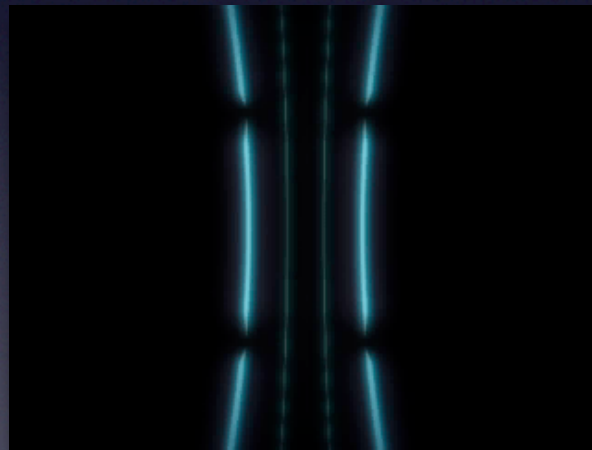


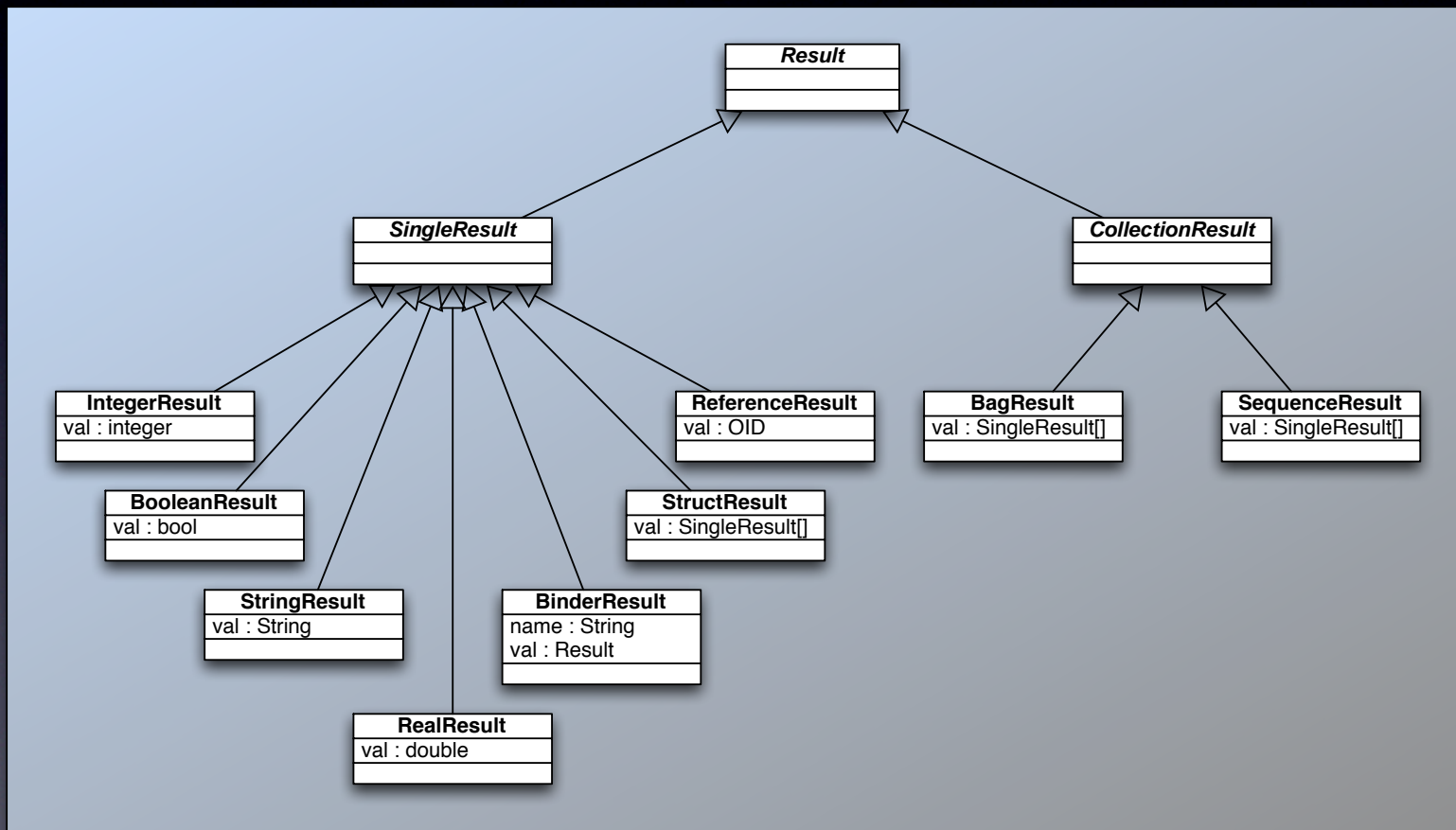


**JPS** ćwiczenia 10.

## Interpreter

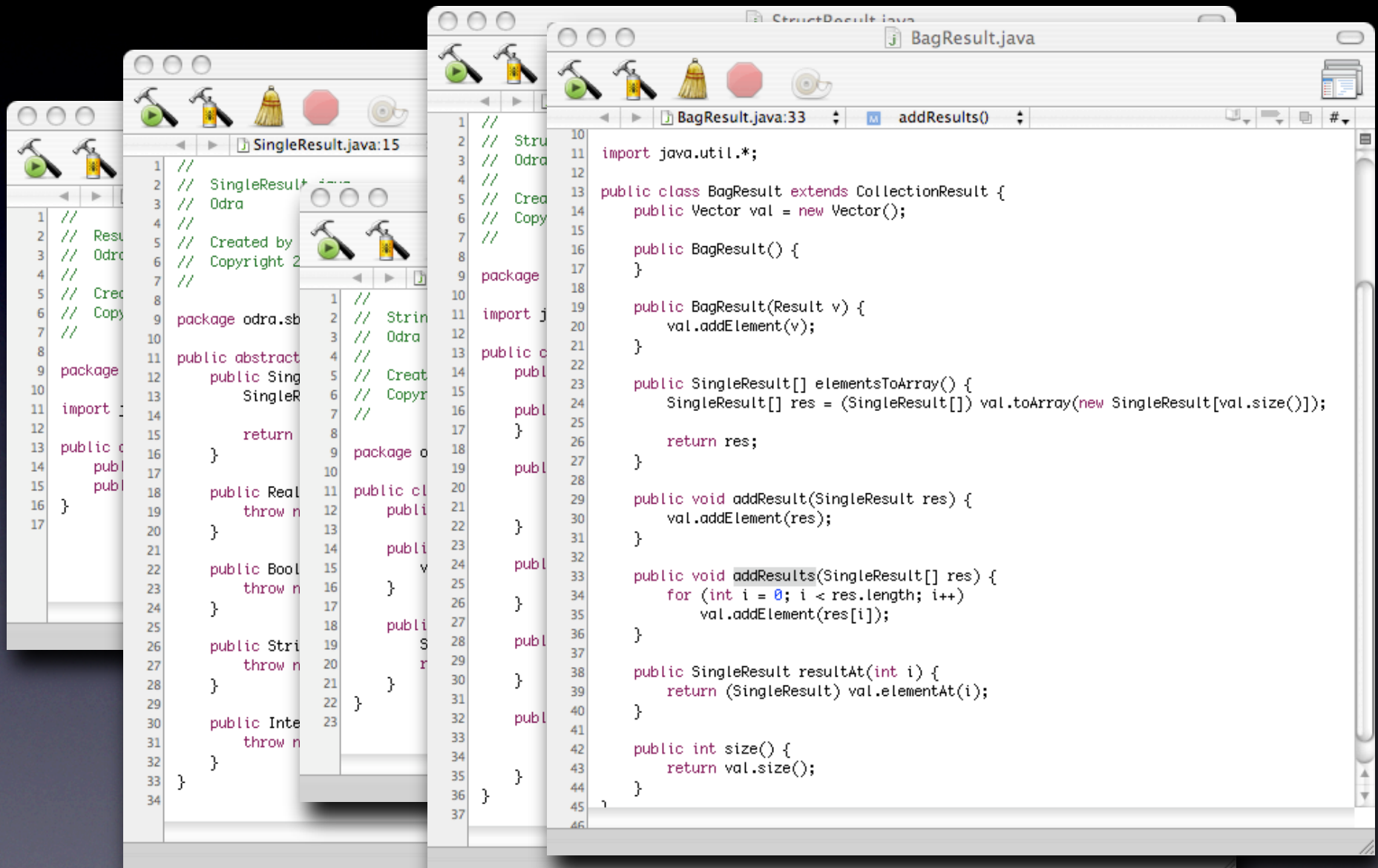


# Rezultaty wyrażeń





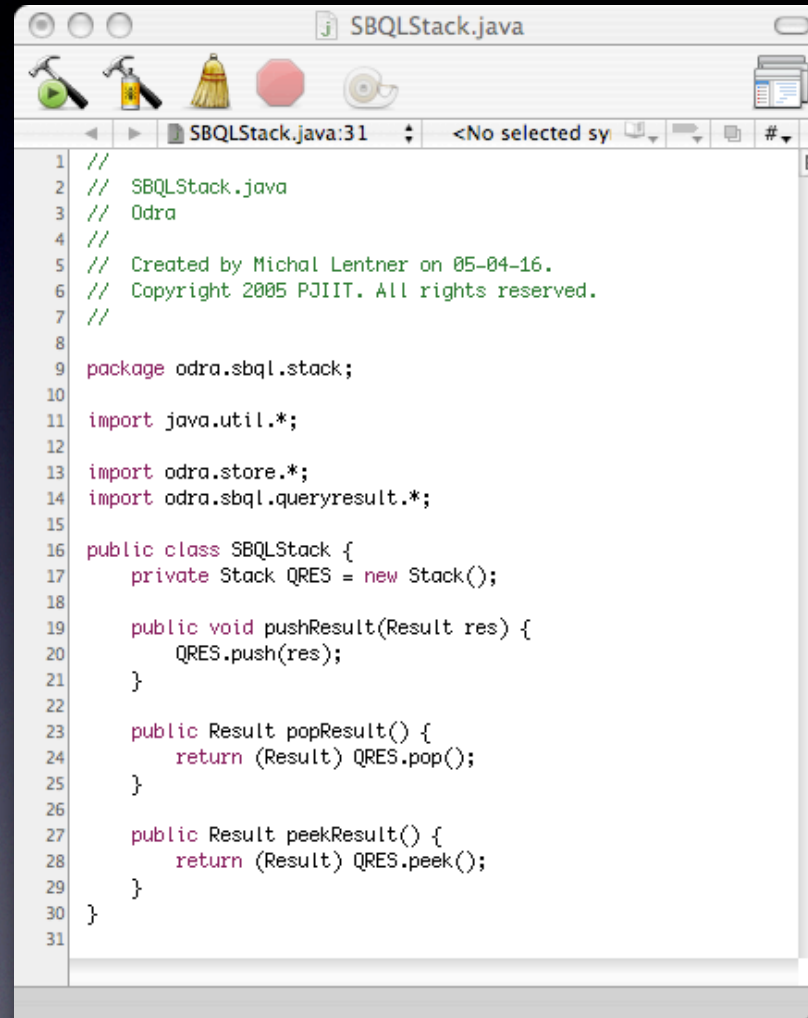
# Przykład implementacji



The image displays three overlapping code editor windows from an IDE, showing Java code. The windows are titled 'SingleResult.java:15', 'StructResult.java', and 'BagResult.java'. The 'BagResult.java' window is the most prominent and shows the following code:

```
10 import java.util.*;
11
12
13 public class BagResult extends CollectionResult {
14     public Vector val = new Vector();
15
16     public BagResult() {
17     }
18
19     public BagResult(Result v) {
20         val.addElement(v);
21     }
22
23     public SingleResult[] elementsToArray() {
24         SingleResult[] res = (SingleResult[]) val.toArray(new SingleResult[val.size()]);
25
26         return res;
27     }
28
29     public void addResult(SingleResult res) {
30         val.addElement(res);
31     }
32
33     public void addResults(SingleResult[] res) {
34         for (int i = 0; i < res.length; i++)
35             val.addElement(res[i]);
36     }
37
38     public SingleResult resultAt(int i) {
39         return (SingleResult) val.elementAt(i);
40     }
41
42     public int size() {
43         return val.size();
44     }
45 }
46
```

# Stos (rezultatów)



```
1 //
2 // SBQLStack.java
3 // Odra
4 //
5 // Created by Michal Lentner on 05-04-16.
6 // Copyright 2005 PJIIT. All rights reserved.
7 //
8
9 package odra.sbql.stack;
10
11 import java.util.*;
12
13 import odra.store.*;
14 import odra.sbql.queryresult.*;
15
16 public class SBQLStack {
17     private Stack QRES = new Stack();
18
19     public void pushResult(Result res) {
20         QRES.push(res);
21     }
22
23     public Result popResult() {
24         return (Result) QRES.pop();
25     }
26
27     public Result peekResult() {
28         return (Result) QRES.peek();
29     }
30 }
31
```

# Metody visit dla prostych wyrażeń

```
public Object visitIntegerExpression(IntegerExpression node, Object obj) {
    stack.pushResult(new IntegerResult(node.L.V));

    return null;
}
```

```
public Object visitBooleanExpression(BooleanExpression node, Object obj) {
    stack.pushResult(new BooleanResult(node.L.V));

    return null;
}
```

```
public Object visitGroupAsExpression(GroupAsExpression node, Object obj) {
    node.E.accept(this, obj);

    Result eres = stack.popResult();

    stack.pushResult(new BinderResult(node.I.V, eres));

    return null;
}
```

```
public Object visitCommaExpression(CommaExpression node, Object obj) {
    BagResult commares = new BagResult();

    node.E1.accept(this, obj);
    node.E2.accept(this, obj);

    Result e2res = stack.popResult();
    Result e1res = stack.popResult();

    SingleResult[] e2sres = (SingleResult[]) e2res.elementsToArray();
    SingleResult[] e1sres = (SingleResult[]) e1res.elementsToArray();

    for (int i = 0; i < e1sres.length; i++) {
        for (int j = 0; j < e2sres.length; j++) {
            StructResult strres = new StructResult();

            strres.addFields(e1sres[i].fieldsToArray());
            strres.addFields(e2sres[j].fieldsToArray());

            commares.addResult(strres);
        }
    }

    stack.pushResult(commares);

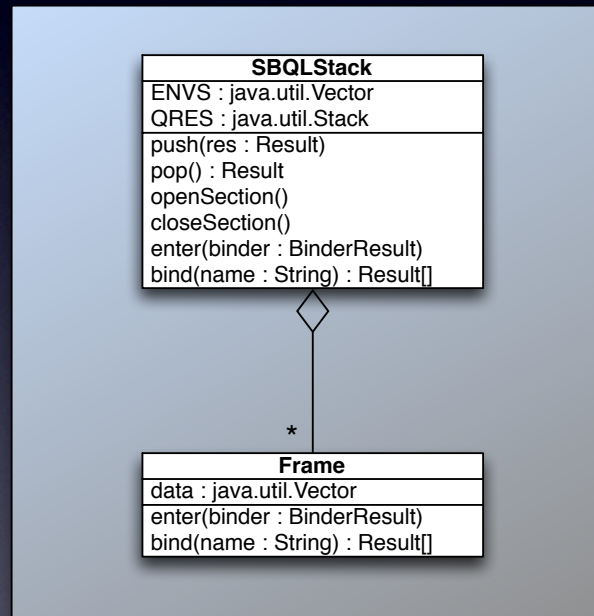
    return null;
}
```



# Problem z visitBinaryExpression

+	IntegerResult	StringResult	BooleanResult	RealResult	StructResult	BinderResult	ReferenceResult	BagResult	SequenceResult
IntegerResult	ok	koercja int->string	błąd	koercja int->real	błąd?	wziąć wartość bindera	dereferencja	błąd?	błąd?
StringResult	koercja int->string	ok	koercja bool->string	koercja real->string	błąd?	wziąć wartość bindera	dereferencja	błąd?	błąd?
BooleanResult	błąd	koercja bool->string	ok	błąd	błąd?	wziąć wartość bindera	dereferencja	błąd?	błąd?
RealResult	koercja int->real	koercja real->string	błąd	ok	błąd?	wziąć wartość bindera	dereferencja	błąd?	błąd?
StructResult	błąd?	błąd?	błąd?	błąd?	błąd?	błąd	błąd	błąd	błąd
BinderResult	wziąć wartość bindera	wziąć wartość bindera	wziąć wartość bindera	wziąć wartość bindera	błąd	wziąć 2 wartości binderów	dereferencja, wartość bindera	błąd	błąd
ReferenceResult	dereferencja	dereferencja	dereferencja	dereferencja	błąd	dereferencja, wartość bindera	2 dereferencje	błąd	błąd
BagResult	błąd?	błąd?	błąd?	błąd?	błąd	błąd	błąd	błąd	błąd
SequenceResult	błąd?	błąd?	błąd?	błąd?	błąd	błąd	błąd	błąd	błąd

# Stos (rezultatów i środowisk)





# Operator .

```
public Object visitDotExpression(DotExpression node, Object obj) {
    BagResult dotres = new BagResult();

    node.E1.accept(this, obj);

    SingleResult[] e1sres = stack.popResult().elementsToArray();

    for (int i = 0; i < e1sres.length; i++) {
        createEnvironment(e1sres[i]);
        node.E2.accept(this, obj);
        deleteEnvironment();

        SingleResult[] e2sres = stack.popResult().elementsToArray();

        dotres.addResults(e2sres);
    }

    stack.pushResult(dotres.size() == 1 ? (Result) dotres.resultAt(0) : (Result) dotres);

    return null;
}
```

```
private void createEnvironment() {
    stack.openSection();
}

private void createEnvironment(Result res) {
    stack.openSection();
    nestedM0(res);
}

private void deleteEnvironment() {
    stack.closeSection();
}
```

# Metoda nested

```
private void nestedM0(Result res) {
    if (res instanceof ReferenceResult) {
        ReferenceResult rres = (ReferenceResult) res;

        OID[] subobj = store.getNestedObjects(rres.val);

        for (int i = 0; i < subobj.length; i++)
            stack.enter(new BinderResult(store.getObjectByName(subobj[i]), new ReferenceResult(subobj[i])));
    }
    else if (res instanceof StructResult) {
        StructResult ssres = (StructResult) res;

        for (Enumeration e = ssres.val.elements(); e.hasMoreElements(); )
            nestedM0((Result) e.nextElement());
    }
    else if (res instanceof BinderResult) {
        BinderResult bres = (BinderResult) res;

        stack.enter(bres);
    }
}
```

```
public OID[] getNestedObjects(OID oid) {
    SBAObject obj = (SBAObject) data.get(oid);

    switch (obj.getObjectType()) {
        case SBAObject.SBA_OBJECT_SPIDER:
            SpiderObject sobj = (SpiderObject) obj;

            return (OID[]) (sobj.val.toArray(new OID[sobj.val.size()]));

        case SBAObject.SBA_OBJECT_REFERENCE:
            ReferenceObject robj = (ReferenceObject) obj;

            OID[] rval = { robj.val };

            return rval;

        default:
            throw new RuntimeException("Unexpected object");
    }
}
```

Ćwiczenia