

```
#ifndef Utilisateur-h
#define Utilisateur-h
```

\* 4  
An ann 1887 → Quelle  
relation : la classe abstraite ?  
association

```
class Utilisateur;
class Utilisateur Utilisateur {
```

```
Public:
    void ConfigurerGenerateur();
    void ObserverSignal();
```

```
Public: GenerateurSignal *myGenerateurSignal;
        Utilisateur *myUtilisateur;
```

```
};
```

```
#endif // Utilisateur-h
```

pourquoi appeler sa propre  
class ? Est-ce que cela du  
sens ?  
2

```
#ifndef Signal-h
#define Signal-h
```

```
class GenerateurSignal Signal;
class Signal {
```

```
Public:
```

```
Virtual void CalculerEchantillons(int Amplitude, int DutyCycle);
void AfficherEchantillons();
```

```
Private:
```

```
int GetPeriodeEch();
```

```
Public:
    int m_TableEch[20];
    int m_periodeEchantillonnage;
```

```
Public:
```

```
GenerateurSignal *myGenerateurSignal;
```

```
};
```

```
#endif // signal-h
```

5

si vous regardez le code n'y a-t-il pas quelque à placer

quels sont les liens entre  
la class Signal et les autres classes ?

```
#include Triangle-h
#define Triangle-h
#include "Signal.h"
```

```
class Triangle:
    Public Signal {
```

```
Public:
    void CalculerEchantillons(int Amplitude, int DutyCycle);
```

```
};
```

```
#endif // Triangle-h
```

```
#include GenerateurSignal-h
#define GenerateurSignal-h
#include Signal-h
```

```
class Utilisateur GenerateurSignal;
```

```
class GenerateurSignal {
```

```
Public:
```

```
int ChoisirSignal();
```

```
void ChoisirFrequence();
```

```
void ChoisirAmplitude();
```

const  
des

```
Private:
```

```
int m_Frequence;
```

```
int m_Amplitude;
```

```
Private:
Public:
```

```
Utilisateur *myUtilisateur;
GenerateurSignal *myGenerateurSignal;
```

```
#endif // GenerateurSignal-h
```

Donner que vous  
regardez la classe ?

dynamique

3