#1.

Do wykonania jest biblioteka matematyczna, pozwalająca na pisanie działań w ciągu: `math.Add(1).Sub(1).Div(1);`

Operacje wykonywane mają być na wartościach całkowitych (dla uproszczenia).

Każda z metod zwraca wynik operacji na końcu bieżącego działania. Sugeruje się wykorzystanie wzorca **Fluent lub Extension Method.**

Na początek:

```
git init
git status
git flow config
git flow release start IDENTYFIKATOR_WYDANIA
git flow release track IDENTYFIKATOR_WYDANIA
git flow release publish IDENTYFIKATOR_WYDANIA
git flow feature start IDENTYFIKATOR
git flow feature finish IDENTYFIKATOR
git flow feature publish IDENTYFIKATOR
git flow feature pull IDENTYFIKATOR
git push --tags
```

1.1. Ustawienie wartości bazowej.

Jako użytkownik biblioteki Matematycznej, chcę mieć możliwość ustawienia wartości początkowej, ponieważ chcę mieć kontrolę nad podstawą wyliczenia.

Zadania:

1) Stworzenie **feature** – ustawiania.
2) Wypisanie testów.
3) Implementacja testów.
4) Implementacja **feature**.
5) Refakoryzacja.
6) Testowanie.
7) Integracja z wydaniem.

Pomoc:

Testy: Fact, Theory, InLineData

Asercje: ShouldBe()

1.1. Ustawienie wartości bazowej.

return_0__when__set_initial_data_to_0_and_expression_contains_only_init_data

default_set_to_0__when__init_without_initial_data

1.2. Dodawanie liczb.

Jako użytkownik biblioteki Matematycznej chcę mieć możliwość użycia w wyrażeniu operacji dodawania, ponieważ chcę dodawać wartości.

Zadania:

1) Stworzenie **feature** – dodawania.
2) Wypisanie testów.
3) Implementacja testów.
4) Implementacja **feature**.
5) Refaktoryzacja.
6) Testowanie.
7) Integracja z wydaniem.

Pomoc:

```
git flow feature start
git flow feature publish
git flow feature  pull
git flow feature  finish
```

1.2. Dodawanie liczb.

increase_new_value__when__add_value_to_earlier_value

throw_exception__when__overflow_max_value_for_type
throw_exception__when__overflow_min_value_for_type

1.3. Odejmowanie liczb.

Jako użytkownik biblioteki Matematycznej chcę mieć możliwość użycia w wyrażeniu operacji odejmowania, ponieważ chcę odejmować wartości.

Zadania:

1) Stworzenie **feature** – odejmowania.
2) Wypisanie testów.
3) Implementacja testów.
4) Implementacja **feature**.
5) Refaktoryzacja.
6) Testowanie.
7) Integracja z wydaniem.

Pomoc:

git flow...

git log

1.3. Odejmowanie liczb.

result_must_be_correct__when__subtraction_earlier_value_with_new_value

throw_exception__when__overflow_max_value_for_type
throw_exception__when__overflow_min_value_for_type

1.4. Mnożenie liczb.

Jako użytkownik biblioteki Matematycznej chcę mieć możliwość użycia w wyrażeniu operacji mnożenia, ponieważ będę mnożył wartości.

Zadania:

1) Stworzenie **feature** – mnożenia.
2) Wypisanie testów.
3) Implementacja testów.
4) Implementacja **feature**.
5) Refaktoryzacja.
6) Testowanie.
7) Integracja z wydaniem.

Pomoc:

1.4. Mnożenie liczb.

value_must_be_correct__when__multiplication_earlier_value_with_new

1.5. Dzielenie liczb.

Jako użytkownik biblioteki Matematycznej chcę mieć możliwość użycia operacji dzielenia, ponieważ chcę skorzystać z dzielenia wartości.

Zadania:

1) Stworzenie **feature** – dzielenia.
2) Wypisanie testów.
3) Implementacja testów.
4) Implementacja **feature**.
5) Refaktoryzacja.
6) Testowanie.
7) Integracja z wydaniem.

Pomoc:

1.5. Dzielenie liczb.

value_must_be_correct__when__division_value_with_new_value

division_are_round__when__division_value_are_greate_than_the_division_multiplication

throw_exception__when__division_by_zero

1.6. Potęgowanie liczb.

Jako użytkownik biblioteki Matematycznej chcę mieć możliwość użycia operacji potęgowania, ponieważ zamierzam podnosić wartości do potęgi.

Zadania:

1) Stworzenie **feature** – potęgowania.
2) Wypisanie testów.
3) Implementacja testów.
4) Implementacja feature.
5) Refaktoryzacja.
6) Testowanie.
7) Integracja z wydaniem.

Pomoc:

1.6. Potęgowanie liczb.

return_value_1__when__exponentiation_to_0

return_same_value__when__exponentiation_to_1

return_correct_value__when__exponentiation

1.7. Wyliczenie procentów.

Jako użytkownik biblioteki Matematycznej chcę mieć możliwość użycia wyliczenia procentowej wartości, ponieważ chcę zamierzam podnosić wartości do potęgi.

Zadania:

1) Stworzenie **feature** – wyliczenia procentów.
2) Wypisanie testów.
3) Implementacja testów.
4) Implementacja feature.
5) Refaktoryzacja.
6) Testowanie.
7) Integracja z wydaniem.

Pomoc:

1.7. Wyliczanie procentów.

return_same_value__when__calculate_100_percent
return_zero__when__calculate_0_percent

throw_exception__when__try_calculate_less_0_value
throw_exception__when__try_calculate_greater_than_100_percent

1.8. Zaokrąglanie klasyczne.

Jako użytkownik biblioteki Matematycznej chcę mieć możliwość użycia operacji zaokrąglania klasycznego, ponieważ potrzebuję pozbyć się miejsc po przecinku.

Zadania:

1) Stworzenie **feature** – zaokrąglania klasycznego.
2) Wypisanie testów.
3) Implementacja testów.
4) Implementacja feature.
5) Refaktoryzacja.
6) Testowanie.
7) Integracja z wydaniem.

Pomoc:

Math.Round(wartość)

1.8. Zaokrąglanie klasyczne.

rounds_to_up__when__value_after_the_decimal_point_is_greater_than_5

rounds_to_down__when__value_after_the_decimal_point_is_less_or_equal_to_5

1.9. Zaokrąglanie w górę.

Jako użytkownik biblioteki Matematycznej chcę mieć możliwość użycia operacji zaokrąglenia w górę, ponieważ chcę zaokrąglać wartości do wartości górnej.

Zadania:

1) Stworzenie **feature** – zaokrąglania w górę.
2) Wypisanie testów.
3) Implementacja testów.
4) Implementacja feature.
5) Refaktoryzacja.
6) Testowanie.
7) Integracja z wydaniem.

Pomoc:

Math.Ceiling(wartość)

1.9. Zaokrąglanie w górę.

rounds_to_up__wneh__any_value_after_the_decimal

1.10. Zaokrąglanie w dół.

Jako użytkownik biblioteki Matematycznej chcę mieć możliwość użycia operacji zaokrąglenia w dół, ponieważ chcę zaokrąglać wartości do wartości dolnej.

Zadania:

1) Stworzenie **feature** – zaokrąglania w dół.
2) Wypisanie testów.
3) Implementacja testów.
4) Implementacja feature.
5) Refaktoryzacja.
6) Testowanie.
7) Integracja z wydaniem.

Pomoc:

Math.Floor(wartość)

1.10. Zaokrąglanie w dół.

rounds_to_down__when__any_value_after_the_decimal