Create an example that demonstrates proper utilization of <u>a callback</u>.

A callback is a piece of executable code that is passed as an argument to other code, which is expected to call back (execute) the argument at some convenient time.

Our aim is to realize a class that will take the name of the method as a parameter to the constructor:

```
MyClass m = new MyClass(myMethod());
```

But Java does not provide this mechanism, and we can pass a pointer to the object as a parameter:

```
MyClass m = new MyClass(this);
```

It remains to make the object **m** execute the required method of the object. We have to write an interface and implement a base class:

```
// The interface for handling the event:
public interface IEventListener {
   public void onEvent();
 }
// and main class:
public class MainClass implements IEventListener {
   // implement the method:
   public void onEvent() {
    //...
    System.out.print("--- Hello from callback-function! ---");
    //...
   }
 }
```

Implementing Class:

```
public class MyClass {
   protected IEventListener sender;    // reference to the object of the main class here

   public MyClass(IEventListener sender) {
     // save link
     this.sender = sender;
   }

   public void doWork() {
     //...
     sender.onEvent();
   }
 }
```

At the end we must create the object MyClass in the main class, passing in a reference to itself. We modify the realization of the main class:

```
public class MainClass implements IEventListener {
  MyClass myObject; // our object in which the event will occur

  // class constructor
  public MainClass() {
    myObject = new MyClass(this);
  }

  public void onEvent() {
   //...
   System.out.print("--- Hello from callback-function! ---");
   //...
  }
 }
```

So we create an example of work of a callback that demonstrates its proper utilization.