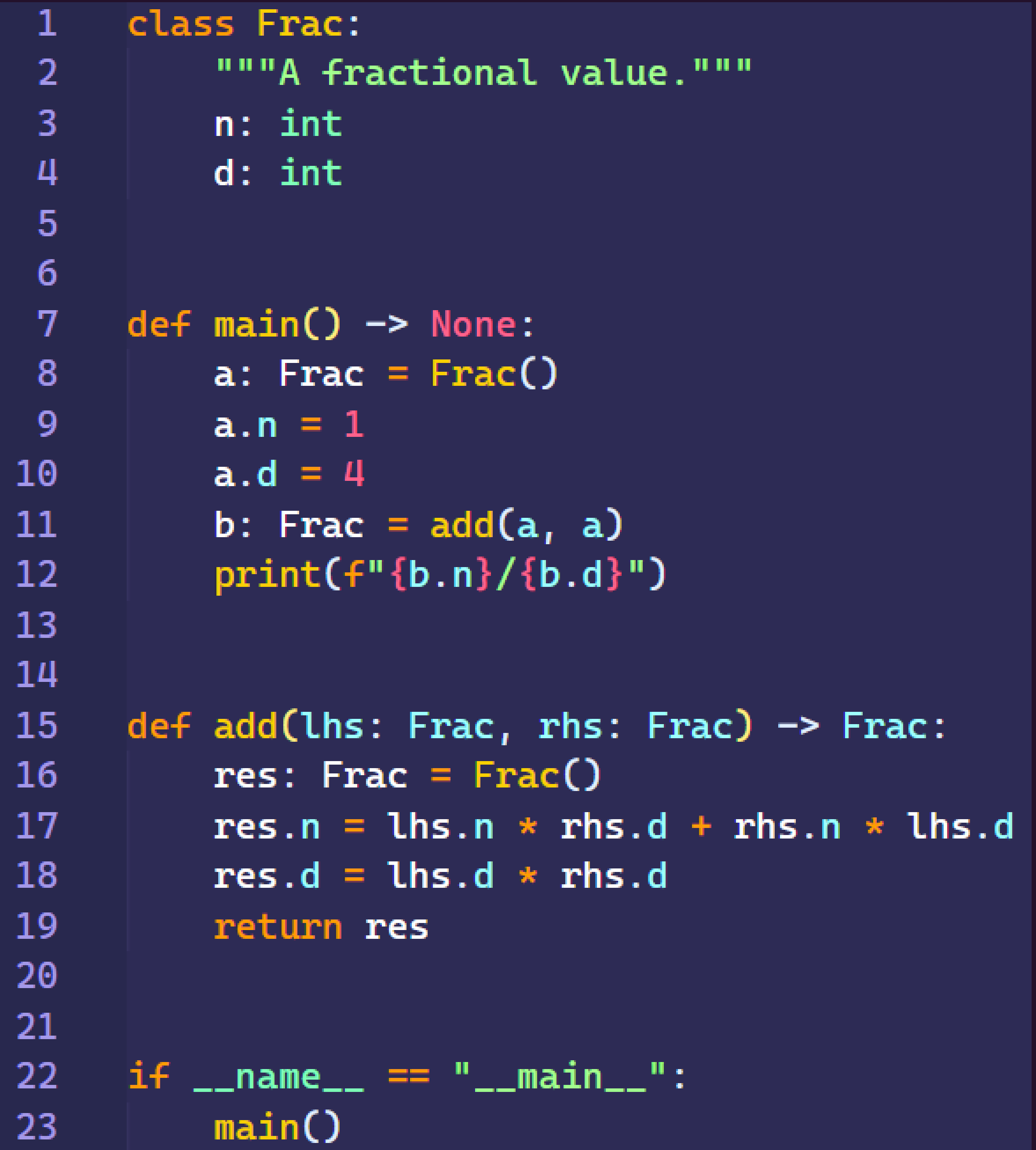
Classes + Objects in Diagrams!

Q0



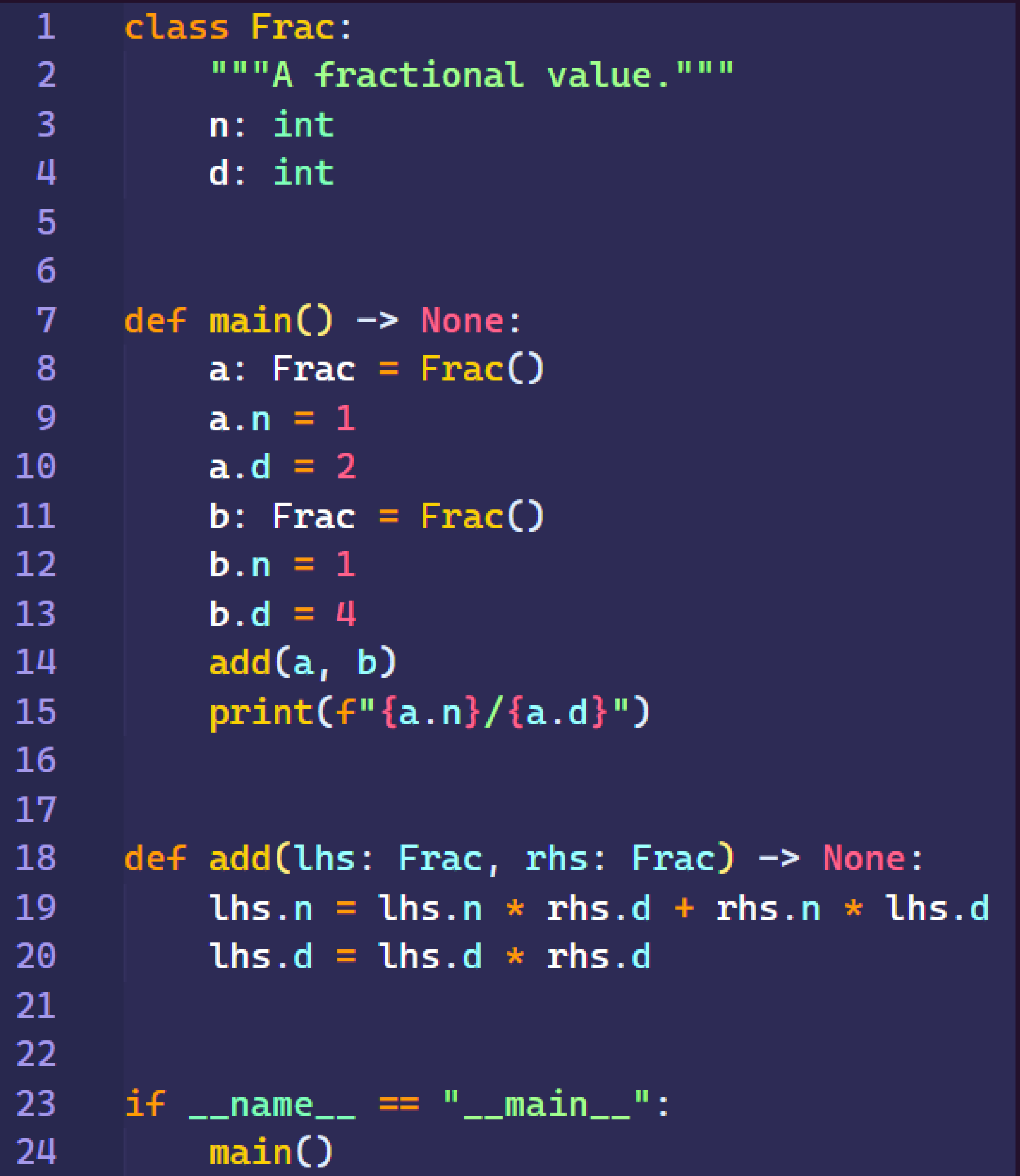
```python
class Frac:
    """A fractional value."""
    n: int
    d: int


def main() -> None:
    a: Frac = Frac()
    a.n = 1
    a.d = 4
    b: Frac = add(a, a)
    print(f"{b.n}/{b.d}")


def add(lhs: Frac, rhs: Frac) -> Frac:
    res: Frac = Frac()
    res.n = lhs.n * rhs.d + rhs.n * lhs.d
    res.d = lhs.d * rhs.d
    return res


if __name__ == "__main__":
    main()
```

Q1



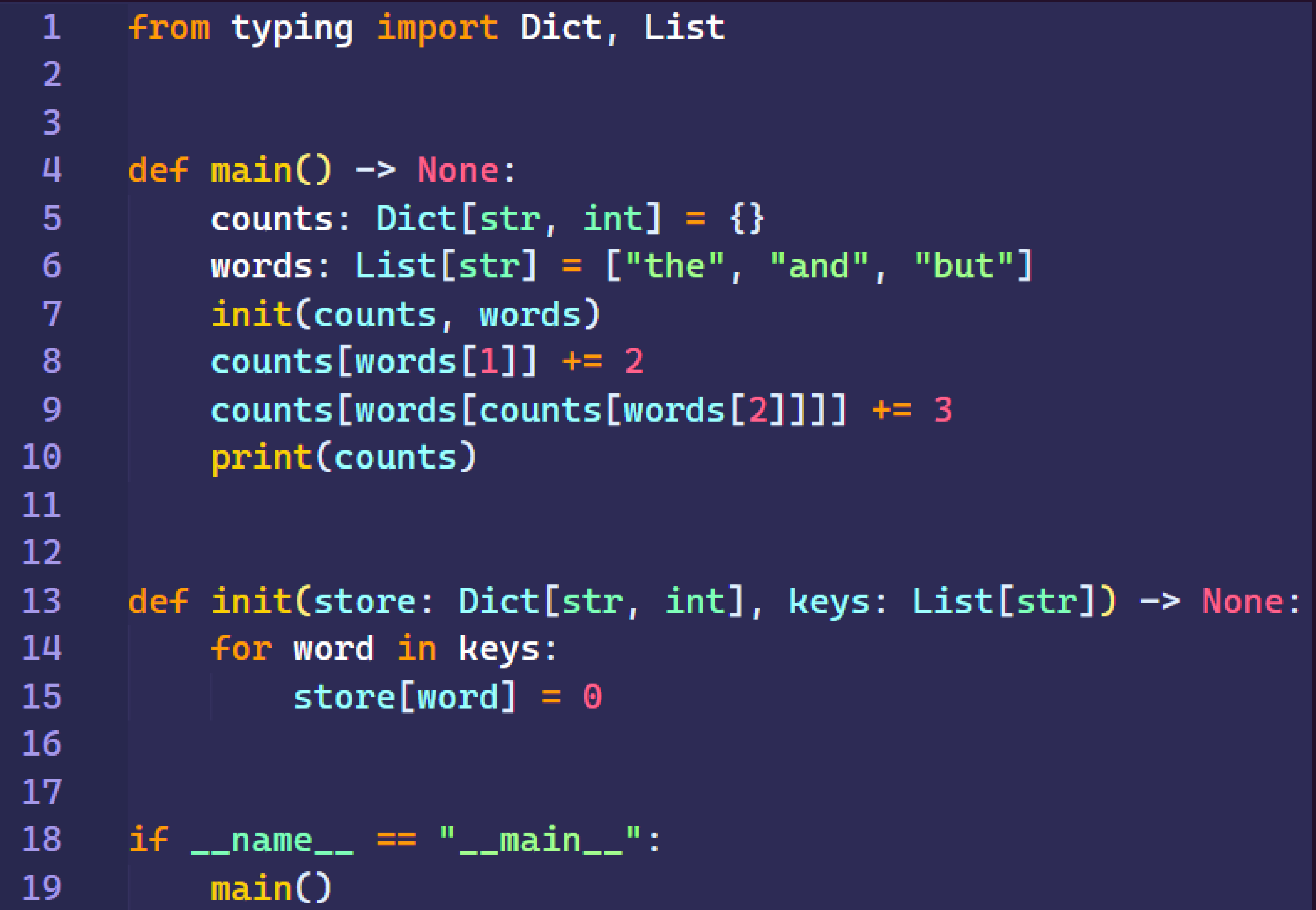
```python
class Frac:
    """A fractional value."""
    n: int
    d: int


def main() -> None:
    a: Frac = Frac()
    a.n = 1
    a.d = 2
    b: Frac = Frac()
    b.n = 1
    b.d = 4
    add(a, b)
    print(f"{a.n}/{a.d}")


def add(lhs: Frac, rhs: Frac) -> None:
    lhs.n = lhs.n * rhs.d + rhs.n * lhs.d
    lhs.d = lhs.d * rhs.d


if __name__ == "__main__":
    main()
```

Q2



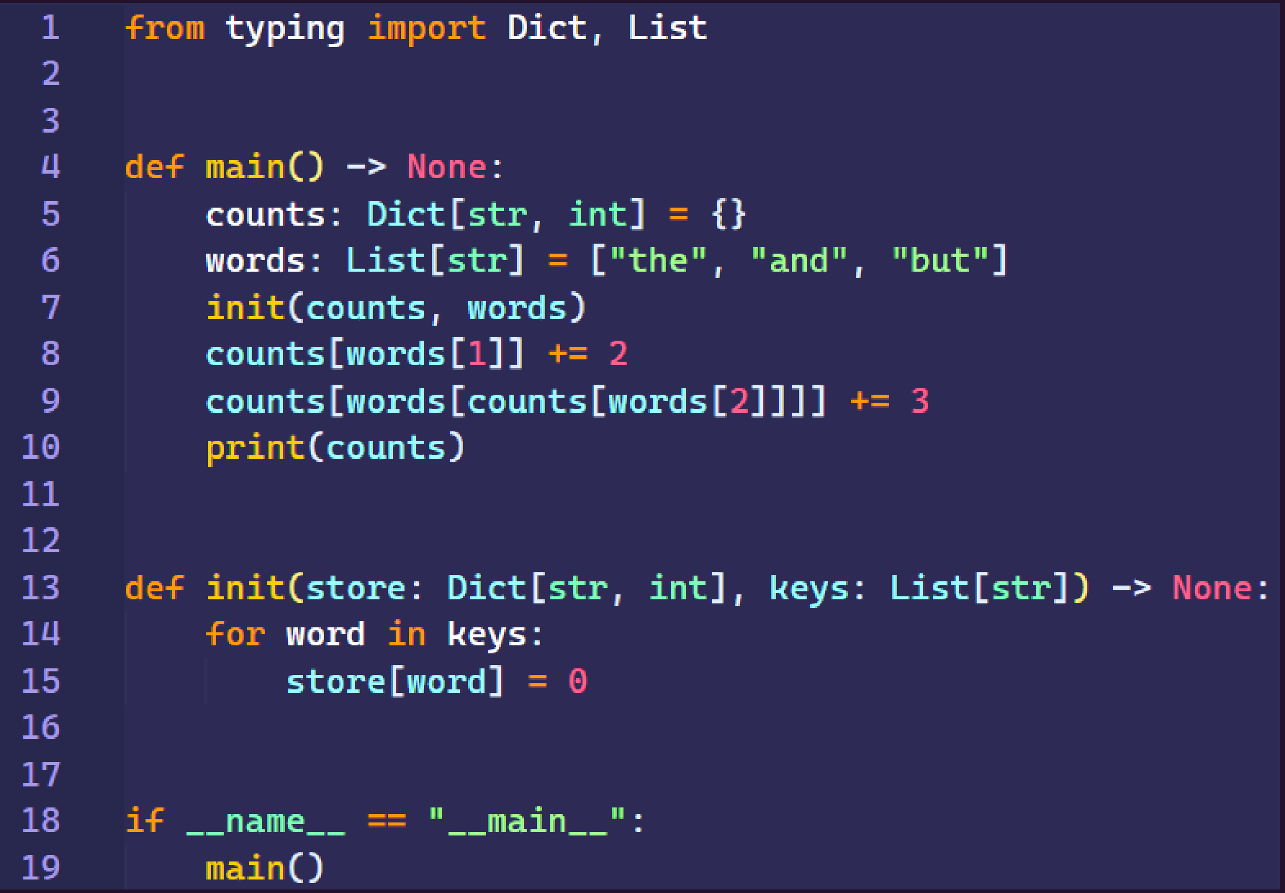
```python
from typing import Dict, List


def main() -> None:
    counts: Dict[str, int] = {}
    words: List[str] = ["the", "and", "but"]
    init(counts, words)
    counts[words[1]] += 2
    counts[words[counts[words[2]]]] += 3
    print(counts)


def init(store: Dict[str, int], keys: List[str]) -> None:
    for word in keys:
        store[word] = 0


if __name__ == "__main__":
    main()
```

Q2



```python
from typing import Dict, List


def main() -> None:
    counts: Dict[str, int] = {}
    words: List[str] = ["the", "and", "but"]
    init(counts, words)
    counts[words[1]] += 2
    counts[words[counts[words[2]]]] += 3
    print(counts)


def init(store: Dict[str, int], keys: List[str]) -> None:
    for word in keys:
        store[word] = 0


if __name__ == "__main__":
    main()
```