

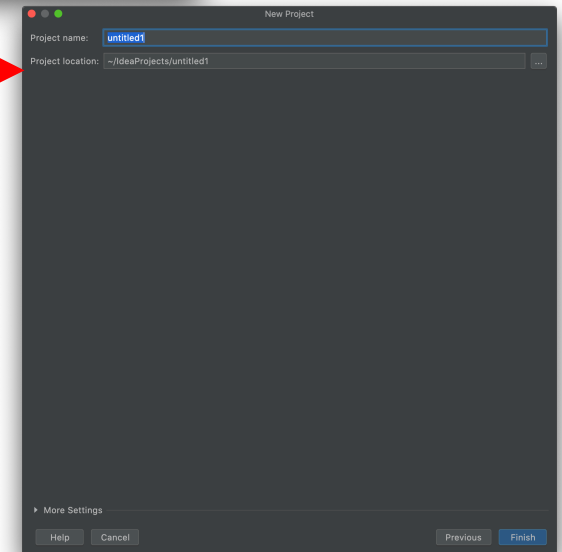
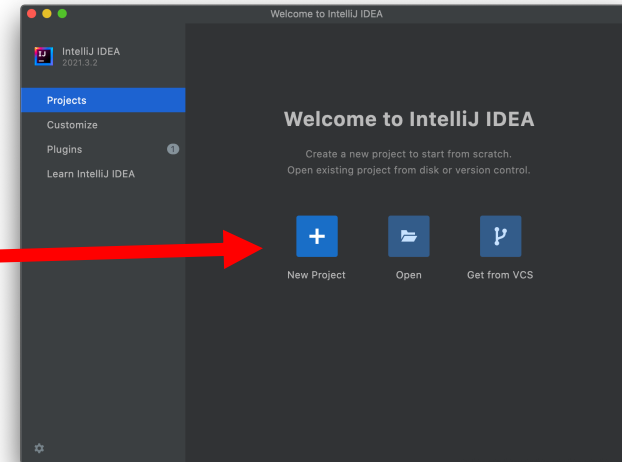
Programmazione 2

Esercitazione 1

vedi anche: <https://www.jetbrains.com/help/idea/working-with-source-code.html>

Nuovo progetto:

- 1) New Project
- 2) Next
- 3) Next
- 4) Insert Project Name (e.g. "MyFirstProject")
- 5) Finish

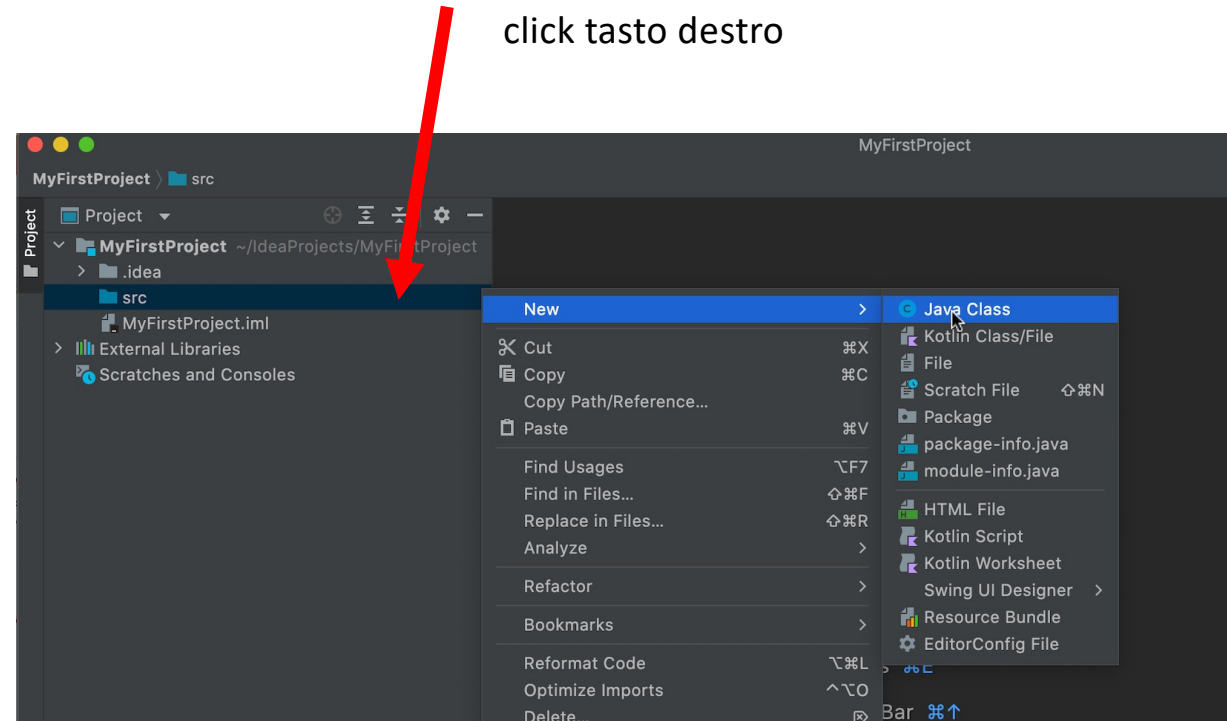


Create new class

- src->New->Java Class

oppure:

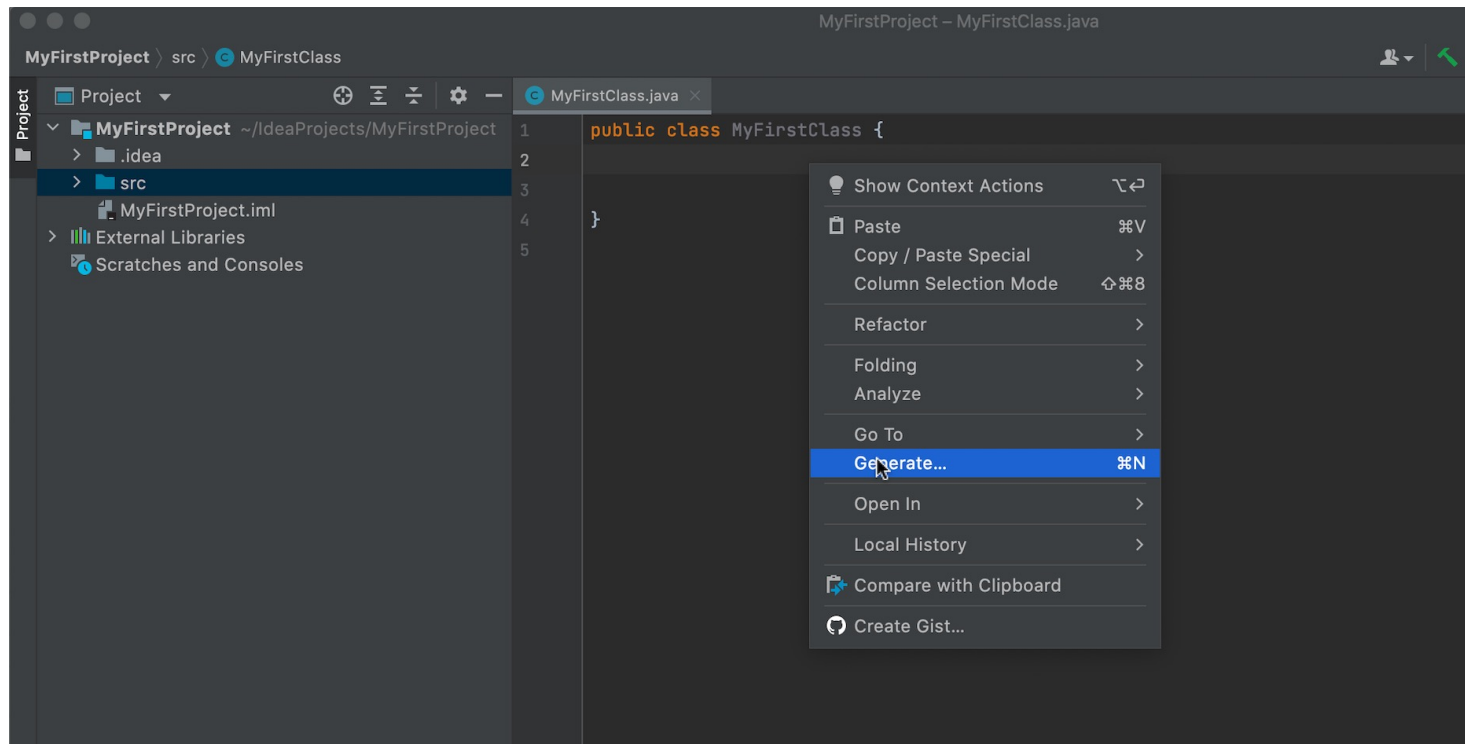
- File ->New->Java Class



then: specify the name (e.g. "MyFirstClass").

Creiamo il costruttore

- Click con tasto destro, oppure Command-N:
generate constructor



Aggiungiamo il main ed eseguiamo

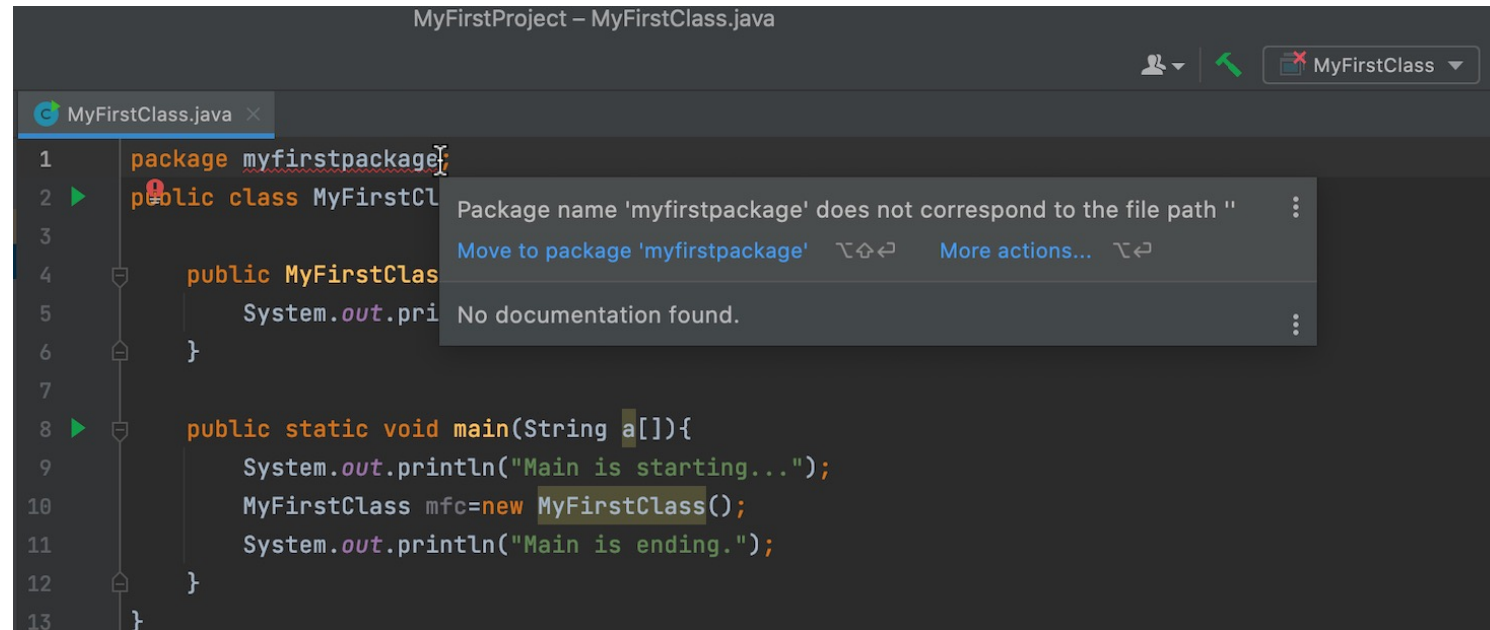
```
MyFirstClass.java x
1  ▶ public class MyFirstClass {
2
3      public MyFirstClass() {
4          System.out.println("Constructor called");
5      }
6
7  ▶ public static void main(String a[]){
8      System.out.println("Main is starting...");
9      MyFirstClass mfc=new MyFirstClass();
10     System.out.println("Main is ending.");
11 }
12 }
13
```

tasto destro: "run MyFirstClass"

```
Run: MyFirstClass x
▶ ↑ /Users/ronchet/Library/Java/JavaVirtualMach
⚙ ↓ Main is starting...
☐ ↺ Constructor called
📷 ⬇ Main is ending.
🔍 🖨 Process finished with exit code 0
🗑
```

abbiamo dimenticato il package!

proviamo ad aggiungerlo...

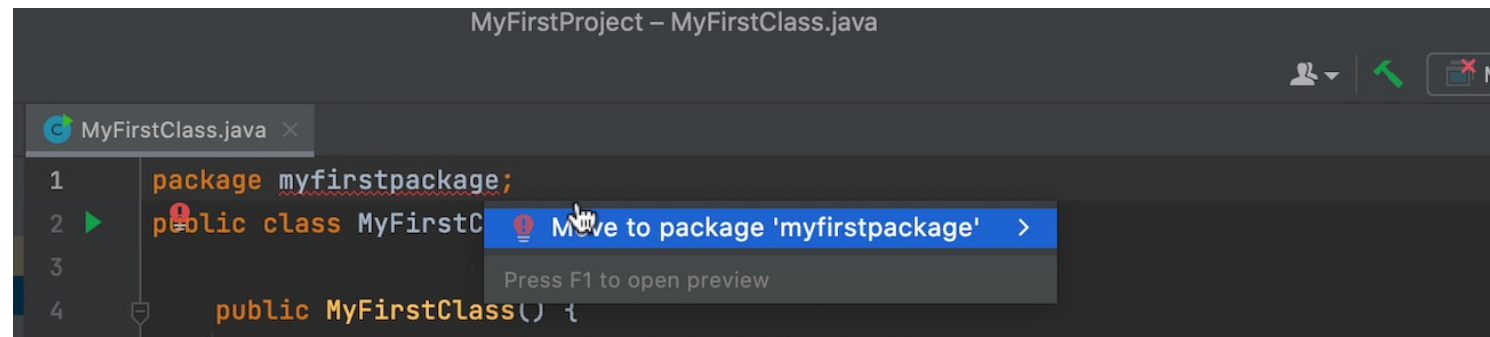


The screenshot shows an IDE window titled "MyFirstProject - MyFirstClass.java". The code in the editor is as follows:

```
1 package myfirstpackage;  
2 public class MyFirstClass {  
3  
4     public MyFirstClass() {  
5         System.out.println("Main is starting...");  
6     }  
7  
8     public static void main(String a[]){  
9         System.out.println("Main is starting...");  
10        MyFirstClass mfc=new MyFirstClass();  
11        System.out.println("Main is ending.");  
12    }  
13 }
```

A red squiggly line is under the package declaration on line 1. A tooltip is visible over the package name, displaying the error: "Package name 'myfirstpackage' does not correspond to the file path ". The tooltip also includes a "Move to package 'myfirstpackage'" button and a "More actions..." link.

tasto destro:
"Show context actions"

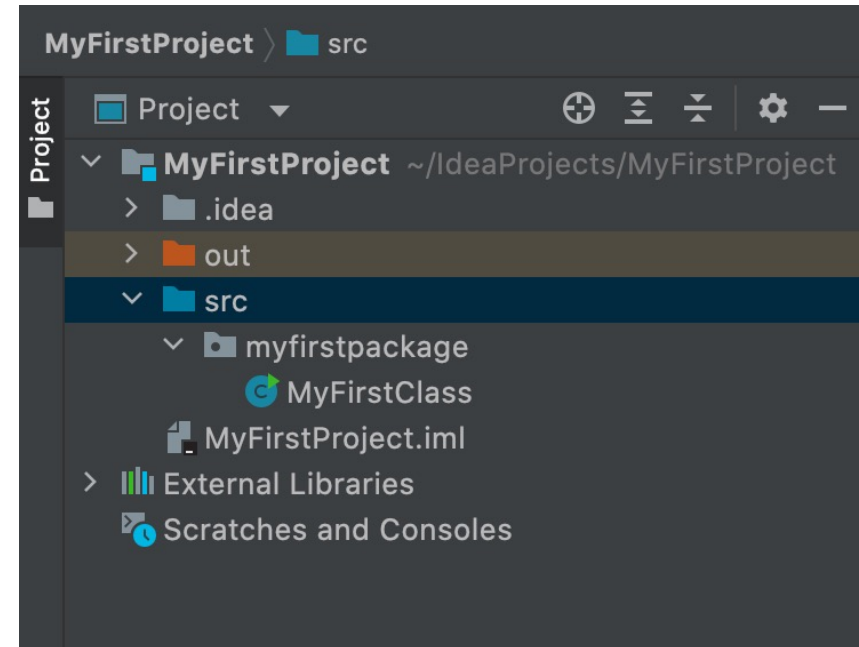
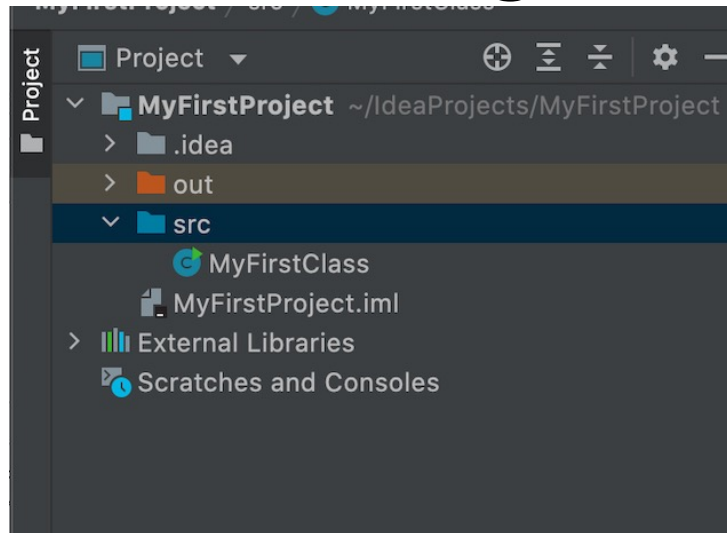


The screenshot shows the same IDE window. A right-click context menu is open over the package declaration on line 1. The menu contains the following items:

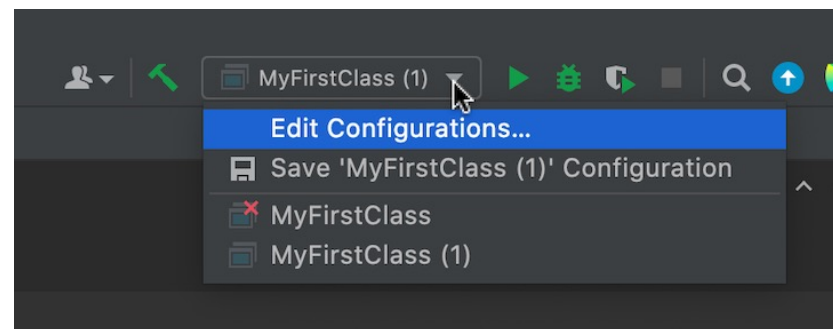
- Move to package 'myfirstpackage' >
- Press F1 to open preview

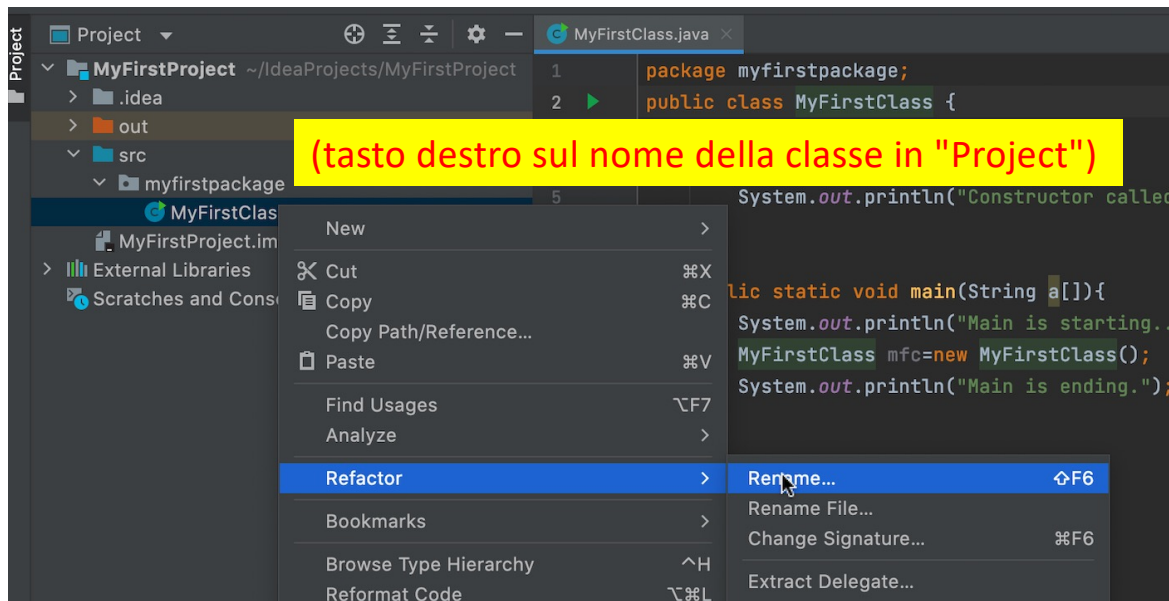
The "Move to package 'myfirstpackage' >" item is highlighted by the mouse cursor.

"refactoring"

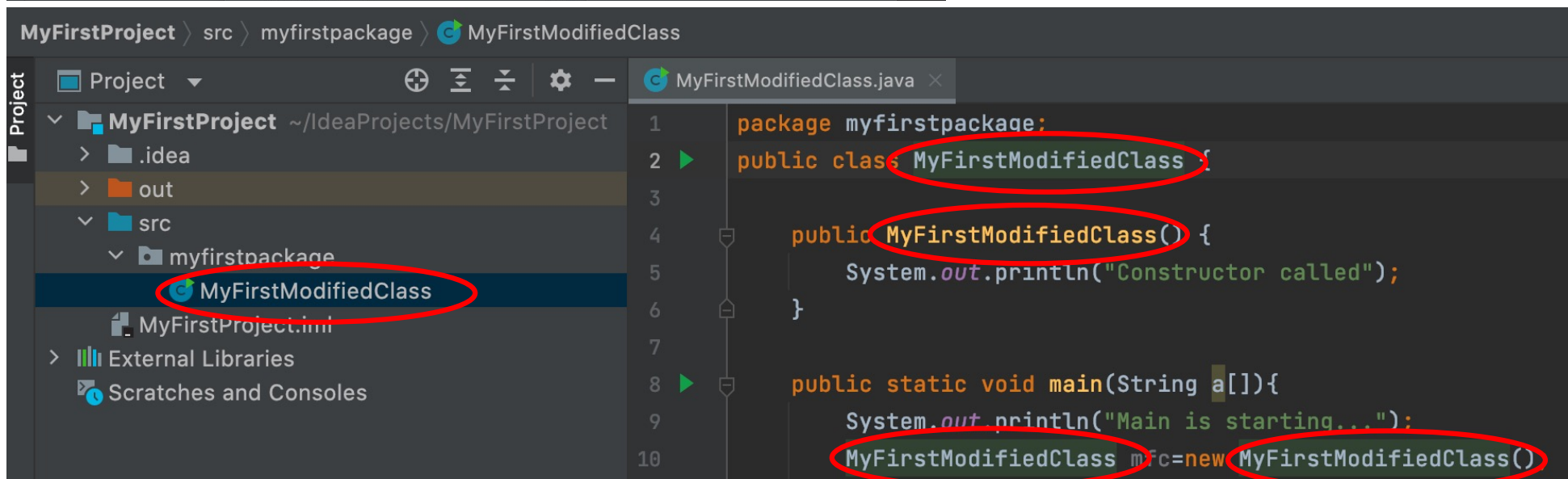


Eseguiamo nuovamente (come prima), e osserviamo:





refactoring

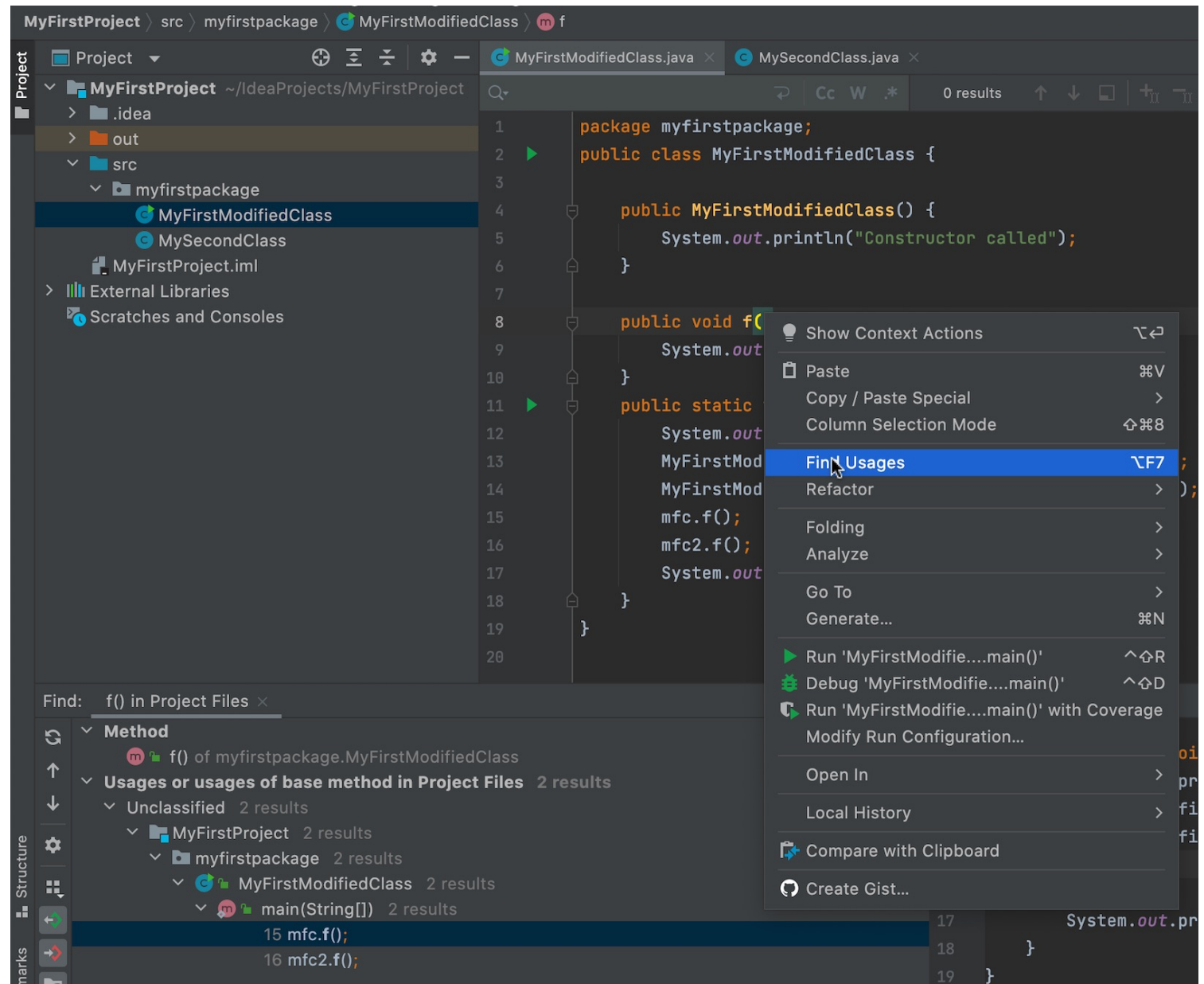


Let's add a second instance

```
public class MyFirstModifiedClass {  
  
    public MyFirstModifiedClass() {  
        System.out.println("Constructor called");  
    }  
  
    public void f(){  
        System.out.println("called f on " + this);  
    }  
  
    public static void main(String a[]){  
        System.out.println("Main is starting...");  
        MyFirstModifiedClass mfc=new MyFirstModifiedClass();  
        MyFirstModifiedClass mfc2=new MyFirstModifiedClass();  
        mfc.f();  
        mfc2.f();  
        System.out.println("Main is ending.");  
    }  
}
```

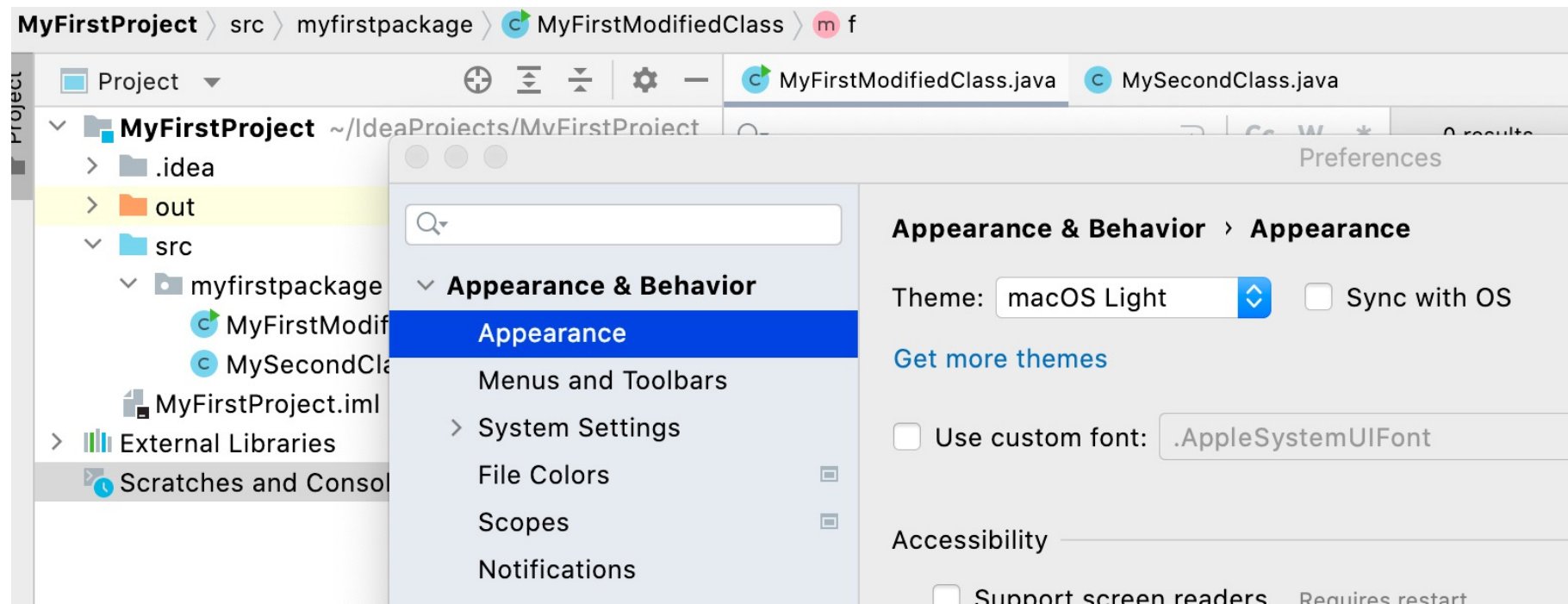
```
/Users/ronchet/Library/Java/JavaVirtualMachines/openjdk-1  
Main is starting...  
Constructor called  
Constructor called  
called f on myfirstpackage.MyFirstModifiedClass@36baf30c  
called f on myfirstpackage.MyFirstModifiedClass@7a81197d  
Main is ending.  
  
Process finished with exit code 0
```

find usages



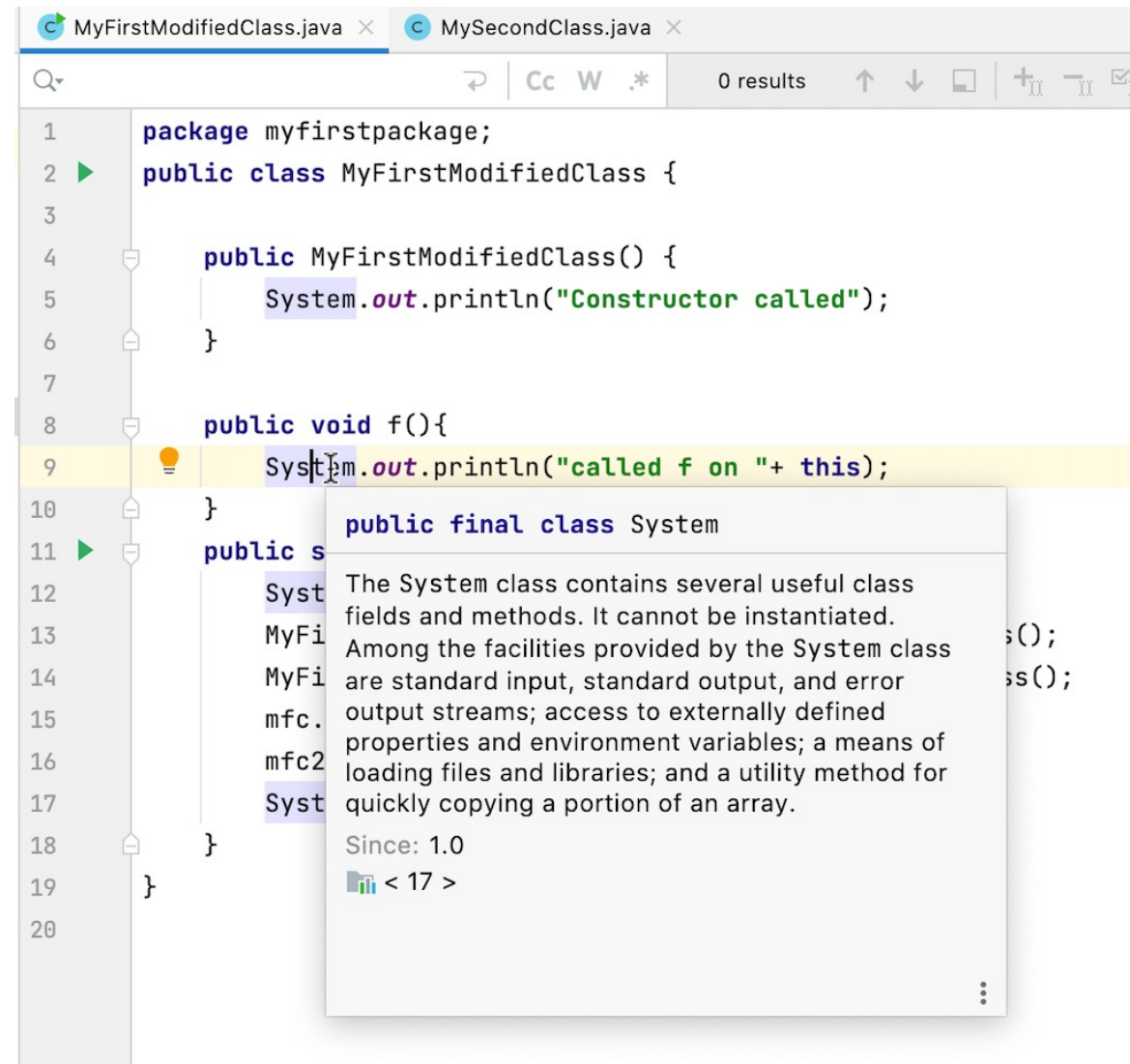
Change look

- go to Preferences



Check documentation

lascia il mouse sul codice per un uno o due secondi



The screenshot shows an IDE with two tabs: `MyFirstModifiedClass.java` and `MySecondClass.java`. The `MyFirstModifiedClass.java` tab is active, displaying the following code:

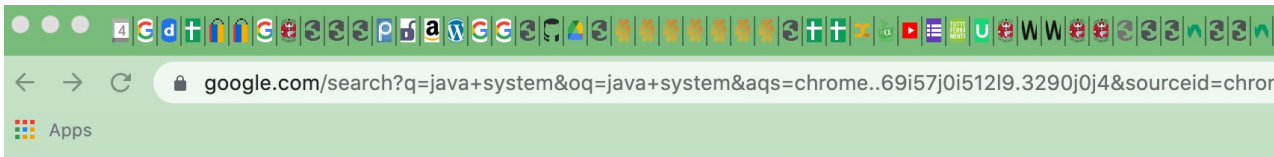
```
1 package myfirstpackage;
2 public class MyFirstModifiedClass {
3
4     public MyFirstModifiedClass() {
5         System.out.println("Constructor called");
6     }
7
8     public void f(){
9         System.out.println("called f on " + this);
10    }
11    public static void main(String[] args) {
12        MyFirstModifiedClass mfc = new MyFirstModifiedClass();
13        mfc.f();
14        MyFirstModifiedClass mfc2 = new MyFirstModifiedClass();
15        mfc2.f();
16        System.out.println("End of main");
17    }
18 }
19
20
```

A tooltip is displayed over the `System` class reference on line 9. The tooltip contains the following information:

- public final class System**
- The System class contains several useful class fields and methods. It cannot be instantiated.
- Among the facilities provided by the System class are standard input, standard output, and error output streams; access to externally defined properties and environment variables; a means of loading files and libraries; and a utility method for quickly copying a portion of an array.
- Since: 1.0
- < 17 >

oppure cerca su un browser

Check documentation



java system

Tutti

Immagini

Video

Shopping

Notizie

Altro

Strumenti

Circa 2.740.000.000 risultati (0,50 secondi)

Suggerimento: Cerca risultati solo in italiano. Puoi specificare la lingua di ricerca in Preferenze.

<https://docs.oracle.com/en/java/javase/7/docs/api/java.lang/System.html> Traduci questa pagina

System (Java Platform SE 7) - Oracle Help Center

Among the facilities provided by the **System** class are standard input, standard output, and error output streams; access to externally defined properties and ...

<https://docs.oracle.com/en/java/javase/12/docs/api/java.lang/System.html> Traduci questa pagina

System (Java SE 12 & JDK 12) - Oracle Help Center

It cannot be instantiated. Among the facilities provided by the **System** class are standard input, standard output, and error output streams; access to externally ...

O segna tra i tuoi bookmarks:

<https://docs.oracle.com/en/java/javase/17/docs/api/java.base/module-summary.html>



Module java.base

Package java.lang

Class System

java.lang.Object
java.lang.System

```
public final class System
extends Object
```

The System class contains several useful class fields and methods. It cannot be instantiated. Among the facilities provided by the System class are standard input, standard output, and error output streams; access to externally defined properties and environment variables; a means of ...

Since:

1.0

Nested Class Summary

Esercizio 1 :

- Aggiungere sia alla classe una variabile di istanza;
- modificare la funzione f in modo che incrementi la variabile di istanza e la stampi;

Esercizio 2 :

- Aggiungere una seconda classe che abbia un metodo g che stampa qualcosa;
- Creare nel main due istanze sia della prima che della seconda classe, e richiamare su di esse i relativi metodi.

Domanda : posso creare un metodo f anche nella seconda classe (ad esempio che stampi "Ciao da f"?)

Esercizio 3 :

- Creare un nuovo progetto, che stampi I numeri da 1 a 10.