Department of Telecooperation

JKU
JOHANNES KEPLER
UNIVERSITY LINZ

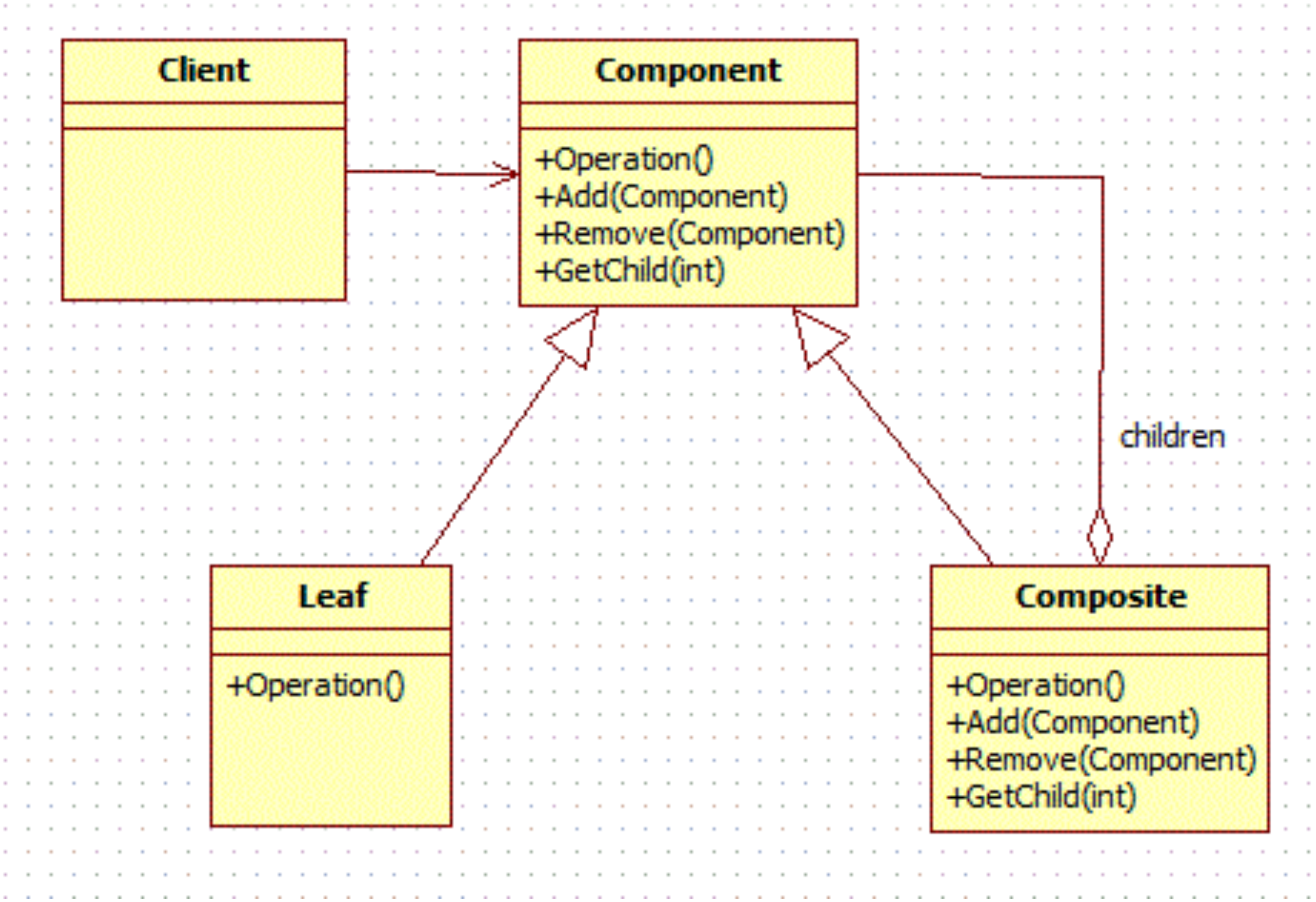
Übung Softwareentwicklung 2 für Wirtschaftsinformatik

Übung 03 C
Composite Pattern

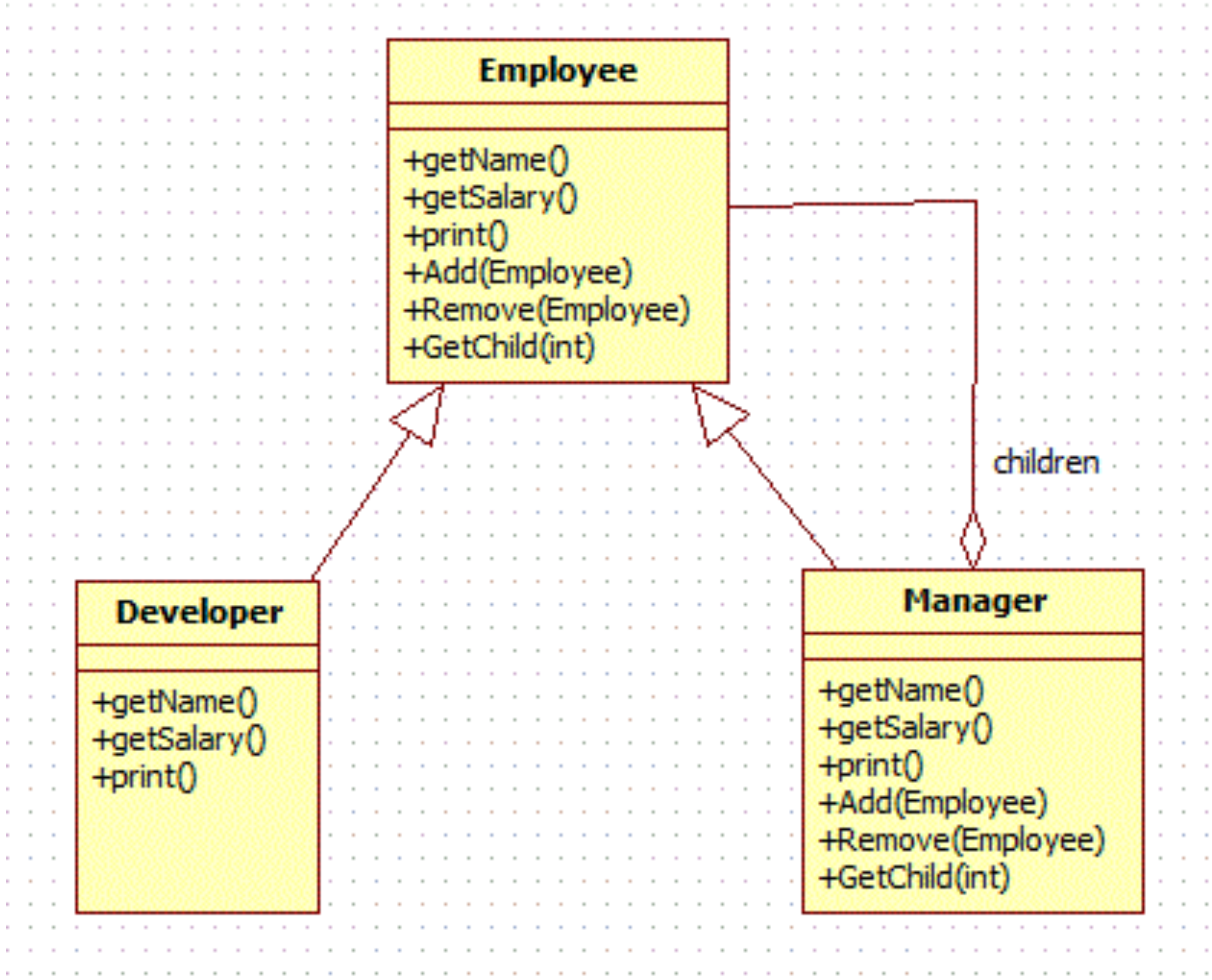
Ismail Khalil
ismail.khalil@jku.at

https://moodle.jku.at/jku/course/view.php?id=8729

Composite Design Pattern



Composite Design Pattern



Composite Design Pattern

```
public interface Employee {
public void add(Employee employee);
public void remove(Employee employee);
public Employee getChild(int i);
public String getName();
public double getSalary();
public void print();
}
```

Composite Design Pattern

```
public class Manager implements Employee{
private String name;
private double salary;
public Manager(String name,double salary){
this.name = name;
this.salary = salary;
}
List<Employee> employees = new ArrayList<Employee>();
public void add(Employee employee) {
employees.add(employee);
}
public Employee getChild(int i) {
return employees.get(i);
}
public String getName() {
return name;
public double getSalary() {
return salary;
}
```

Composite Design Pattern

```
public void print() {
System.out.println("-------------");
System.out.println("Name ="+getName());
System.out.println("Salary ="+getSalary());
System.out.println("-------------");

Iterator<Employee> employeeIterator = employees.iterator();
while(employeeIterator.hasNext()){
Employee employee = employeeIterator.next();
employee.print();
}
}
public void remove(Employee employee) {
employees.remove(employee);
}
}
```