Remove Control Flag

Simplify code by replacing control flags with **early returns or breaks**

Code Smell

```
flag = true;
while flag:
  if A: flag = false;   else ...
=>
while flag: # in a loop
  if A: break
=>
while flag: # in a method
  if A: return
```

Example 1 - FindInt

- Find the value (taget) in the List.

```python
def find(data: List[int], target: int) -> bool:
    flag = False
    i = 0
    while i < len(data) and not flag:
        if data[i] == target:
            flag = True
        i += 1
    return flag
```

Refactor by removing control flag

```python
def find(data: List[int], target: int) -> bool:
    found = False
    for i in range(len(data)):
        if data[i] == target:
            found = True
            break
    return found
```

Even better with return (in a method)

```python
def find(data: List[int], target: int) -> bool:
    for i in range(len(data)):
        if data[i] == target:
            return True
    return False
```

Example 2 - SimpleDatabase

Before

```python
self._map: Dict[str, str] = {}
flag = False

while not flag:
    tmp = reader.readline()
    if not tmp:  # EOF
        flag = True
```

After

```python
while True:
    line = reader.readline()
    if not line:  # EOF
        break
```

Tip

However, we can sense smell

```
equal_index = line.find('=')
if equal_index > 0:
    key = line[:equal_index]
    value = line[equal_index + 1:] # ugly and hard to read
    self._map[key] = value
```

Refactoring with Regex

```python
self._pattern = re.compile(r'([^=]+)=(.*)')
match = self._pattern.match(line)

if match:
    key = match.group(1)
    value = match.group(2)
    self._map[key] = value
```

Discussion

Using `break` and `continue` statements always makes code more readable.

No. While they can eliminate control flags, overuse of `break` and `continue` can make code harder to follow. Use them judiciously