

# Modelim algjebrik: OptimJ

Faton Berisha



Fakulteti i Shkencave Kompjuterike  
Universiteti i Prizrenit

## Qëllimet dhe objektivat

- Instalimi i një gjuhe modeluese për optimizim të mbështetur mbi Java: OptimJ

# Përmbajtja

- 1 Një gjuhë modeluese për optimizim: OptimJ
- 2 Konstruktet gjuhësore për modelim
  - Hyrje: Konceptet optimizuese në Java
  - Variablat vendimmarrëse
  - Kufizimet
  - Objektivi
- 3 Shembuj optimizimi

## Instalimi i OptimJ

- Gjuhe modeluese për optimizim të mbështetur mbi Java (Java-based Modeling Language for Optimization): *OptimJ*
- Instalimi mbi Eclipse IDE
- OptimJ është zgjerimi i Java
  - Vepron drejtpërdrejt mbi Java objekte dhe mund të kombinohet me çfarëdo klase tjetër në Java.
  - E tërë Java biblioteka është në dispozicion
  - Ndërvepruese me shumicën e veglave programuese të mbështetura mbi Java (p.sh., unit testing, GUI designers)
- OptimJ është gjuhë modeluese
  - Koncepte të modelimit algjebrik të cilat gjenden në gjuhët modeluese (AIMMS, AMPL, MPL etj.)
  - OptimJ duhet të mbështetet nga një engine optimizues (zgjidhës, *solver*) që ofron Java API (Ip\_solve, glpk, Gurobi, Mosek, CPLEX)

# OptimJ si gjuhë modeluese me funksion të plotë

- Shpreh modele optimizuese në notacion natyror me orientim matematik
- Është e pavarur nga solver
  - Vetë gjuha mund të shprehë çfarëdo modeli optimizimi
  - Vetëm pas përcaktimit të solver merren parasysht kufizimet përkatëse
- Lejon qasje tërë API të solver
- Pjesa modeluese e gjuhës ofron
  - Modele, variabla vendimmarrjeje, kufizime
  - Operatorë agregatë
  - Deklarime të përgjithësuara
  - Kufizime globale dhe aso të rendit të lartë
  - $n$ -she të renditura
- Mundëson modelim të orientuar nga objekte

## OptimJ si zgjerim i gjuhës programuese Java

- Çdo Java shprehje e tipit `boolean` është një kufizim.
- Një model OptimJ dhe një aplikacion Java funksionojnë së bashku në mënyrë gjenerike.
- Kodi i modelizimit vepron drejtpërdrejt mbi aplikacionin
- Të gjitha konstruktet e gjuhës Java janë në dispozicion brenda modeleve
- E tërë biblioteka e Java është në dispozicion brenda modeleve

## OptimJ e integruar në Eclipse IDE

- Editimi, kompilimi, dibagimi i zakonshëm
- Procesi i zhvillimit i pandryshuar me veglat e zakonshme

## Konceptet optimizuese në Java

```
package examples;
```

```
// Një model i thjeshtë për problemin e ngjyrosjes së hartës  
public model SimpleColoring solver lpsolve {  
    // Numri maksimal i ngjyrave  
    int nbColors = 4;  
    // Variablat vendimmarrëse ruajnë ngjyrën e secilit vend  
    var int belgium in 1 .. nbColors;  
    var int denmark in 1 .. nbColors;  
    var int germany in 1 .. nbColors;  
    // Vendet fqinje të kenë ngjyra të ndryshme  
    constraints {  
        belgium != germany;  
        germany != denmark;  
    }  
}
```



## Konceptet optimizuese në Java (Vazhdim.)

```
// Pika inicuese për testimin e modelit
public static void main(String[] args) {
    // Instanco modelin
    SimpleColoring m = new SimpleColoring();
    // ... pastaj zgjidhe atë
    m.extract();
    m.solve();
    // Shtyp zgjidhjet
    System.out.println("Belgium: " + m.value(m.belgium));
    System.out.println("Denmark: " + m.value(m.denmark));
    System.out.println("Germany: " + m.value(m.germany));
}
}
```

## Konceptet optimizuese në Java (Vazhdim.)

- Konstrukte shitesë ndaj Java: `model`, `var`, `constraints`
- Konstrukte shitesë ndaj gjuhëve modeluese: `package`, `main()`
- **Model:** Java klasë e zgjeruar me koncepte specifike për optimizimin.
  - Modelet mund të përdoren sikur klasat tjera.
  - Modelet janë klasa `final`.
  - Për më tepër, modelet mund të definojnë *variabla vendimmarrëse*, *kufizime* dhe *objektiva*.

## Konteksti i solver-it

- OptimJ është gjuhë e disenjuar pavarësisht nga ndonjë solver ose makinë zidhëse.
- Por, kur modeli tipizohet me një solver, lejohen vetëm modele që kuptohen nga solver-i.
- Konteksti i një solver-i përcaktohet nga deklarimi i modelit me fjalën kyçe `solver`.

## Konteksti i solver-it (Vazhdim.)

- Skopi i kontekstit të solver-it është fushat e modelit, bloqet e kufizimeve dhe klauza e objektivit.
- Konteksti i solver-it vë restriksione mbi tipin e lejuar për variablat vendimmarrëse, mbi operacionet e lejuara për shkruarjen e kufizimeve dhe mbi strukturën e kufizimeve (lineare, kuadratike etj.)
- Konteksti i solver-it specifikon metodat që mund të invokohen mbi variablat vendimmarrëse, objektin solver që mund të qaset përmes solver dhe metodat që ofron ai.

## Variablat vendimmarrëse

- Variablat (imperative) janë emra që u refereoohen hapësirave memorike, ku mund të ruhen vlera.
- *Variablat vendimmarrëse* deklarohen pas fjalës kyçe var. Janë vende për ruajtjen e zgjidhjes së problemit

```
abstract model VarTypes {  
    var float f;  
    var double d;  
    var int i;  
    var boolean b;  
    var int i2 in 0 .. 10;  
    var double d2 in 0.0 .. 10.0;  
    String[] stringDomain = { "abc", "def" };  
    var String s2 in stringDomain;  
}
```

# Kufizimet

- *Kufizim*: Një Java shprehje e tipit `boolean` ku referencohen variabla vendimmarrëse.
- Kufizimet definojnë brenda `constraints` bloqeve.

```
abstract model BasicConstraints {  
    var int i1, i2;  
    var double d1, d2;  
    constraints {  
        2 * i1 != i2 + 3;  
        Math.sin(d1) < d2*d2;  
    }  
}
```

# Objektivi

- *Objektivi*: Java shprehje numerike pas fjalës kyçe minimize ose maximize ku referencohet ndonjë variabël vendimmarrëse
- Objektivi shkruhet në model kur dëshirohet të minimizohet ose maksimizohet një shprehje.

```
abstract model BasicObjective {  
    var double d1, d2;  
    constraints {  
        d1 >= 0;  
        d2 >= 0;  
        d1 + d2 == 5;  
    }  
    maximize d1 - d2;  
}
```

## Shembuj optimizimi

### Shembull

Një fermer ka 75 hektarë për të mbjellur dy kultura: grurë dhe elb. Për të prodhuar të lashtat, kostot e farmerit janë 120 € për hektar për grurë dhe 210 € për hektar për elb. Farmeri ka 15000 € në dispozicion. Por pas korrjes farmeri duhet të ruajë të korrat për të pritur kushte të favorshme tregu. Ai ka hapësirë deponimi prej 4000 bushel. Çdo hektar jep mesatarisht 110 bushel grurë dhe 30 bushel elb. Në qoftë se profiti për bushel gruri është 1.30 € dhe për bushel elbi 2.00 €, si duhet t'i mbjellë farmeri të 75 hektarët për të maksimizuar profitin.



## Zgjidhje

```
package examples;
public model LinearOptimization solver lpsolve {

    var double x1;
    var double x2;

    constraints {
        120*x1 + 210*x2 <= 15000;
        110*x1 + 30*x2  <= 4000;
        x1 +      x2   <= 75;
        x1              >= 0;
        x2              >= 0;
    }

    maximize 110*1.3*x1 + 30*2.0*x2;
```

## Zgjidhje (Vazhdim.)

```
public static void main(String[] args) {  
  
    com.ateji.optimj.lpsolverlib.LibraryLoader.load();  
  
    LinearOptimization m = new LinearOptimization();  
  
    try {  
        m.extract();  
        if (m.solve()) {  
            System.out.println("z= " + m.objValue());  
            System.out.println("x1= " + m.value(m.x1));  
            System.out.println("x2= " + m.value(m.x2));  
        } else {  
            System.out.println("No solution.");  
        }  
    }  
}
```

## Zgjidhje (Vazhdim.)

```
        finally {  
            m.dispose();  
        }  
    }  
}
```

## Udhëzime për lexim të mëtejme

- <http://www.fberisha.org>
- [www.ateji.com/optimj](http://www.ateji.com/optimj)
- <http://lpsolve.sourceforge.net/5.5/>
- *The OptimJ manual*
- *lp\_solve reference guide menu*
- D. Schmidt, *Programming principles in Java: architectures and interfaces*

# Përfundim

- Instalimi i OptimJ
- Java dhe OptimJ
- Modelimi dhe zgjidhja e problemeve algjebrike me anë të OptimJ