Exercice 1 – Syntaxe des objets

Q 1.1 Classe `Complexe`
La classe `Complexe` possède :
- deux attributs double `reelle` et `imag`,
- un constructeur à 2 arguments initialisant les deux attributs
  NB : signature obligatoire : `public Complexe(double reelle, double imag)`,
- un constructeur sans argument, qui initialise les arguments aléatoirement entre -2 et 2.
  NB : pas besoin de réfléchir à l'inclusion ou pas des extrêmes (faire simple)
  NB2 : utiliser obligatoirement la commande `this()` dans ce second constructeur.

Q 1.1.1 Donner le code de la classe `Complexe`

Q 1.1.2 Ajouter les méthodes suivantes (à vous de déterminer les signatures) :
- `toString` qui génère une chaine de caractère de la forme : (*reelle* `+` *imag* `i`)
- `addition` de deux complexes,
- `multiplication` de deux complexes,
- `estReel` qui teste si le complexe est en fait réel (dans le cas où la partie imaginaire est nulle).

Q 1.1.3 Donner le code de la classe `TestComplexe` qui, dans un `main`, effectue les opérations suivantes :
- créer 3 complexes, les afficher,
- tester s'ils sont réels ou pas,
- les additionner, multiplier et afficher les résultats
  NB : vous ajouterez en commentaire les résultats attendus

Q 1.1.4 Donner les instructions pour compiler ces deux classes et exécuter le test.

Q 1.2 Segment Complexe
Donner le code de la classe `SegmentComplexe` qui contient 2 complexes en attributs. Ajouter un constructeur à deux arguments et des accesseurs.

Exercice 2 – Code mystère + question

On imagine que toutes les fonctions précédentes sont opérationnelles et qu'une méthode `toString` a été redéfinie dans `SegmentComplexe` qui rend une chaine contenant la description des deux points composant le segment. On fait également l'hypothèse que la méthode suivante est ajoutée dans la classe `Complexe` :

```
1 // Dans la classe Complexe
2 public void translateDeUn(){ reelle+= 1; imag += 1; }
```

Dans une classe `Test`, à l'intérieur d'un `main`, on tape le code suivant :

```
1 Complexe c1 = new Complexe(1,0);
2 Complexe c2 = new Complexe(2,0);
3 Complexe c3;
4 SegmentComplexe s1 =
5     new SegmentComplexe(c1, c2);
6
7 c3 = c1.addition(c2);
8 System.out.println(c3);
9 if(c3.estReel())
10   System.out.println("c3 est reel");
11
12 System.out.println(s1);
13 c1.translateDeUn();
14 System.out.println(s1);
15 c1 = c3;
16 System.out.println(s1);
17
18 Complexe c4;
19 if(c4.estReel())
20   System.out.println("c4 est reel");
```

Q 2.1 Ce code contient une erreur (qui empêche la compilation et l'exécution) : trouver là (après la ligne 10).

Q 2.2 Donner les affichages produits lors de l'exécution (en imaginant que le problème précédent a été réglé).