LU03.L06 - Transformation von Daten

```
def transform_employee_data(employees, transformation_function, *args):
    """
    Higher-Order function that accepts a list of employees and a
transformation function.
    Applies the transformation function to the list of employees.

    :param employees: List of employees, each employee is a dictionary.
    :param transformation_function: The function to be applied to the
employee list.
    :param *args: Additional arguments that may be required by the
transformation function.
    :return: Result of applying the transformation function.
    """
    return transformation_function(employees, *args)


def increase_salary_by_department(employees, department, percentage):
    """
    Increases the salary of all employees in a specific department by a set
percentage.

    :param employees: List of employees
    :param department: Department where the salary should be increased
    :param percentage: Percentage of the salary increase
    :return: List of updated employee data
    """
    for employee in employees:
        if employee['department'] == department:
            employee['salary'] += employee['salary'] * percentage / 100
    return employees


def filter_by_age(employees, age, comparison='greater'):
    """
    Filters employees older or younger than a specific age.

    :param employees: List of employees
    :param age: Age for comparison
    :param comparison: Either 'greater' for older or 'less' for younger
    :return: List of filtered employees
    """
    result = []
    for employee in employees:
        if comparison == 'greater' and employee['age'] > age:
            result.append(employee)
        elif comparison == 'less' and employee['age'] < age:
            result.append(employee)
```

```
    return result


def convert_names_to_uppercase(employees):
    """
    Converts the names of all employees to uppercase.

    :param employees: List of employees
    :return: List of updated employee data
    """
    for employee in employees:
        employee['name'] = employee['name'].upper()
    return employees


if __name__ == '__main__':
    employees = [
        {'name': 'Alice', 'age': 30, 'salary': 5000, 'department': 'HR'},
        {'name': 'Bob', 'age': 40, 'salary': 6000, 'department': 'IT'},
        {'name': 'Charlie', 'age': 25, 'salary': 4000, 'department':
'Sales'},
        {'name': 'David', 'age': 35, 'salary': 5500, 'department':
'Marketing'},
        {'name': 'Eva', 'age': 28, 'salary': 4800, 'department': 'IT'},
        {'name': 'Frank', 'age': 45, 'salary': 6500, 'department':
'Finance'},
        {'name': 'Grace', 'age': 32, 'salary': 5100, 'department': 'HR'},
        {'name': 'Hannah', 'age': 29, 'salary': 4600, 'department':
'Sales'},
        {'name': 'Ivy', 'age': 31, 'salary': 5300, 'department':
'Marketing'},
        {'name': 'Jack', 'age': 50, 'salary': 7000, 'department': 'Finance'}
    ]

    # Test: Increasing salary in the IT department by 10%
    updated_employees = transform_employee_data(employees,
increase_salary_by_department, 'IT', 10)
    print(updated_employees)

    # Test: Filtering employees older than 35
    older_employees = transform_employee_data(employees, filter_by_age, 35,
'greater')
    print(older_employees)

    # Test: Converting names to uppercase
    uppercase_names_employees = transform_employee_data(employees,
convert_names_to_uppercase)
    print(uppercase_names_employees)
```