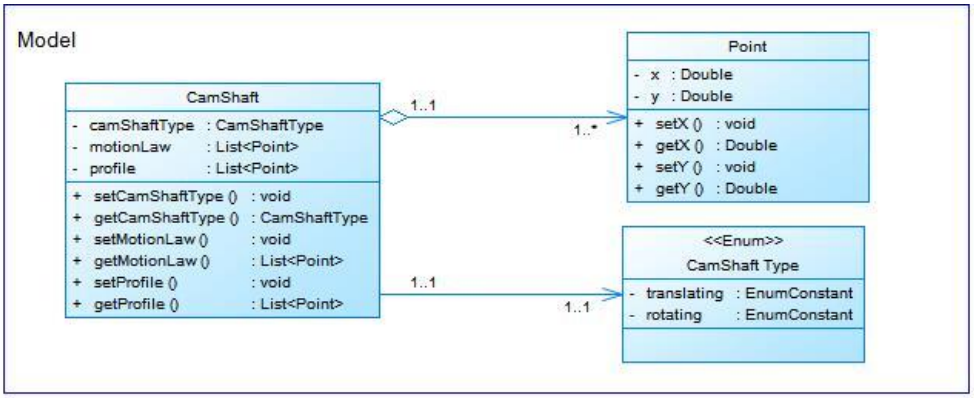
Development goal

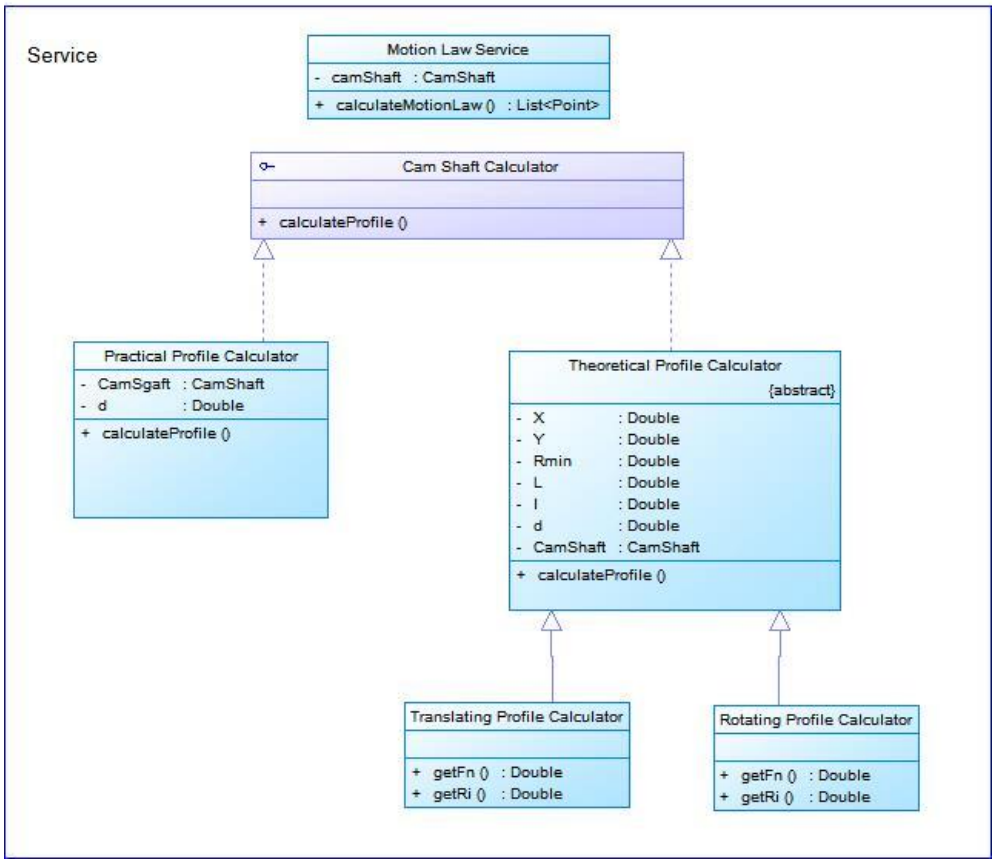
Development goal is creating application Desktop CamShaft. This is for calculating practical and theoretical profile of cam. User selects one of the two available types of cam (translating or rotating), parameters and motion law. The result of program is the visual representation of cam profile.

Application area

The application can be used by students for educational purposes and mechanics for the design profile of cam with any parameters and motion law.

Architectural design





Used technology and approaches

Used technology:

- IDE – IntelliJ IDEA;
- Assembly/deploy – Maven;
- GUI – JavaFX;
- Testing – JUnit 4;
- Build expression from string - exp4j;

Used approaches:

The project was built piecemeal. First developed GUI. Later module «Geometry», which is responsible for determining the basic parameters of the cam. The next module - «Motion Law» allows to set, save, and edit law of motion. The main module - «Profile» calculates the coordinates of theoretical and practical profile of the cam mechanism. The final part of the project is Unit-tests.

Pattern MVC has been used for the design. Therefore, all part of the project themselves independent and bound by controllers only.

CamShaft API

com.amcbridge.camshaft.model.CamShaft	Contains all data about cam.
com.amcbridge.camshaft.model.CamShaftType	Enum. Types of follower.
com.amcbridge.camshaft.model.Point	2D Coordinates of point (x, y).
com.amcbridge.camshaft.service.CamShaftCalculator	Interface. Provides method “calculate”.
com.amcbridge.camshaft.service.PracticalProfileCalculator	Calculate practical profile.
com.amcbridge.camshaft.service.TheoreticalProfileCalculator	Abstract. Calculate theoretical profile.
com.amcbridge.camshaft.service.TranslatingProfileCalculator	Calculate theoretical profile with translating follower.
com.amcbridge.camshaft.service.RotatingProfileCalculator	Calculate theoretical profile with rotating follower.
com.amcbridge.camshaft.service.CommonCamShaftCalculator	Build object contains theoretical and practical calculators.
com.amcbridge.camshaft.service.MotionLawService	Static class for manage law of motion.

MotionLawService API

Name of method	Parameters	Return
calculateMotionLaw	String, double	List<Point>
writeMotionLawToFile	java.io.File	java.io.File
readMotionLawFromFile	java.io.File	List<Point>

Example

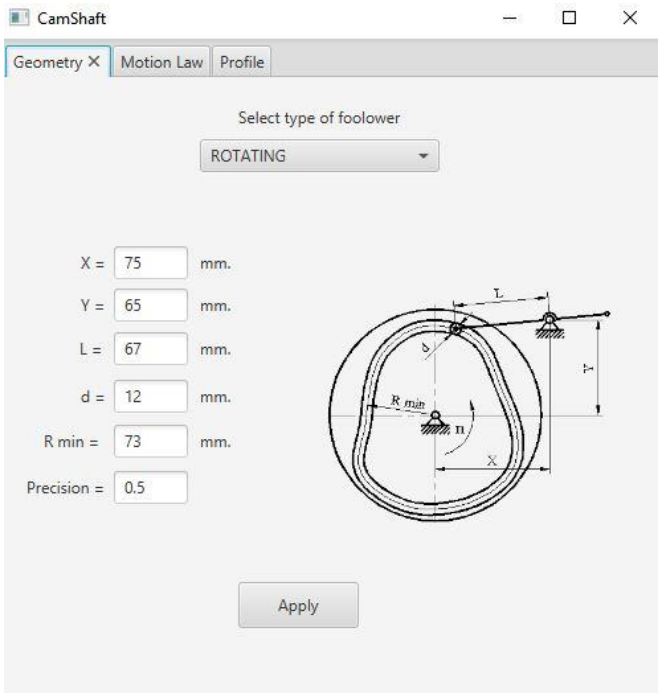
```
CamShaft camshaft = newInstance();
camshaft.setType(CamShaftType.TRANSLATING);
camshaft.setParameters(parameters);
// parameters – Map contains parameters of cam.
camshaft.setMotionLaw(motionLaw);
//motionLaw – List of Point contains data about motion law.
CamShaftCalculator camShaftCalculator = CommonCamShaftCalculator.build(camShaft);
camShaftCalculator.calculateProfile();
```

User manual

The workflow include 3 main steps:

1) Chosen type of cam and fill all geometry data.
2) Input motion law of follower;
3) Representation of camshaft profile.

To run application need to execute file CamShaft.jar. The window will be shown after application start up.



Input geometry parameters

Here you mast select type of follower and input all parameters. After that press “Apply” and click on “Motion Law” tab.

There are 3 ways to set law of motion:

1) Click “Function” and input expression and step;

   the following operators are supported:

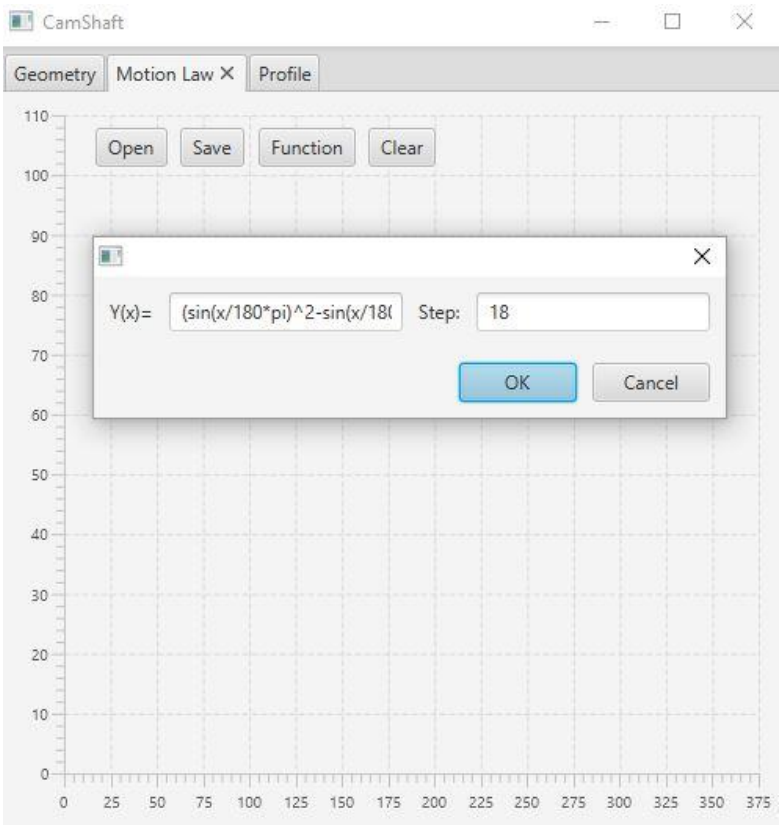
   - Addition: '2 + 2'
   - Subtraction: '2 - 2'
   - Multiplication: '2 * 2'
   - Division: '2 / 2'
   - Exponential: '2 ^ 2'
   - Unary Minus,Plus (Sign Operators): '+2 - (-2)'
   - Modulo: '2 % 2'

   the following functions are supported:

   - abs: absolute value
   - acos: arc cosine
   - asin: arc sine
   - atan: arc tangent
   - cbrt: cubic root
   - ceil: nearest upper integer
   - cos: cosine
   - cosh: hyperbolic cosine
   - exp: euler's number raised to the power (e^x)
   - floor: nearest lower integer
   - log: logarithmus naturalis (base e)
   - log2: logarithm to base 2
   - log10: logarithm to base 10
   - sin: sine
   - sinh: hyperbolic sine
   - sqrt: square root
   - tan: tangent
   - tanh: hyperbolic tangent
   - signum: signum of a value

2) Click “Open” and select file in your file system;

3) Double click on chart creates new point;



Input law of motion

Finally, click on tab “Profile” and press “Build”.