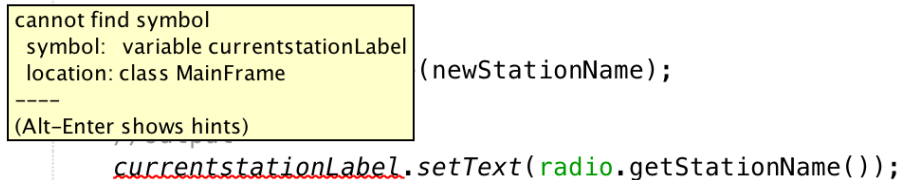


Erreurs fréquentes

Noms d'attributs, paramètres, variables

Message d'erreur :



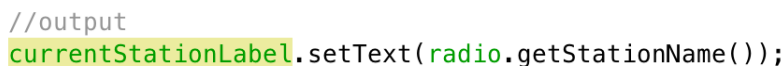
```
cannot find symbol
  symbol:   variable currentstationLabel
  location: class MainFrame
  ----
  (Alt-Enter shows hints)
currentstationLabel.setText(...);
```

Problème :

Il n'y a pas de variable/attribut nommé(e) `currentstationLabel`.

Que faire ?

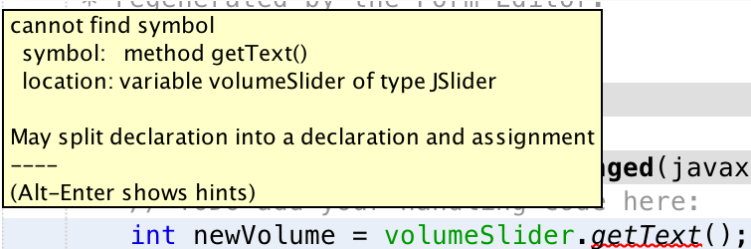
Vérifier si les noms de variables sont écrits correctement. Vérifier surtout si les majuscules et minuscules sont aux bons endroits.



```
//output
currentStationLabel.setText(...);
```

Noms de méthodes

Message d'erreur :



```
cannot find symbol
  symbol:   method getText()
  location: variable volumeSlider of type JSlider
  ----
  May split declaration into a declaration and assignment
  (Alt-Enter shows hints)
int newVolume = volumeSlider.getText();
```

Problème :

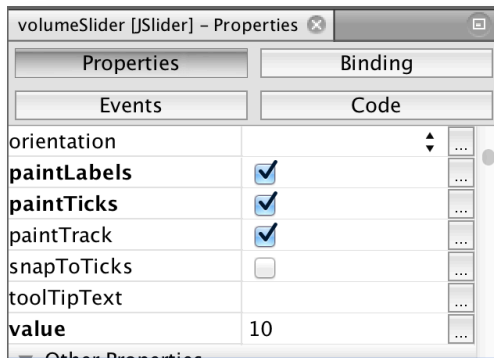
Le nom de la méthode est faux. Soit il est mal écrit, soit la méthode s'appelle autrement.

Que faire ?

1. Vérifier si le nom de la méthode est correct. Pour les méthodes des glissières, boutons, champs de texte et autres éléments graphiques il suffit de regarder dans l'inspecteur des propriétés quel est le nom de la propriété qu'on veut accéder ou manipuler.

2. Vérifier si les noms des méthodes utilisées sont écrits correctement. Vérifier surtout si les majuscules et minuscules sont aux bons endroits.

Dans l'exemple d'avant, le nom de la méthode est faux. En regardant les propriétés du `volumeSlider`, on se rend compte que la propriété s'appelle **value** et que par conséquent l'accesseur est nommé `getValue`.



Version correcte :

```
int newVolume = volumeSlider.getValue();
```

Si on obtient message d'erreur **symbol not found** pour une méthode du modèle, on peut simplement regarder le code de la classe du modèle pour vérifier si le nom de la méthode est correct.

Parenthèses des méthodes

Message d'erreur :

```
cannot find symbol
symbol:   variable getStationName
location: variable radio of type Radio
-----
(Alt-Enter shows hints)

currentStationLabel.setText(radio.getStationName);
```

Problème :

On veut appeler la méthode `getStationName` or on obtient un message d'erreur que la **variable** du nom `getStationName` n'existe pas. Le problème est que lors d'un appel d'une méthode le nom de celle-ci est toujours suivi de parenthèses. Ici il n'y en a pas ce qui amène le compilateur à chercher une variable du nom indiqué. Evidemment celle-ci n'existe pas.

Que faire ?

Mettre des parenthèses derrière le nom de la méthode.

```
//output
currentStationLabel.setText(radio.getStationName());
```

Paramètres des méthodes

Message d'erreur :

```
//
constructor Radio in class Radio cannot be applied to given types;
required: int,String
found: int
reason: actual and formal argument lists differ in length

Field radio can be final
----
(Alt-Enter shows hints)
```

a

```
private Radio radio = new Radio(10);
```

Problème :

Le problème est qu'il y a soit trop soit pas assez de paramètres qui sont passés au constructeur. En lisant les différentes lignes du message on constate :

- **required: int, String**
Il faut passer un entier et une chaîne de caractères au constructeur
- **found: int**
On n'a passé qu'un entier.
- **reason: actual and formal argument lists differ in length**
Le nombre de paramètres requis et le nombre de paramètres passés sont différents, ce qui cause l'erreur.

Que faire ?

Passer un entier et une chaîne de caractères comme paramètre.

```
private Radio radio = new Radio(10, "RTL");
```

Ordre et type des paramètres

Message d'erreur :

```

incompatible types: String cannot be converted to int
Field radio can be final
----
(Alt-Enter shows hints)
swing.JFrame

private Radio radio = new Radio("RTL", 10);

```

Problème :

En lisant la partie **String cannot be converted to int** on peut déduire que le compilateur Java essaie de convertir une chaîne de caractères en entier. Ceci arrive quand on essaie de passer une chaîne de caractères à un endroit où le compilateur s'attend à un entier.

Que faire ?

Vérifier le type et l'ordre de tous les paramètres. Ici on constate que dans la définition du constructeur de la classe `Radio` ...

```

public Radio(int pVolume, String pStationName){
    volume = pVolume;
    stationName = pStationName;
}

```

... l'ordre des paramètres est différent. Il faut passer le volume en premier lieu.

```

private Radio radio = new Radio(10, "RTL");

```

Conversion explicite des types

Message d'erreur :

```

// TODO add your handling code here:
incompatible types: int cannot be converted to String.getValue();
----
(Alt-Enter shows hints)
volumeLabel.setText(radio.getVolume());

```

Problème :

Le problème est similaire à celui de l'exemple précédent. Le compilateur Java essaie de convertir un entier en chaîne de caractères ce qui ne fonctionne pas automatiquement. La

méthode `setText` accepte une chaîne de caractères en paramètre, tandis qu'on fournit un entier.

Que faire ?

Il faut convertir l'entier qu'on désire afficher en chaîne de caractères. Il y a deux méthodes pour le faire.

1ère méthode :

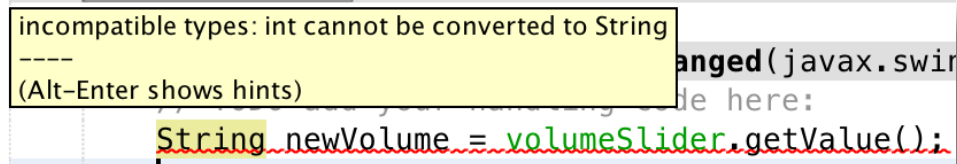
```
volumeLabel.setText(audio.getVolume() + "");
```

2ème méthode :

```
volumeLabel.setText(String.valueOf(audio.getVolume()));
```

Type de variable et type de retour

Message d'erreur :



```
incompatible types: int cannot be converted to String
----
(Alt-Enter shows hints)
String newVolume = volumeSlider.getValue();
```

Problème :

A nouveau on a un problème de conversion de types. Ici le compilateur Java essaie de convertir un entier en chaîne de caractères ce qui cause des problèmes.

Que faire ?

Il faut considérer le type de retour des méthodes. Ici le type de retour de la méthode `getValue` est un entier. Par conséquent le type de la variable `newVolume` doit être un entier (`int`).

```
int newVolume = volumeSlider.getValue();
```

NullPointerException

Message d'erreur dans la console (après le démarrage du programme) :

```
Exception in thread "AWT-EventQueue-0" java.lang.NullPointerException  
at MainFrame.volumeSliderStateChanged(MainFrame.java:84)
```

Problème :

L'attribut permettant la communication de la vue avec le modèle n'a pas été initialisé avec un objet.

Rappel : La valeur d'un attribut non primitif est `null` tant qu'une initialisation n'a pas eu lieu.

```
private Radio radio;
```

Que faire ?

Invoquer le constructeur de la classe `Radio` pour initialiser l'attribut `radio` avec un nouvel objet.

```
private Radio radio = new Radio(10, "RTL");
```