الاسم: احمد وليد عوض عبدالمجيد سيكشن: 1

Chapter 2: Python Data Model and Internals (Page 36)

:Point Class with slots (Section 2.4)

- **:Purpose** Defining __slots__ replaces the instance's dynamic dictionary (__dict__) with a fixed-size array.
- **:Result of adding p.z = 5** It will raise an **AttributeError**.
- **:Why** Because __slots__ restricts the valid attributes to only those explicitly defined (e.g., 'x' and 'y'). This results in **reduced memory footprint** and **faster attribute access**.

:Disassembling Analysis (Section 1.3)

- LOAD_FAST: Loads a local variable onto the stack.
- BINARY_ADD: Pops the top two values, adds them, and pushes the result.
- RETURN_VALUE: Returns the top value of the stack.

Significance: These instructions demonstrate that the **Python Virtual Machine (PVM)** is a **stack-based interpreter**.

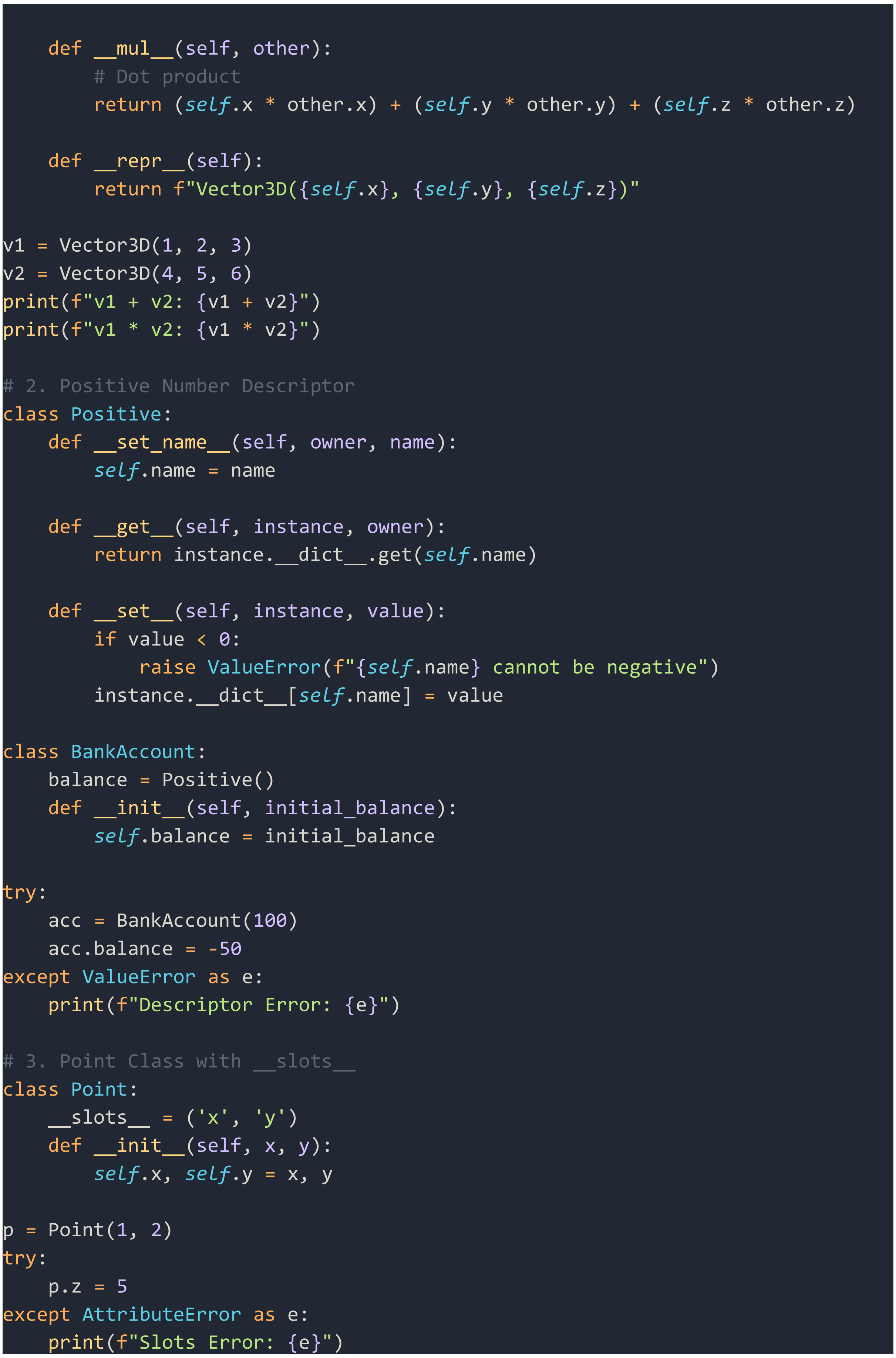
```python
# ==========================================
# Chapter 2: Python Data Model
# ==========================================
print("\n--- Chapter 2 Solutions ---")

# 1. Vector3D Class
class Vector3D:
    def __init__(self, x, y, z):
        self.x, self.y, self.z = x, y, z

    def __add__(self, other):
        return Vector3D(self.x + other.x, self.y + other.y, self.z + other.z)

    def __sub__(self, other):
        return Vector3D(self.x - other.x, self.y - other.y, self.z - other.z)
```

الاسم: احمد وليد عوض عبدالمجيد	سيكشن: 1



```python
    def __mul__(self, other):
        # Dot product
        return (self.x * other.x) + (self.y * other.y) + (self.z * other.z)

    def __repr__(self):
        return f"Vector3D({self.x}, {self.y}, {self.z})"

v1 = Vector3D(1, 2, 3)
v2 = Vector3D(4, 5, 6)
print(f"v1 + v2: {v1 + v2}")
print(f"v1 * v2: {v1 * v2}")

# 2. Positive Number Descriptor
class Positive:
    def __set_name__(self, owner, name):
        self.name = name

    def __get__(self, instance, owner):
        return instance.__dict__.get(self.name)

    def __set__(self, instance, value):
        if value < 0:
            raise ValueError(f"{self.name} cannot be negative")
        instance.__dict__[self.name] = value

class BankAccount:
    balance = Positive()
    def __init__(self, initial_balance):
        self.balance = initial_balance

try:
    acc = BankAccount(100)
    acc.balance = -50
except ValueError as e:
    print(f"Descriptor Error: {e}")

# 3. Point Class with __slots__
class Point:
    __slots__ = ('x', 'y')
    def __init__(self, x, y):
        self.x, self.y = x, y

p = Point(1, 2)
try:
    p.z = 5
except AttributeError as e:
    print(f"Slots Error: {e}")
```

الاسم: احمد وليد عوض عبدالمجيد

سيكشن: 1

```python
# 4. Disassembling calculate_sum
def calculate_sum(a, b):
    return a + b

print("\nDisassembly of calculate_sum:")
dis.dis(calculate_sum)
```