

과 제 4 (연산자 오버로딩)

2024.5.16

다음과 같은 클래스 Vector가 주어져 있다.

Vector
+v: (float)
__init__(*args)
__add__(Vector): Vector
__sub__(Vector): Vector
__mul__(Vector): float
+ distance(Vector): float
+ norm(): float
__str__(): str

다음의 표는 위의 클래스에 대한 속성(attributes)과 메소드(methods)에 대한 사항을 정리한 것이다.

속성/메소드	설 명 (description)
v	n-차원 벡터 원소 저장, 자료형은 원소 자료형이 float인 tuple
__init__()	생성자로서 매개변수는 가변인수를 처리하도록 정의 임의의 차원에 대한 인수를 모두 처리 가능해야 함.
__add__()	같은 차원의 벡터간의 + 연산에 대한 메소드 구체적인 연산은 다음의 실행 예를 참고
__sub__()	같은 차원의 벡터간의 - 연산에 대한 메소드 구체적인 연산은 다음의 실행 예를 참고
__mul__()	같은 차원의 벡터간의 * 연산(스칼라 곱)에 대한 메소드 구체적인 연산은 다음의 실행 예를 참고
distance()	매개변수로 받은 벡터와의 거리를 계산하여 반환
norm()	벡터의 길이를 계산하여 반환
__str__()	인스턴스 변수의 값을 문자열로 반환 구체적인 반환값 포맷은 실행 예를 참고

[참고] 벡터 연산 공식

임의의 $\mathbf{u} = (u_1, u_2, \dots, u_n)^T \in \mathbb{R}^n$, $\mathbf{v} = (v_1, v_2, \dots, v_n)^T \in \mathbb{R}^n$ 에 대해,

- 벡터의 덧셈: $\mathbf{u} + \mathbf{v} = (u_1 + v_1, u_2 + v_2, \dots, u_n + v_n)^T \in \mathbb{R}^n$
- 벡터의 뺄셈: $\mathbf{u} - \mathbf{v} = (u_1 - v_1, u_2 - v_2, \dots, u_n - v_n)^T \in \mathbb{R}^n$

- 벡터의 스칼라 곱: $\mathbf{u} * \mathbf{v} = u_1 v_1 + u_2 v_2 + \dots + u_n v_n$
 - 벡터간 거리 : $|\mathbf{u} - \mathbf{v}| = \sqrt{(u_1 - v_1)^2 + (u_2 - v_2)^2 + \dots + (u_n - v_n)^2}$
 - 벡터의 길이: $|\mathbf{u}| = \sqrt{u_1^2 + u_2^2 + \dots + u_n^2}$
- ☞ 벡터간의 연산은 두 벡터가 같은 크기(즉, 같은 차원)이어야 함.

실행 예:

```
>>> u1 = Vector(1, 2, 3, 4)
>>> u2 = Vector(-1, 0, 1, 0)
>>> u3 = Vector(1, 0, 1)
>>> u4 = Vector(-1, 0, 1)
>>> u1 + u2
(0, 2, 4, 4)

>>> u3 - u4
(2, 0, 0)

>>> u1 * u2
2

>>> u3.distance(u4)
2

>>> u1.norm()
5.4772

>>> u1 + u3
벡터의 크기가 서로 달라 + 연산이 불가능
```

다음 사항을 스크립트 모드로 프로그래밍하시오.

1. 주어진 Vector 클래스를 정의하시오.
2. 위의 실행 예에서의 4개의 벡터 객체를 생성하고, 다음의 벡터 연산 결과를 출력하는 프로그램을 작성하시오.
 - ♦ $u1 + u2$, $u1 + u3$
 - ♦ $u3 - u4$
 - ♦ $u1 * u2$, $u1 * u3$
 - ♦ $u1$ 과 $u2$ 간의 거리, $u3$ 와 $u4$ 간의 거리
 - ♦ $u1$, $u2$, $u3$, $u4$ 의 길이