

Projekt z pkt. 1 lit. b polegał na utworzeniu klasy `StringCalculator`, prostego kalkulatora o jednej metodzie `add`, przyjmującej argument typu `String`, zawierającego listę liczb i zwracającej ich sumę. Kolejno były przedstawiane kolejne wymagania, jakie ta klasa ma spełnić.

Po zapoznaniu się z każdym wymaganiem, pisano test, badający, czy jest ono spełnione, a następnie tworzone kod, który przeszedłby ten test (i wszystkie poprzednie). Jedno z wymagań wymagało zakomentowania jednego z poprzednich testów, ponieważ znosiło wymaganie przez niego testowane.

Lista wymagań:

0. Klasa ma mieć metodę `add (string numbers)`
1. Metoda ta może przyjmować 0, 1 lub 2 liczby rozdzielone przecinkami (a) i ma zwracać ich sumę (c) (dla pustego łańcucha 0 (b)), np. „”, „1” lub „1,2”.
2. Pozwolić metodzie `add` na obsługiwanie nieznanej liczby liczb.
3. Pozwolić metodzie `add` na obsługiwanie nowych linii między liczbami (zamiast przecinków).
4. Wprowadzić obsługę innych separatorów. W celu zmiany separatora początek łańcucha musi zawierać dodatkową linię o formacie `//[separator]` (całość wygląda tak: `//[separator]\n[liczby]`), np. `//;\n1;2` powinno zwrócić 3. Pierwsza linia jest opcjonalna, wszystkie dotychczasowe scenariusze powinny wciąż działać.
5. Wywołanie metody `add` z liczbami ujemnymi ma zwrócić wyjątek, mówiący, że nie są one dozwolone oraz listę podanych liczb ujemnych.
6. Liczby większe od 1000 powinny zostać zignorowane ($2 + 1001 = 2$).
7. Separatory mogą mieć dowolną długość (format taki sam, `//[separator]\n`), np. `//[—]\n1—2—3` powinno zwrócić 6.
8. Dopuszczalne wiele separatorów: `//[sep1][sep2]\n`, np. `//[-][%]\n1-2%3` powinno zwrócić 6.
9. Upewnij się, że możliwa jest obsługa wielu separatorów dłuższych niż 1 znak.

Nazwy metod testowych, podobnie jak w blogu, mają postać `when [akcja] then [weryfikacja]`.

Ad 1.

(a)

Funkcja `whenMoreThan2NumbersAreUsedThenExceptionIsThrown` oczekuje, że metoda `add` zgłosi wyjątek, jeśli zostaną podane więcej niż 2 liczby. Test `when2NumbersAreUsedThenNoExceptionIsThrown` sprawdza, czy metoda `add` nie zgłosi żadnego wyjątku po podaniu 2 liczb, natomiast `whenNonNumberIsUsedThenExceptionIsThrown`, czy metoda zwróci wyjątek gdy napotka `NaN`.

(b)

Test `whenZeroNumbersAreUsedThenReturnValueIs0` sprawdza, czy wywołanie metody z pustym łańcuchem zwróci wartość 0.

(c)

Test `whenOneNumberIsUsedThenReturnValueIsThatSameNumber` sprawdza, czy jeśli podano jedną liczbę, zwrócona wartość to ta sama liczba, a `whenTwoNumbersAreUsedThenReturnValueIsTheirSum` — czy wynik jest sumą dwóch podanych liczb.

Ad 2.

Test `whenAnyNumberOfNumbersIsUsedThenReturnValueIsTheirSum` sprawdza, czy dla dowolnie długiej listy liczb wynik to ich suma.

Ad 3.

`whenNewlineIsUsedBetweenNumbersThenReturnValueIsTheirSum` sprawdza, czy nowa linia traktowana jest jako separator (zamiennie z przecinkiem).

Ad 4.

`whenDelimiterIsSpecifiedThenItIsUsedToSeparateNumbers` sprawdza, czy podanie separatora zgodnie z podanym formatem sprawia, że jest on rozpoznawany przez funkcję i stosowany do rozdzielania liczb.

Ad 5.

`whenNegativeNumberIsUsedThenRuntimeExceptionIsThrown` sprawdza, czy pojawienie się liczby ujemnej na liście liczb spowoduje zgłoszenie wyjątku, natomiast

`whenNegativeNumbersAreUsedThenRuntimeExceptionIsThrown` — czy odpowiedni jest komunikat tego wyjątku („Negatives not allowed:” + lista użytych liczb ujemnych).

Ad 6.

`whenOneOrMoreNumbersGreaterThan1000IsUsedThenItIsNotIncludedInSum` sprawdza, czy użyte liczby większe od 1000 są ignorowane przy sumowaniu. Sprawdzony jest warunek brzegowy: na liście pojawiają się m.in. liczby 1000 i 1001: pierwsza jest uwzględniana, druga nie — zgodnie z wymaganiami.

Ad 7.

`whenDelimiterOfAnyLengthIsSpecifiedThenItIsUsedToSeparateNumbers` sprawdza, czy możliwe jest ustawienie separatora o dowolnej długości, zgodnie z podanym formatem.

Ad 8.

`whenSeveralDelimitersAreSpecifiedThenAllAreUsedToSeparateNumbers` sprawdza, czy jeśli zgodnie z podanym formatem podano kilka separatorów, wszystkie mogą być użyte do rozdzielania liczb (ten test sprawdza separatory jednoznakowe)

Ad 9.

`whenSeveralDelimitersOfAnyLengthAreSpecifiedThenAllAreUsedToSeparateNumbers` sprawdza, czy można podać kilka separatorów wieloznakowych (jeden z separatorów ma długość jednego znaku, pozostałe są dłuższe, by upewnić się, że jednoznakowe wciąż są obsługiwane, jeśli niektóre mają większą długość). Kod utworzony na potrzeby poprzedniego testu nie wymagał zmiany, by przejść ten test (od razu obsługiwał dłuższe separatory).