Which one is
better:
a linked list or
an array?

An array, list.
Are there any
other kinds of
data collections?

Collections

Java API Collections Framework

Abstract data types

Generics

Generic methods

Knights have to be military organised :

- Knights can be arranged into
  squads
- A squad can be a part of another
  squad

=> A squad can consist of other
squads and individual knights

Getting the knights out of any military unit
has to be transparent to the client code

Object-oriented means
enable to provide for
structures for an
effective processing of
complex data

Data often have to
survive the
program execution:
they have to be
persistent

Rytieri môžu byť organizovaní
do oddielov s ľubovoľnou
hĺbkou hierarchie,
pričom je potrebné rovnakým
spôsobom z akéhokoľvek
oddielu vybrať všetkých rytierov

# Composite

elements may be composite
and atomic,
and the way of accesing them
has to be uniform

Enabling an equal access
to the hierarchically
organized data

Program raz skončí.
Ako uchovať údaje,
ktoré zostanú po ňom?

Serialization
is a
realization of
persistence

The input/output
system in Java:
an example in exercises

Lecture 5:

# Structured Data Types, Generics, and Persistance

## Valentino Vranić

**Ústav informatiky, informačných systémov
a softvérového inžinierstva**

STU
FIIT

vranic@stuba.sk                    fiit.sk/~vranic

Which one is better:
a linked list or
an array?

An array, list.
Are there any
other kinds of
data collections?

# Collections

## Java API Collections Framework

| Interface | Hash Table | Resizable Array | Balanced Tree | Linked List | Hash Table + Linked List |
|-----------|-----------|-----------------|---------------|-------------|--------------------------|
| Set | HashSet | | TreeSet | | LinkedHashSet |
| List | | ArrayList | | LinkedList | |
| Deque | | ArrayDeque | | LinkedList | |
| Map | HashMap | | TreeMap | | LinkedHashMap |

https://docs.oracle.com/javase/8/docs/technotes/guides/collections/overview.html

# Abstract data types

# Generics

# Generic methods

Kinghts have to be military organized :

- Knights can be arranged into squads
- A squad can be a part of another squad

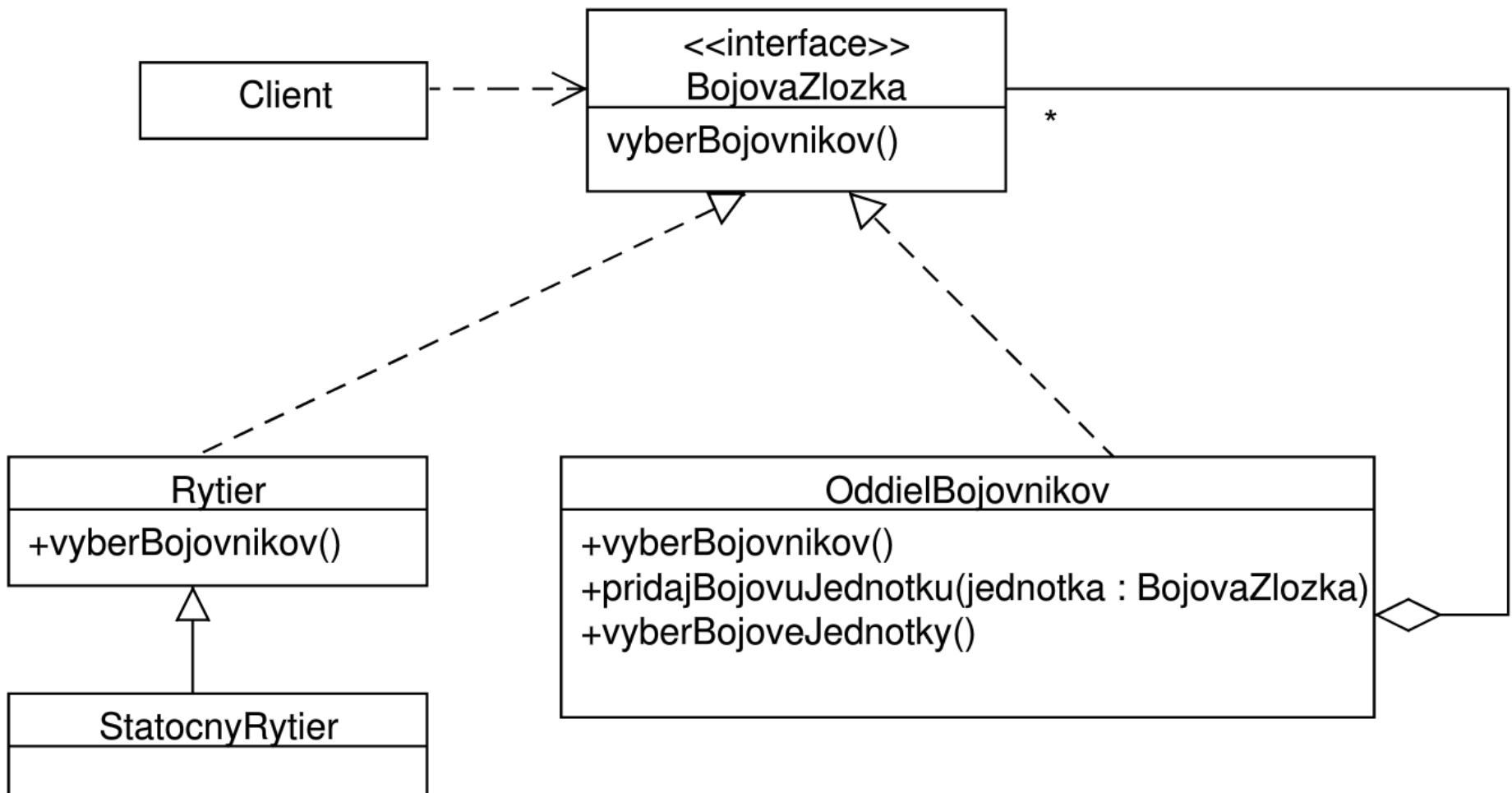=> A squad can consist of other squads and individual knights

Kinghts have to be military organized :

- Knights can be arranged into squads
- A squad can be a part of another squad

=> A squad can consist of other squads and individual knights

Getting the knights out of any military unit has to be transparent to the client code
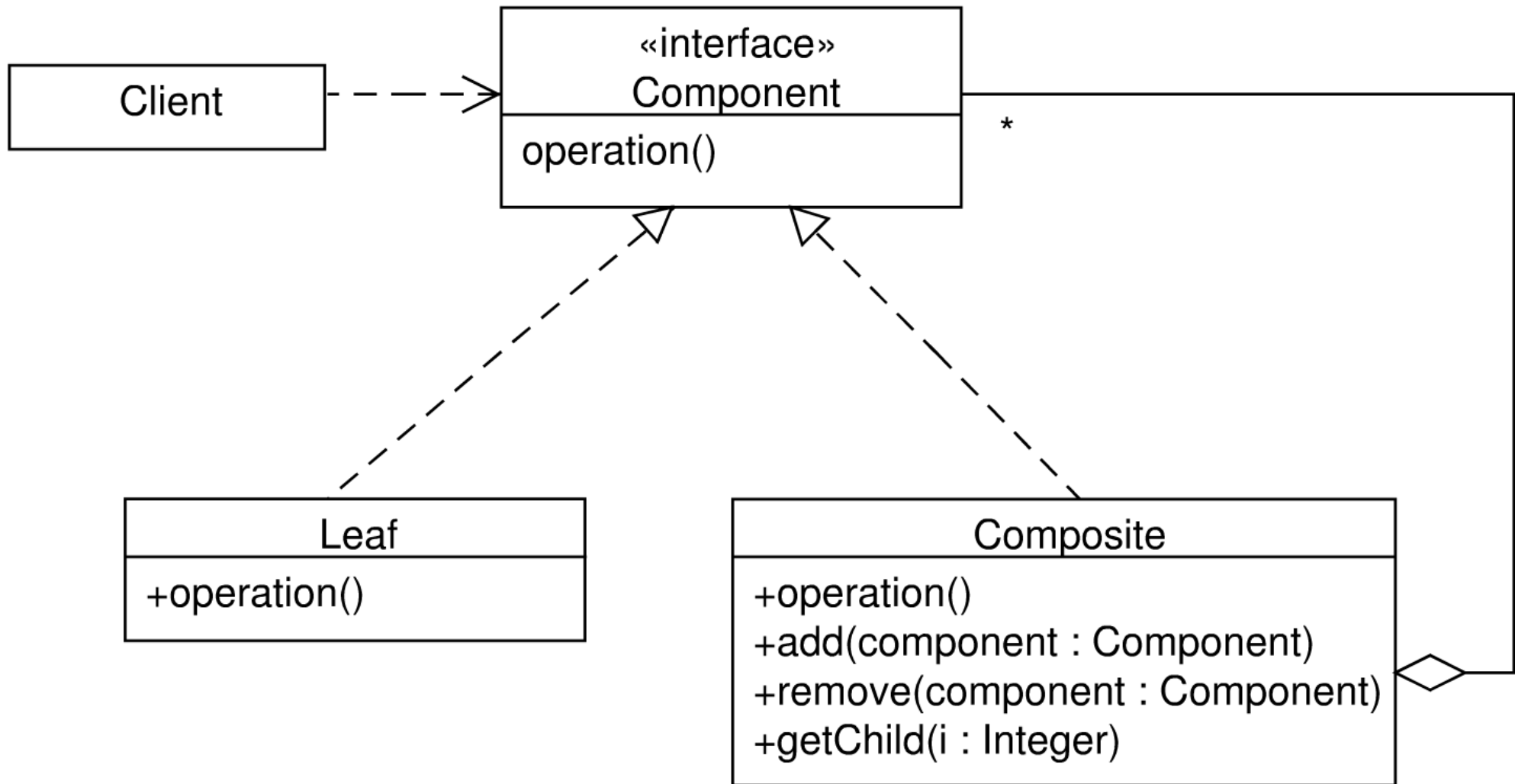
```
                    ┌─────────────────────┐
┌──────────┐        │    <<interface>>    │
│  Client  │------->│    BojovaZlozka     │─────────────────────┐
└──────────┘        ├─────────────────────┤  *                  │
                    │  vyberBojovnikov()  │                      │
                    └─────────────────────┘                      │
                         △         △                             │
                        /           \                            │
                       /             \                           │
                      /               \                          │
┌──────────────────────┐    ┌──────────────────────────────────────────────┐
│       Rytier         │    │              OddielBojovnikov                │
├──────────────────────┤    ├──────────────────────────────────────────────┤
│  +vyberBojovnikov()  │    │  +vyberBojovnikov()                          │
└──────────────────────┘    │  +pridajBojovuJednotku(jednotka : BojovaZlozka)│◇──┘
         △                  │  +vyberBojoveJednotky()                      │
         │                  │                                              │
┌──────────────────────┐    └──────────────────────────────────────────────┘
│    StatocnyRytier    │
├──────────────────────┤
│                      │
└──────────────────────┘
```
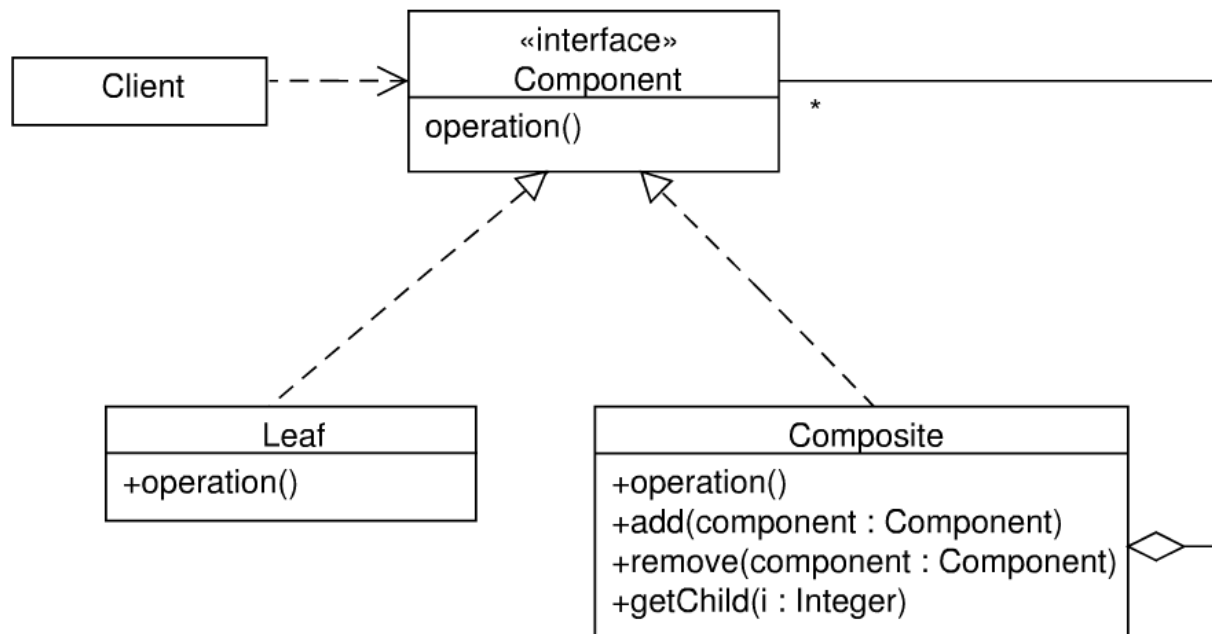
Rytieri môžu byť organizovaní do oddielov s ľubovoľnou hĺbkou hierarchie, pričom je potrebné rovnakým spôsobom z akéhokoľvek oddielu vybrať všetkých rytierov
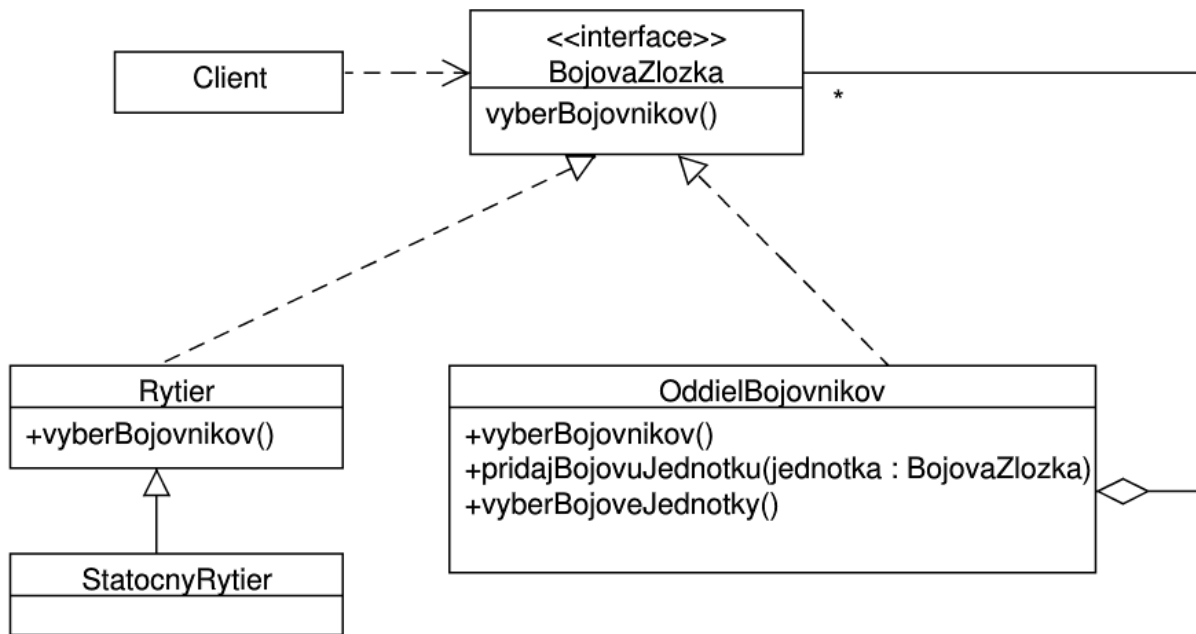
Rytieri môžu byť organizovaní
do oddielov s ľubovoľnou
hĺbkou hierarchie,
pričom je potrebné rovnakým
spôsobom z akéhokoľvek
oddielu vybrať všetkých rytierov

# Composite

elements may be composite
and atomic,
and the way of accesing them
has to be uniform

# Enabling an equal access to the hierarchically organized data

## Diagram 1

```
         Client  - - - - - >    <<interface>>
                                 BojovaZlozka              *
                                vyberBojovnikov()
```

**Rytier**
+vyberBojovnikov()

**StatocnyRytier**

**OddielBojovnikov**
+vyberBojovnikov()
+pridajBojovuJednotku(jednotka : BojovaZlozka)
+vyberBojoveJednotky()

## Diagram 2

```
         Client  - - - - - >    «interface»
                                 Component               *
                                  operation()
```

**Leaf**
+operation()

**Composite**
+operation()
+add(component : Component)
+remove(component : Component)
+getChild(i : Integer)

Object-oriented means enable to provide for structures for an effective processing of complex data

Program raz skončí. Ako uchovať údaje, ktoré zostanú po ňom?

Serialization is a realization of persistence

# The input/output
# system in Java:
# an example in exercises

Data often have to survive the program execution: they have to be persistent

[ Object-oriented means enable to provide for structures for an effective processing of complex data ]

[ Data often have to survive the program execution: they have to be persistent ]