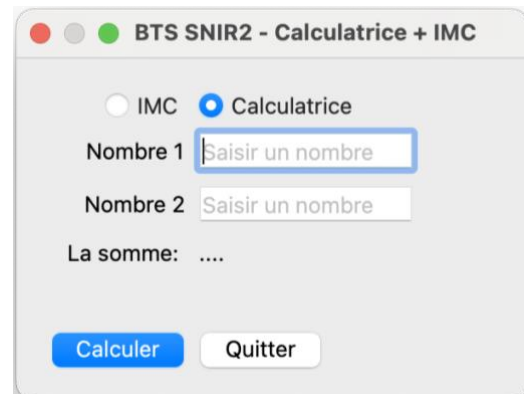
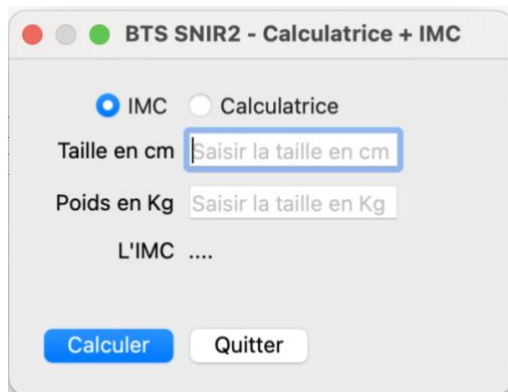


EVALUATION SYNTHESE 1
La programmation en Python
Création d'une Application de Calculatrice et IMC en PySide6

BTS SN-IR
2 <sup>ème</sup> année
Page 1 sur 3

## Exercice 1 : Création d'une Application de Calculatrice IMC en PySide6

### Interface graphique :



### Cahier des charges :

#### Radio Buttons (Boutons radio) :

- Deux boutons radio permettent à l'utilisateur de choisir entre le mode IMC et le mode Calculatrice.
- Lorsque le bouton "IMC" est sélectionné, l'interface est configurée pour calculer l'Indice de Masse Corporelle (IMC).
- Lorsque le bouton "Calculatrice" est sélectionné, l'interface est configurée pour effectuer des calculs de somme entre deux nombres.

#### Champs de Saisie (LineEdit) :

- Les deux champs de saisie permettent à l'utilisateur d'entrer les valeurs nécessaires, comme la taille et le poids pour le calcul de l'IMC ou les nombres pour la calculatrice.

<b>EVALUATION SYNTHESE 1</b>
<b>La programmation en Python</b>
Création d'une Application de Calculatrice et IMC en PySide6

<b>BTS SN-IR</b>
<b>2<sup>ème</sup> année</b>
Page 2 sur 3

**Bouton "Calculer" :**

- En cliquant sur ce bouton, l'application effectue le calcul en fonction du mode sélectionné (IMC ou Calculatrice) et affiche le résultat dans le Label correspondant.

**Label de Résultat :**

- Ce Label affiche le résultat du calcul ou un message d'erreur en cas de saisie incorrecte.

**Bouton "Quitter" :**

- Ce bouton permet de quitter l'application.

**Fonctions de Conversion :**

- Les fonctions calculerIMC et calculerSomme sont définies pour configurer l'interface en fonction du mode sélectionné.

**Fonction choisir :**

- Cette fonction est appelée lorsque le bouton "Calculer" est cliqué. Elle effectue le calcul en fonction du mode sélectionné.

**Fonction effacer :**

- Cette fonction efface les champs de saisie et réinitialise l'étiquette de résultat.

**Messages d'Erreur :**

- Trois messages d'erreur différents sont définis pour informer l'utilisateur en cas de saisie incorrecte lors de la conversion en nombres.

Code pour créer et afficher un MessageBox :

```
self.messageChampsVides = QtWidgets.QMessageBox()
self.messageChampsVides.setText("Erreur de saisie\nVeuillez resaisir des valeurs correctes")
self.messageChampsVides.show()
```

**Exercice 2 : Création d'une Classe pour une Station Météo avec Capteurs**

A partir du diagramme de classe :

<b>WeatherStation</b>
- _temperature: float - _pression: float - _humidite: int
+ convert_temperature(): float + convert_pressure(): float + display_info(): void + getattr(attr: str): any + str(): any

- Créez une classe `WeatherStation` pour représenter une station météo avec trois capteurs : température, pression, et humidité.
- Ajoutez trois attributs privés : `_temperature`, `_pression`, et `_humidite`.
- Ajoutez les accesseurs et mutateurs pour ces attributs privés.
- Définissez une propriété pour chaque attribut privé.
- Ajoutez trois méthodes pour convertir la température de Fahrenheit à Celsius, la pression de Pascal à Bar, et pour afficher toutes les informations de la station météo.
- Ajoutez une méthode spéciale `__doc__` qui renvoie la documentation de la classe.
- Ajoutez une méthode `getattr` qui renvoie la valeur de l'attribut demandé ou une chaîne indiquant que l'attribut n'existe pas.
- Ajoutez une méthode `str` qui renvoie la valeur d'un attribut lorsqu'on essaye d'afficher l'objets

```
def convert_temperature(self):
    return (self._temperature - 32) * 5/9
def convert_pressure(self):
    return self._pression / 100000
```