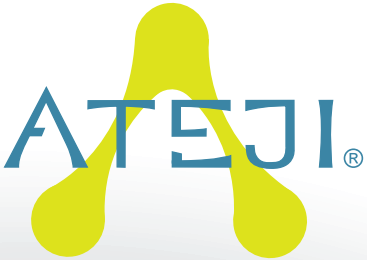
Experts in Optimization, Java Developers, Business Analysts



Object-Oriented Modeling with OptimJ™

OptimJ is an extension
of the **Java**™ programming language
with a **full-featured modeling language** for optimization.
It is available as a plug-in for the widely used **Eclipse**™ IDE.

- Shorter development times
- Better overall quality
- Reusability
- Fast learning curve
- Teamwork
- Integration

```
minimize
    sum(int j : FOOD) ( cost[j]
```

```
var Country c1, c2;
constraints {
    c1.color() != c2.color();
}
```

```
constraints {
    forall(int j : FOOD) {
        Buy[j] >= f_min[j];
```

```
maximize range: v0*v0 * sin(2*alpha) / g;
```

FEATURES

OptimJ is an intuitive and easy to master modeling language

It features decision variables, constraints, aggregate operators such as sums, aggregate constraints such as forall, global constraints and tuples, all with a familiar syntax.

OptimJ is also a full-featured programming language

The full power of the Java language and libraries is available everywhere inside your model, and you won't need a separate scripting language.

OptimJ works directly on your data and object model

A model is a class, decision variables range over any Java type, and constraints refer directly to your existing object model, without expensive runtime conversions.

OptimJ integrates optimization code and application code

OptimJ and Java source files co-exist in the same project. The application directly accesses model data, and the model directly accesses application data.

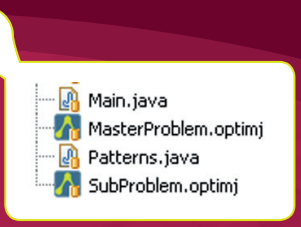
OptimJ brings software engineering techniques and tools to optimization

OO techniques make your optimization code more organized, reusable and scalable. You can edit, run and debug within Eclipse using state-of-the-art quality assurance and productivity tools.

OptimJ is not tied to any particular optimization engine

There are no arbitrary restrictions on the expression of constraints, and compile-time drivers can integrate any optimization engine without computational overhead.

```
* Buy[j] );
```




```
public model Diet solver lpsolve
{
```

```
Diet myModel = new Diet(data);
```

```
var double[] Buy[FOOD.length];
```

BENEFITS

Fast installation and set-up

Simply install the OptimJ plug-in in your Eclipse IDE using the Eclipse Update Manager and immediately start writing OptimJ code : everything behaves as usual.

Talent availability

The OptimJ language looks familiar to both Java experts and optimization experts. This translates into easier team building and faster project start-up times.

Productivity

Never again spend days writing boring integration code. Structure and reuse your OptimJ code with object-oriented techniques and benefit from state-of-the-art IDE support.

Quality

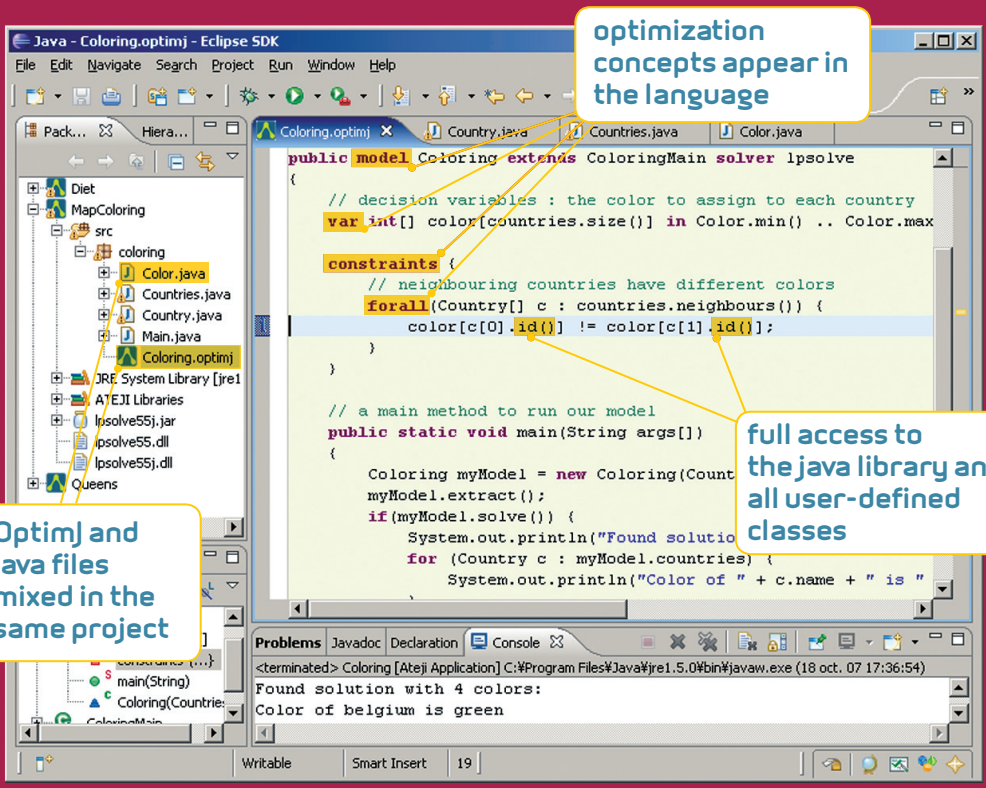
Catch mistakes early with Java static typing and quality assurance tools, and debug across the application/model boundary. No integration code means no bug in the integration code.

Efficiency

OptimJ does all its work at compile-time. Extracting a model or passing data back and forth between application and model incurs no computational overhead at runtime.

Team work

The optimization expert can concentrate on optimization, the programmer can concentrate on programming, they communicate using the same language and share their work on a common software repository.



ATEJI

Language tools for software productivity

Ateji is the world's first provider of integration at the language level.

www.ateji.com