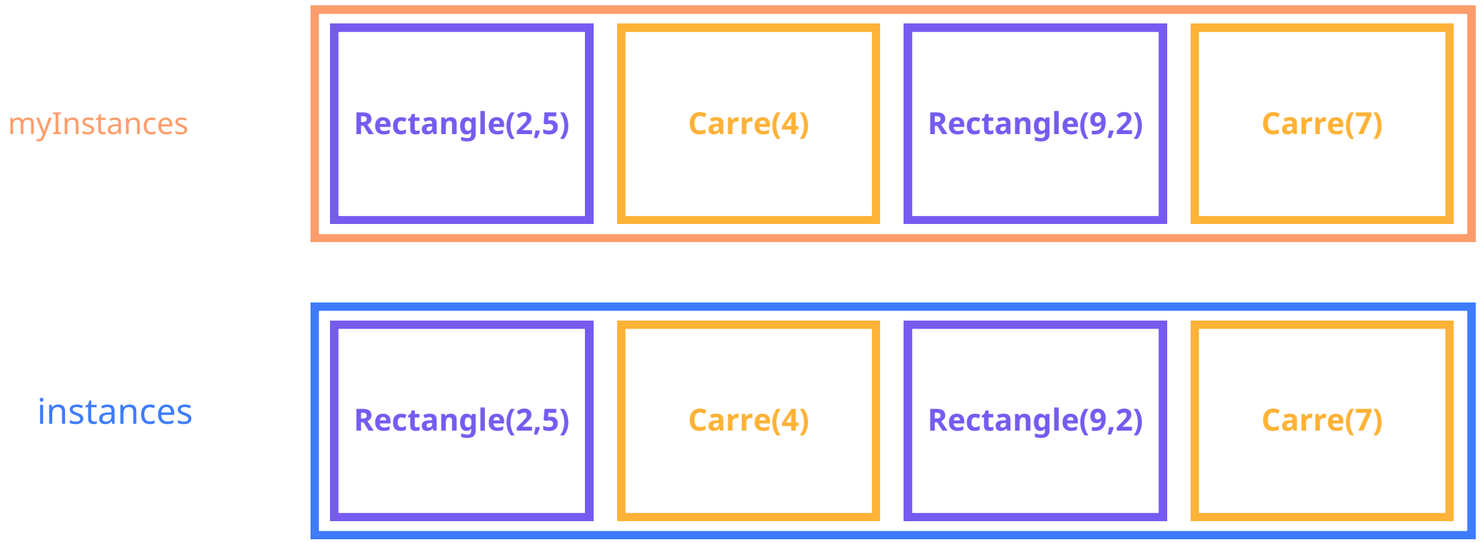
myInstances



EXAM

BEHIND THE SCENES

👈في نهاية الFor loop يتكون لنا الArray myInstances و ArrayList instances الذي يحتويان على 4 instances



EXAM

BEHIND THE SCENES

```
System.out.println("Les Carrés: "+Carré.getInstances().size());


class Carre {
  public static ArrayList <Rectangle> getInstances(){
    ArrayList <Rectangle> instancesCarre =new ArrayList <Rectangle>();
    ArrayList <Rectangle> instances = Rectangle.getInstances();
    for (int i=0;i<instances.size();i++){
      if(instances.get(i) instanceof Carré){
        instancesCarre.add(instances.get(i));
      }
    }
    return instancesCarre;
  }
```

👈سيتم منادة الميثود **getInstances** الخاصة بكلاس **Carre**

👈 **بعد ذلك سيتم صنع ArrayList instanceCarre**

EXAM

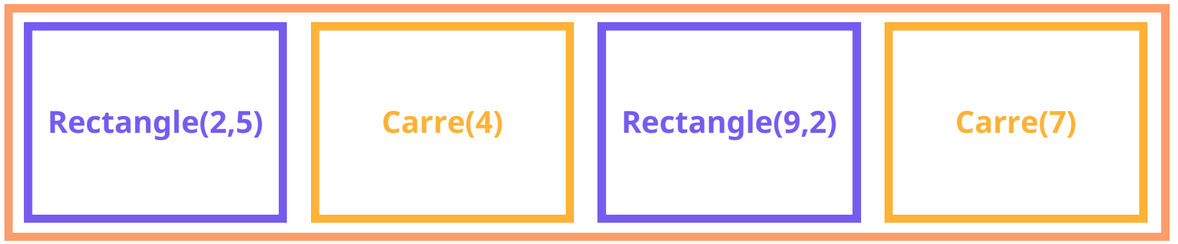
BEHIND THE SCENES

```
System.out.println("Les Carrés: "+Carré.getInstances().size());


class Carre {
  public static ArrayList <Rectangle> getInstances(){
    ArrayList <Rectangle> instancesCarre =new ArrayList <Rectangle>();
    ArrayList <Rectangle> instances = Rectangle.getInstances();
    for (int i=0;i<instances.size();i++){
      if(instances.get(i) instanceof Carré){
        instancesCarre.add(instances.get(i));
      }
    }
    return instancesCarre;
  }
```

سيتم صنع ال**ArrayList instance** ووضع داخله كل ال**instances** الموجودين داخل **ArrayList instances** بداخل الكلاس **Rectangle**

instancesCarre

instances

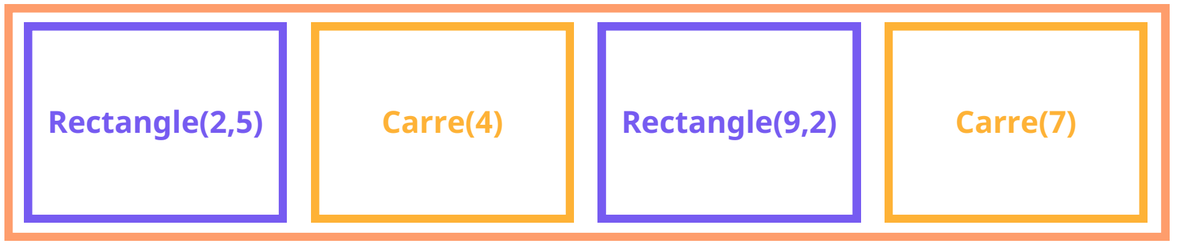


EXAM

BEHIND THE SCENES

👈 **طول ال**ArrayList instances هو **4** اذا سيتم تنفيذ الكود الذي بداخل ال**For loop** اربع مرات.

```
for (int i=0;i<instances.size();i++){

 if(instances.get(i) instanceof Carré){

    instancesCarre.add(instances.get(i));

 }
```



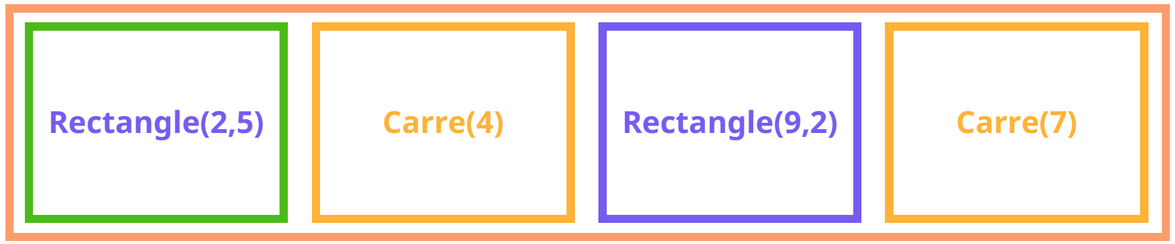
EXAM

BEHIND THE SCENES

EXECUTION OF FOR LOOP 01 : **i = 0**

```
for (int i=0;i<instances.size();i++){

 if(instances.get(0) instanceof Carré){

    instancesCarre.add(instances.get(i));

 }
```

instances



لا ينتمي الى الكلاس Carre

instancesCarre

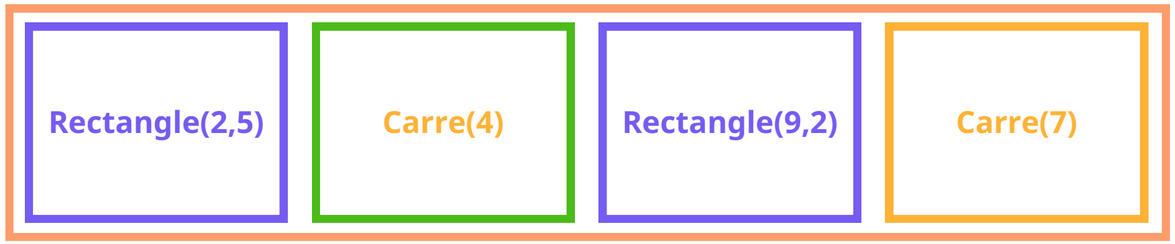
EXAM

BEHIND THE SCENES

EXECUTION OF FOR LOOP 02 : **i = 1**

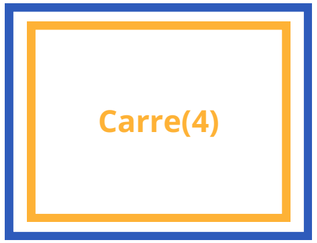
```
for (int i=0;i<instances.size();i++){

 if(instances.get(1) instanceof Carré){

    instancesCarre.add(instances.get(1));

 }
```

instances



ينتمي الى الكلاس Carre

instancesCarre



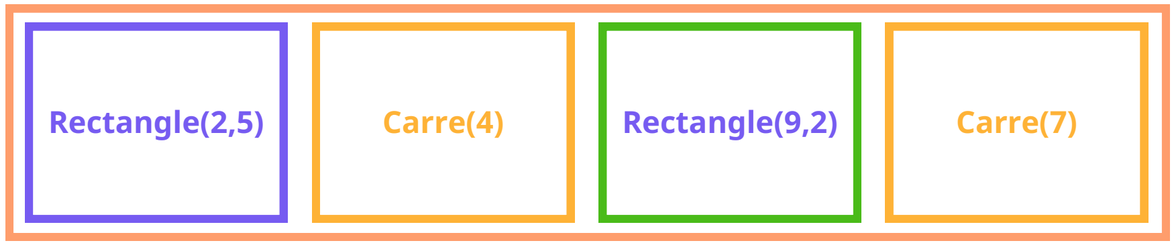
EXAM

BEHIND THE SCENES

EXECUTION OF FOR LOOP 03 : **i = 2**

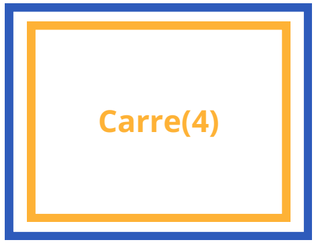
```
for (int i=0;i<instances.size();i++){

 if(instances.get(2) instanceof Carré){

    instancesCarre.add(instances.get(i));

 }
```

instances



لا ينتمي الى الكلاس Carre

instancesCarre



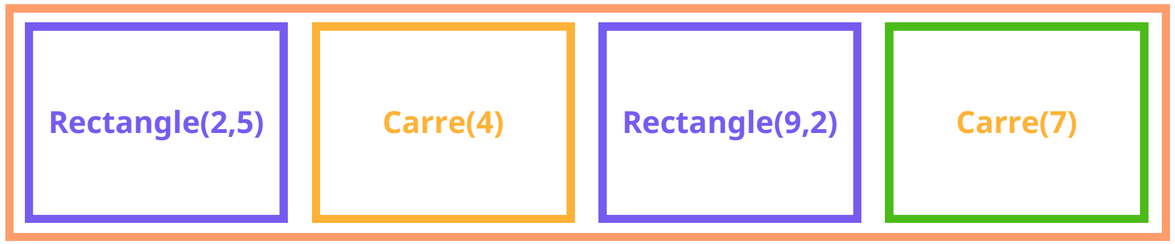
EXAM

BEHIND THE SCENES

EXECUTION OF FOR LOOP 04 : **i = 3**

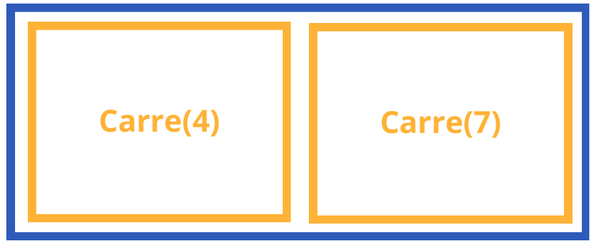
```
for (int i=0;i<instances.size();i++){

 if(instances.get(1) instanceof Carré){

    instancesCarre.add(instances.get(1));

 }
```

instances



ينتمي الى الكلاس Carre

instancesCarre



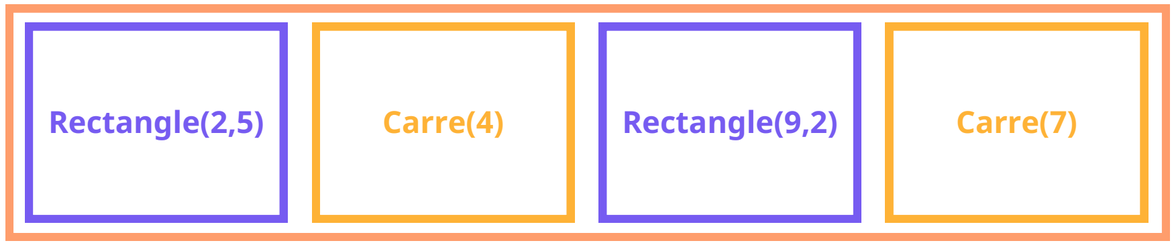
EXAM

BEHIND THE SCENES

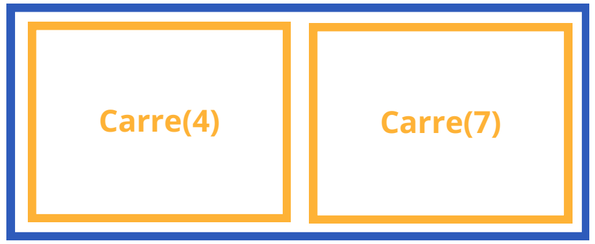
EXECUTION OF FOR LOOP 05 : **i = 4 impossible -> 4 < 4!**

```
for (int i=0;i<instances.size();i++){

 if(instances.get(1) instanceof Carré){

    instancesCarre.add(instances.get(1));

 }
```

instances



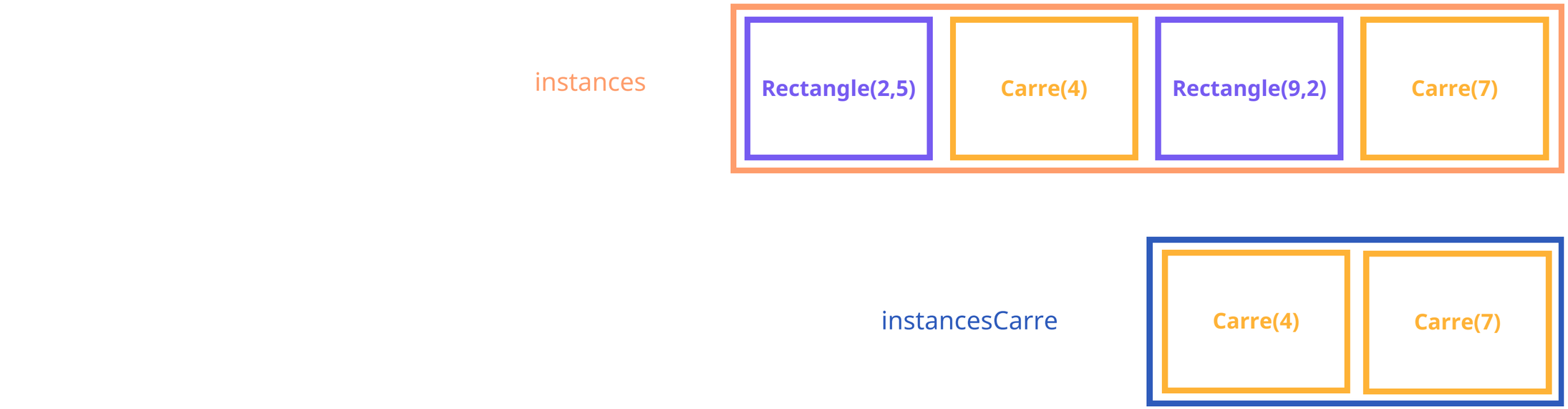
instancesCarre



EXAM

BEHIND THE SCENES

👈في نهاية ال**For loop** يتكون لنا ال**ArrayList instancesCarre** الذي يحتوي على 2 **instances.**

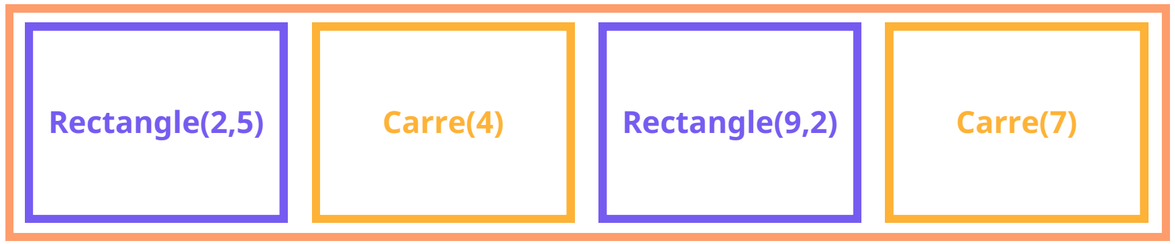


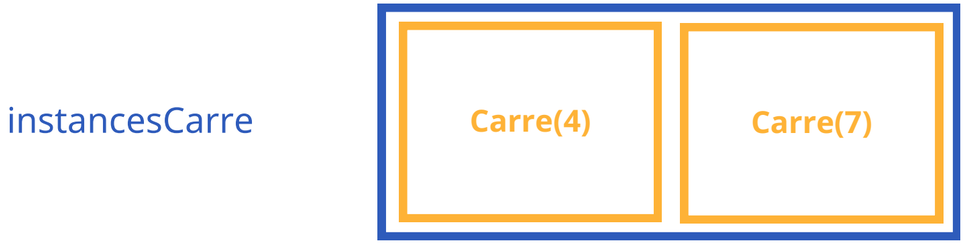
EXAM

BEHIND THE SCENES

```
class Carre {
  public static ArrayList <Rectangle> getInstances(){
    ArrayList <Rectangle> instancesCarre =new ArrayList <Rectangle>();
    ArrayList <Rectangle> instances = Rectangle.getInstances();
    for (int i=0;i<instances.size();i++){
      if(instances.get(i) instanceof Carré){
        instancesCarre.add(instances.get(i));
      }
    }
    return instancesCarre;
  }
```

سيتم ارجاع instanceCarre من .getInstances





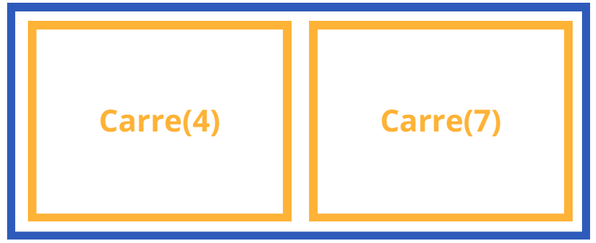
EXAM

BEHIND THE SCENES

👈طول ال **instanceCarre هو 2.**

```
System.out.println("Les Carrés: "+Carré.getInstances().size());
```

Les Carres : 2



EXAM

BEHIND THE SCENES

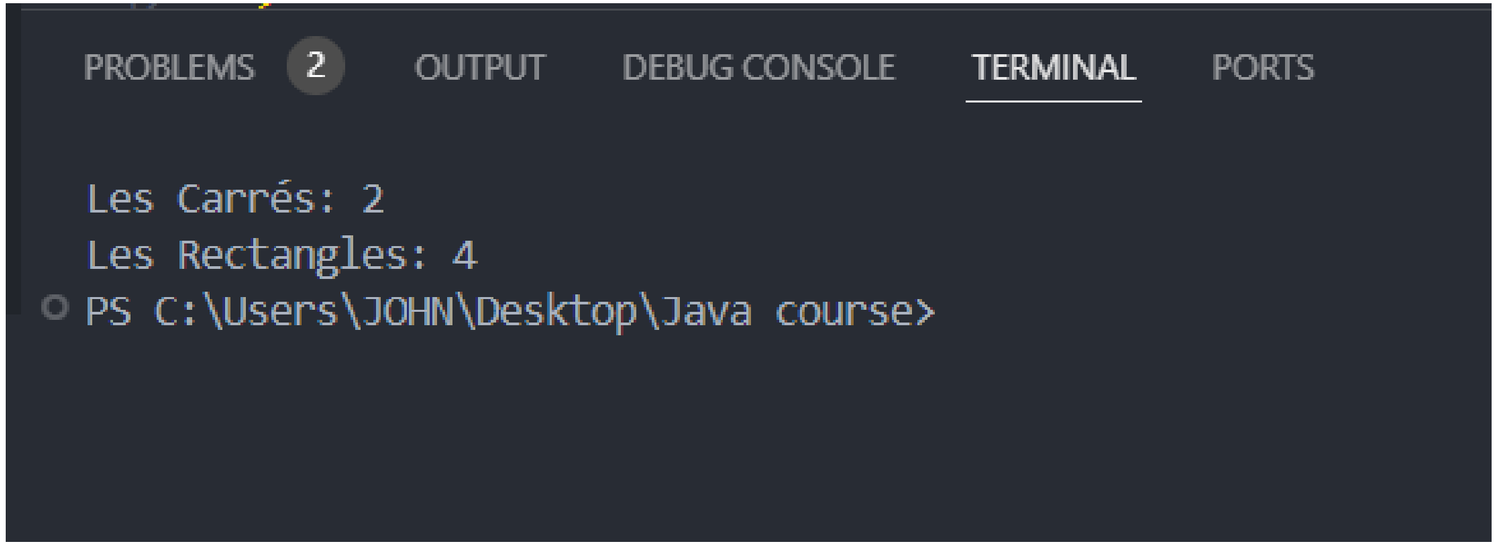
👈طول الinstances **هو 4** .

```
System.out.println("Les Rectangles: "+Rectangle.getInstances().size());
```

Les Rectangles : 4

EXAM

OUTPUT


PROBLEMS 2
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Les Carrés: 2
Les Rectangles: 4
PS C:\Users\JOHN\Desktop\Java course>

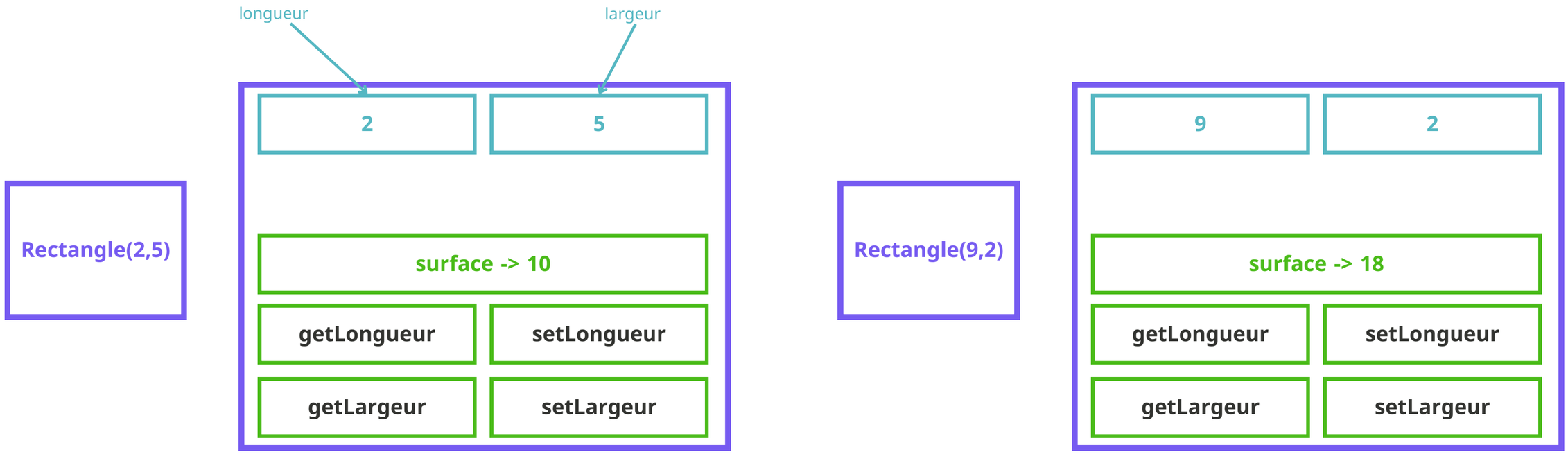
EXAM

OUTPUT

EXAM

ZOOM IN

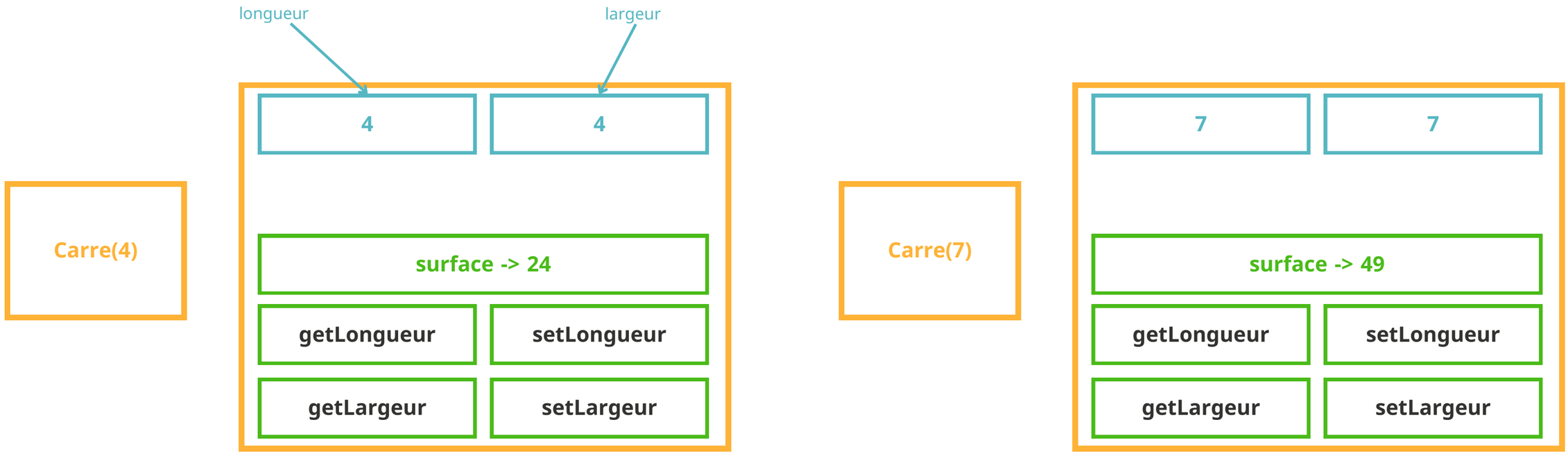
👈نظرة عن قرب لل **instances**.



EXAM

ZOOM IN

👈نظرة عن قرب لل **instances**.



EXAM

CONCLUSION

👈 كل شيء في جافا عبارة عن **object.**

THE END

ANY feedback?

Mehdi Bouchachi

SEE YOU SOON...