

Wprowadzenie do języka Python - Wykład 5

“Obiektowość”

Programowanie obiektowe w Pythonie

Lego jako model programowanie obiektowego

```

1 class Employee:
2     """Common base class for all employees"""
3     empCount = 0
4
5     def __init__(self, name, salary):
6         self.name = name
7         self.salary = salary
8         Employee.empCount += 1
9
10    def displayCount(self):
11        print("Total Employee %d" % Employee.empCount)
12
13    def displayEmployee(self):
14        print("Name : ", self.name, ", Salary: ",
15              self.salary)
16
17
18 emp1 = Employee("John", 2000)
19 emp2 = Employee("Anna", 5000)

```

Name : John , Salary: 2000

Name : Anna , Salary: 5000

Obsługa błędów

Dokumentacja <https://docs.python.org/3/tutorial/errors.html>

```
10 * (1/0)
```

```
4 + spam*3
```

```
'2' + 2
```

```
1 try:
2     x = int(input("Please enter a number: "))
3     print("number", x)
4 except ValueError:
5     print("Oops! That was no valid number. Try again...")
```

```
1 while True:
2     try:
3         x = int(input("Please enter a number: "))
4         break
5     except ValueError:
6         print("Oops! That was no valid number. Try again...")
7 print("number", x)
```

```
1 try:
2     raise Exception('spam', 'eggs')
3 except Exception as inst:
4     print(type(inst))      # the exception instance
5     print(inst.args)      # arguments stored in .args
6     print(inst)           # __str__ allows args to be printed directly,
7                             # but may be overridden in exception subclasses
8     x, y = inst.args      # unpack args
9     print('x =', x)
10    print('y =', y)
```



```
1 def this_fails():
2     x = 1/0
3 try:
4     this_fails()
5 except ZeroDivisionError as err:
6     print('Handling run-time error:', err)
```

```
1
2 try:
3     raise KeyboardInterrupt
4 finally:
5     print('Goodbye, world!')
6
```

```
1 def divide(x, y):
2     try:
3         result = x / y
4     except ZeroDivisionError:
5         print("division by zero!")
6     else:
7         print("result is", result)
8     finally:
9         print("executing finally clause")
10
11
12 divide(2, 1)
13 divide(2, 0)
14 divide("2", "1")
```

Bibliografia

- <https://realpython.com/python-print/>, dostęp online 5.01.2022
- <https://www.programiz.com/python-programming/operators>, dostęp online 5.01.2022
- <https://realpython.com/python-conditional-statements/>, dostęp online 5.01.2022
- <https://realpython.com/python-for-loop/>, dostęp online 5.01.2022
- <https://realpython.com/python-while-loop/>, dostęp online 5.01.2022
- <https://towardsdatascience.com/a-guide-to-python-comprehensions-4d16af68c97e>,
dostęp online 20.02.2023.
- <https://mypy.readthedocs.io/>, dostęp online 20.03.2023.
- <https://realpython.com/introduction-to-python-generators/>, dostęp online 22.03.2023.

