

```
1  class Package:
2      def __init__(self, number, sender, recipient, weight):
3          self.number = number
4          self.sender = sender
5          self.recipient = recipient
6          self.weight = weight
7
8
9  def main():
10     packages = [
11         Package(number=1, sender="Alice", recipient="Bob", weight=10),
12         Package(number=2, sender="Bob", recipient="Charlie", weight=5),
13     ]
14
15
16  main()
```

```
1  class Package:
2      def __init__(self, number, sender, recipient, weight):
3          self.number = number
4          self.sender = sender
5          self.recipient = recipient
6          self.weight = weight
7
8
9  def main():
10     packages = [
11         Package(number=1, sender="Alice", recipient="Bob", weight=10),
12         Package(number=2, sender="Bob", recipient="Charlie", weight=5),
13     ]
14     for package in packages:
15         print(
16             f"{package.number}: {package.sender} to {package.recipient}, {package.weight}kg"
17         )
18
19
20 main()
```

```
1 class Package:
2     def __init__(self, number, sender, recipient, weight):
3         self.number = number
4         self.sender = sender
5         self.recipient = recipient
6         self.weight = weight
7
8     @property
9     def number(self):
10         return self._number
11
12     @number.setter
13     def number(self, value):
14         if value < 1:
15             raise ValueError("Number must be positive")
16         self._number = value
17
18
19 def main():
20     packages = [
21         Package(number=1, sender="Alice", recipient="Bob", weight=10),
22         Package(number=2, sender="Bob", recipient="Charlie", weight=5),
23     ]
24     for package in packages:
25         print(
26             f"{package.number}: {package.sender} to {package.recipient}, {package.weight}kg"
27         )
28
29
30 main()
```

```
1 class Package:
2     def __init__(self, number, sender, recipient, weight):
3         self.number = number
4         self.sender = sender
5         self.recipient = recipient
6         self.weight = weight
7
8     @property
9     def number(self):
10         return self._number
11
12     @number.setter
13     def number(self, value):
14         if not isinstance(value, (int, float)):
15             raise TypeError("Number must be a number")
16         if value < 1:
17             raise ValueError("Number must be positive")
18         self._number = value
19
20
21 def main():
22     packages = [
23         Package(number=1, sender="Alice", recipient="Bob", weight=10),
24         Package(number=2, sender="Bob", recipient="Charlie", weight=5),
25     ]
26     for package in packages:
27         print(
28             f"{package.number}: {package.sender} to {package.recipient}, {package.weight}kg"
29         )
30
31
32 main()
```

```
1  class Package:
2      def __init__(self, number, sender, recipient, weight):
3          self.number = number
4          self.sender = sender
5          self.recipient = recipient
6          self.weight = weight
7
8      @property
9      def number(self):
10         return self._number
11
12     @number.setter
13     def number(self, value):
14         if not isinstance(value, (int, float)):
15             raise TypeError("Number must be a number")
16         if value < 1:
17             raise ValueError("Number must be positive")
18         self._number = value
19
20     @property
21     def weight(self):
22         return self._weight
23
24     @weight.setter
25     def weight(self, value):
26         if not isinstance(value, (int, float)):
27             raise TypeError("Weight must be a number")
28         if value < 0:
29             raise ValueError("Weight must be non-negative")
30         self._weight = value
31
32
33 def main():
34     packages = [
35         Package(number=1, sender="Alice", recipient="Bob", weight=10),
36         Package(number=2, sender="Bob", recipient="Charlie", weight=5),
37     ]
38     for package in packages:
39         print(
40             f"{package.number}: {package.sender} to {package.recipient}, {package.weight}kg"
41         )
42
```

43

44 main()