Pratique de développement piloté par les tests

Remarques avant de commencer :

- Considérez chaque item des requis fonctionnels comme une itération (incrément). Dans chaque itération, écrivez vos tests et ensuite le code pour que les tests passent.
- Faites un commit de votre code après chaque itération (chaque fois qu'un test passe).
- Rappelez-vous des règles d'or du code de qualité :
    - Écrire avec un bon style cohérent. Utilisez des noms clairs et expressifs.
    - Écrire de petites fonctions et classes n'ayant qu'une seule responsabilité.
    - Ne répétez pas de code.
    - Trouvez les solutions les plus simples possibles
    - Etc.
- Rappelez-vous de remanier votre code (si nécessaire) après chaque test.

Travail à réaliser

1. Créer une simple calculatrice avec une méthode *int add(numberString : string)*
    a. Pour un *string* vide, la méthode retourne 0.
        i. `add("")` retourne 0.
    b. Pour un seul nombre, la méthode retourne la valeur du nombre
        i. `add("2")` retourne 2, `add("14")` retourne 14, etc.
2. Permettre à la méthode `add()` de prendre une chaîne ayant jusqu'à 2 nombres (séparés par ',') et retourner leur somme.
    a. `add("2,3")` retourne 5, `add("12,5")` retourne 17, etc.
3. Permettre à la méthode de traiter un nombre arbitraire de valeurs.
    a. `add("2,3,5")` retourne 10, `add("11,10,2,3,5,1,3")` retourne 35, etc.
4. Permettre à la méthode d'utiliser des sauts de ligne au lieu de virgules entre les chiffres.
    a. La chaîne suivante est valide : "1\n2,3" (la méthode retourne 6)
    b. Celle-ci est invalide : "1,\n"
5. Permettre de supporter différents séparateurs:
    a. Pour changer le séparateur, le string doit commencer par une ligne ayant le format suivant : //[séparateur]
    b. Par exemple : `add("//;\n1;2")` devrait retourner 3 en utilisant le séparateur ';'.
    c. La première ligne est optionnelle, les scenarios précédents doivent encore être supportés.