Go Java Core Project

Group #9

“**Fire Phoenix**”

Calculator for two-dimensional arrays

Members:

- Aleksey Timonov
- Viktor Mazurenko

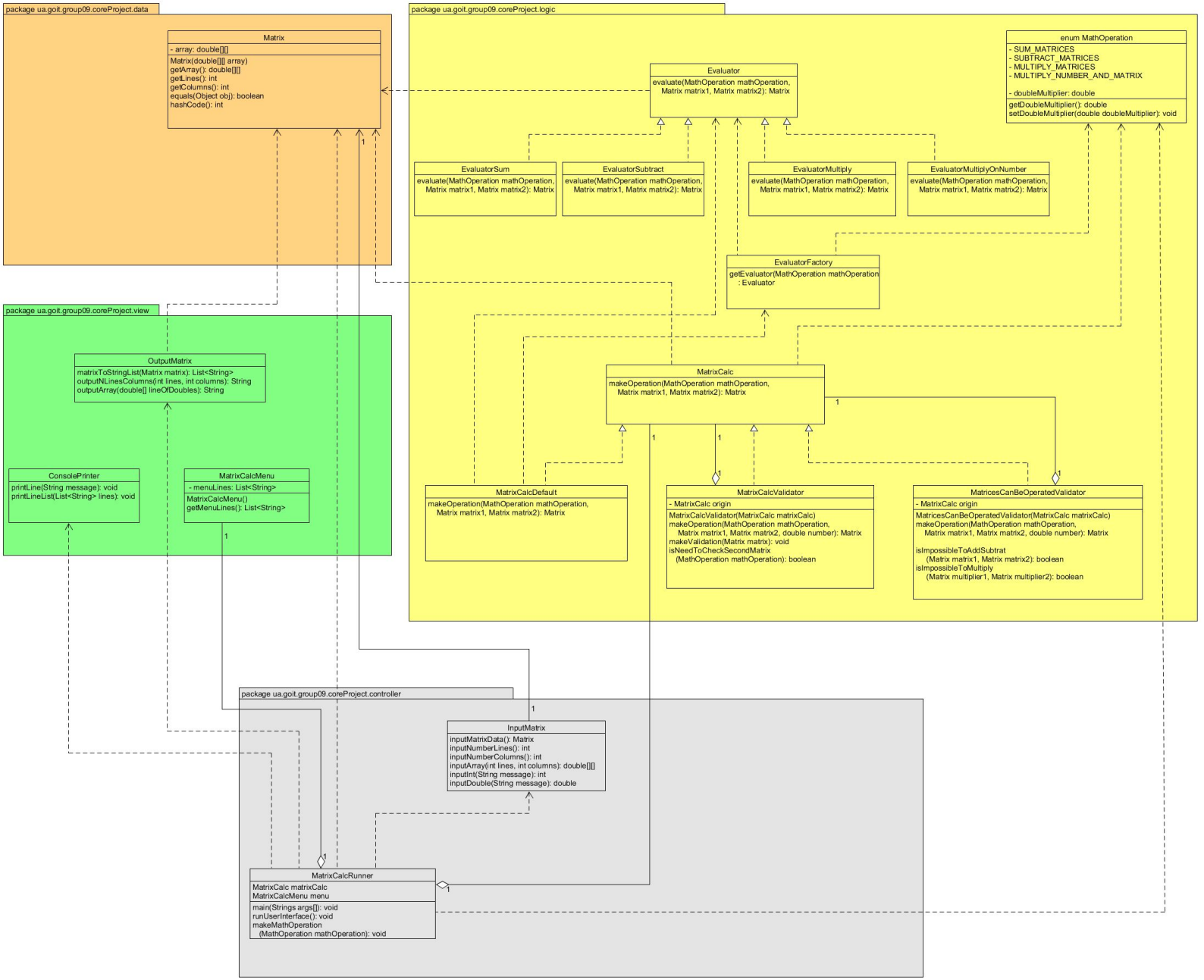
Mentor:

- Artur Dzidzoiev

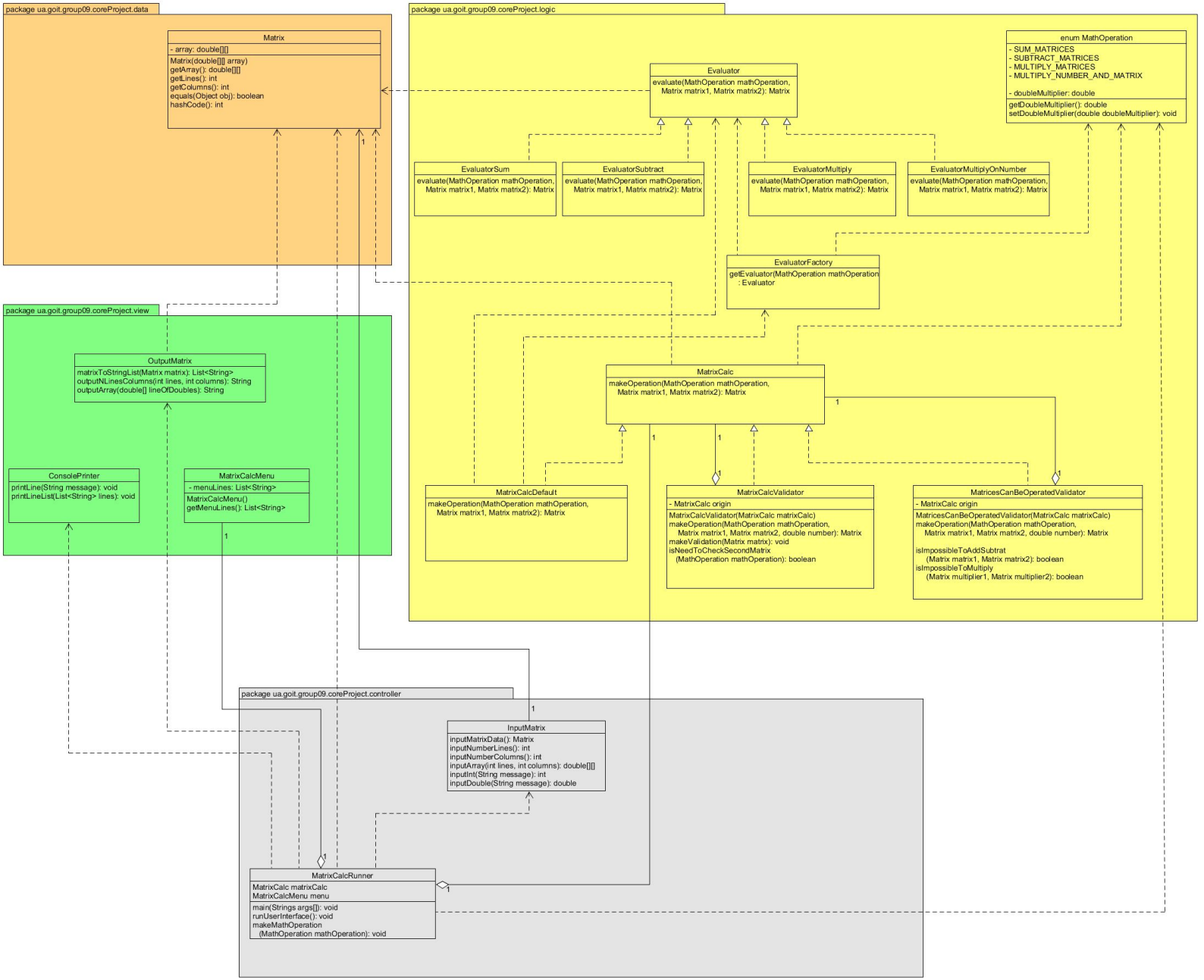
Key decisions

- MVC Pattern
- Immutable class Matrix for data storage
- Validating Decorators
- Factory Method Pattern for math operations
- Algorithms for math operations with matrices
- Custom Exceptions
- Unit tests

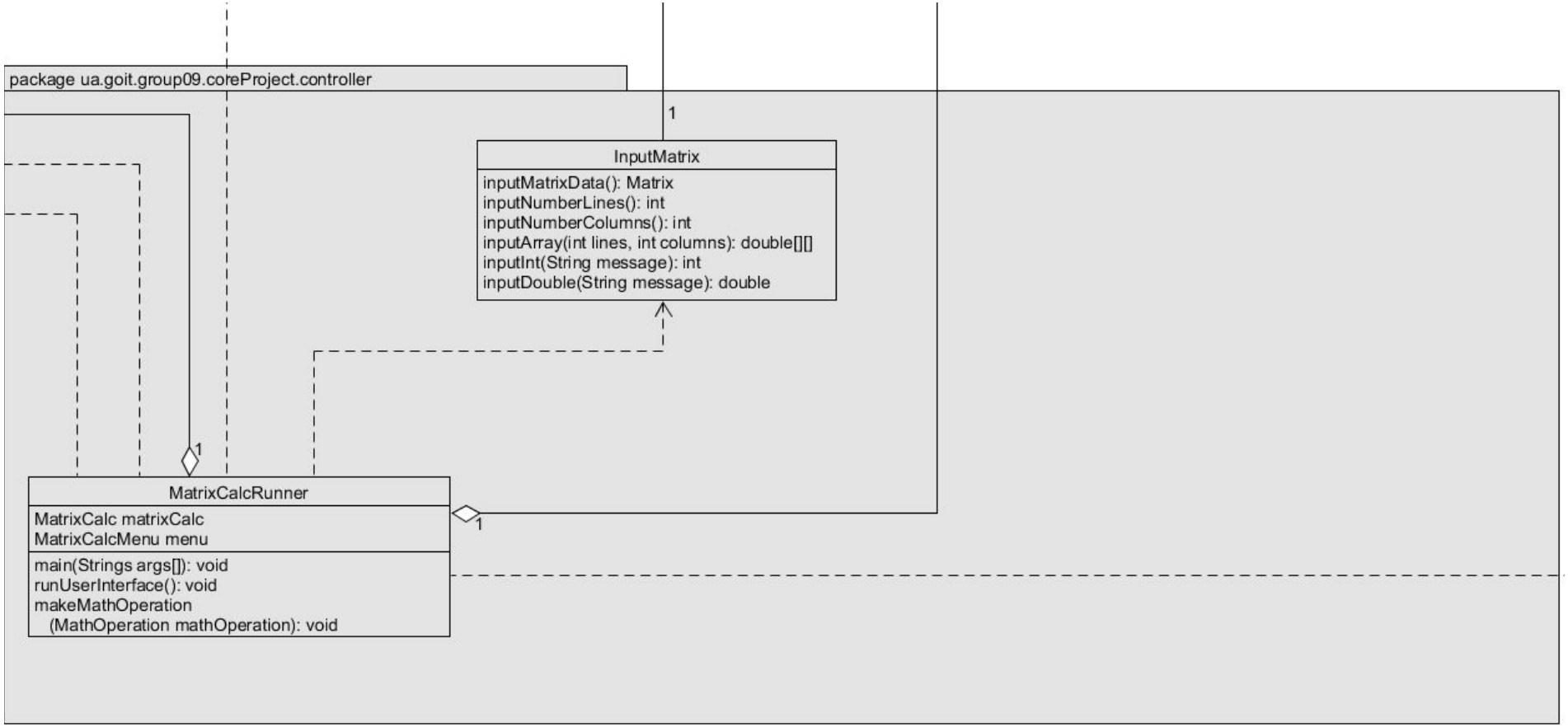
UML Diagram


package ua.goit.group09.coreProject.data
Matrix
- array: double[][]
Matrix(double[][] array)
getArray(): double[][]
getLines(): int
getColumns(): int
equals(Object obj): boolean
hashCode(): int
package ua.goit.group09.coreProject.logic
enum MathOperation
- SUM_MATRICES
- SUBTRACT_MATRICES
- MULTIPLY_MATRICES
- MULTIPLY_NUMBER_AND_MATRIX
- doubleMultiplier: double
getDoubleMultiplier(): double
setDoubleMultiplier(double doubleMultiplier): void
Evaluator
evaluate(MathOperation mathOperation, Matrix matrix1, Matrix matrix2): Matrix
EvaluatorSum
evaluate(MathOperation mathOperation, Matrix matrix1, Matrix matrix2): Matrix
EvaluatorSubtract
evaluate(MathOperation mathOperation, Matrix matrix1, Matrix matrix2): Matrix
EvaluatorMultiply
evaluate(MathOperation mathOperation, Matrix matrix1, Matrix matrix2): Matrix
EvaluatorMultiplyOnNumber
evaluate(MathOperation mathOperation, Matrix matrix1, Matrix matrix2): Matrix
EvaluatorFactory
getEvaluator(MathOperation mathOperation : Evaluator
MatrixCalc
makeOperation(MathOperation mathOperation, Matrix matrix1, Matrix matrix2): Matrix
MatrixCalcDefault
makeOperation(MathOperation mathOperation, Matrix matrix1, Matrix matrix2): Matrix
MatrixCalcValidator
- MatrixCalc origin
MatrixCalcValidator(MatrixCalc matrixCalc)
makeOperation(MathOperation mathOperation, Matrix matrix1, Matrix matrix2, double number): Matrix
makeValidation(Matrix matrix): void
isNeedToCheckSecondMatrix (MathOperation mathOperation): boolean
MatricesCanBeOperatedValidator
- MatrixCalc origin
MatricesCanBeOperatedValidator(MatrixCalc matrixCalc)
makeOperation(MathOperation mathOperation, Matrix matrix1, Matrix matrix2, double number): Matrix
isImpossibleToAddSubtrat (Matrix matrix1, Matrix matrix2): boolean
isImpossibleToMultiply (Matrix multiplier1, Matrix multiplier2): boolean
package ua.goit.group09.coreProject.view
OutputMatrix
matrixToStringList(Matrix matrix): List<String>
outputNLinesColumns(int lines, int columns): String
outputArray(double[] lineOfDoubles): String
ConsolePrinter
printLine(String message): void
printLineList(List<String> lines): void
MatrixCalcMenu
- menuLines: List<String>
MatrixCalcMenu()
getMenuLines(): List<String>
package ua.goit.group09.coreProject.controller
InputMatrix
inputMatrixData(): Matrix
inputNumberLines(): int
inputNumberColumns(): int
inputArray(int lines, int columns): double[][]
inputInt(String message): int
inputDouble(String message): double
MatrixCalcRunner
MatrixCalc matrixCalc
MatrixCalcMenu menu
main(Strings args[]): void
runUserInterface(): void
makeMathOperation (MathOperation mathOperation): void

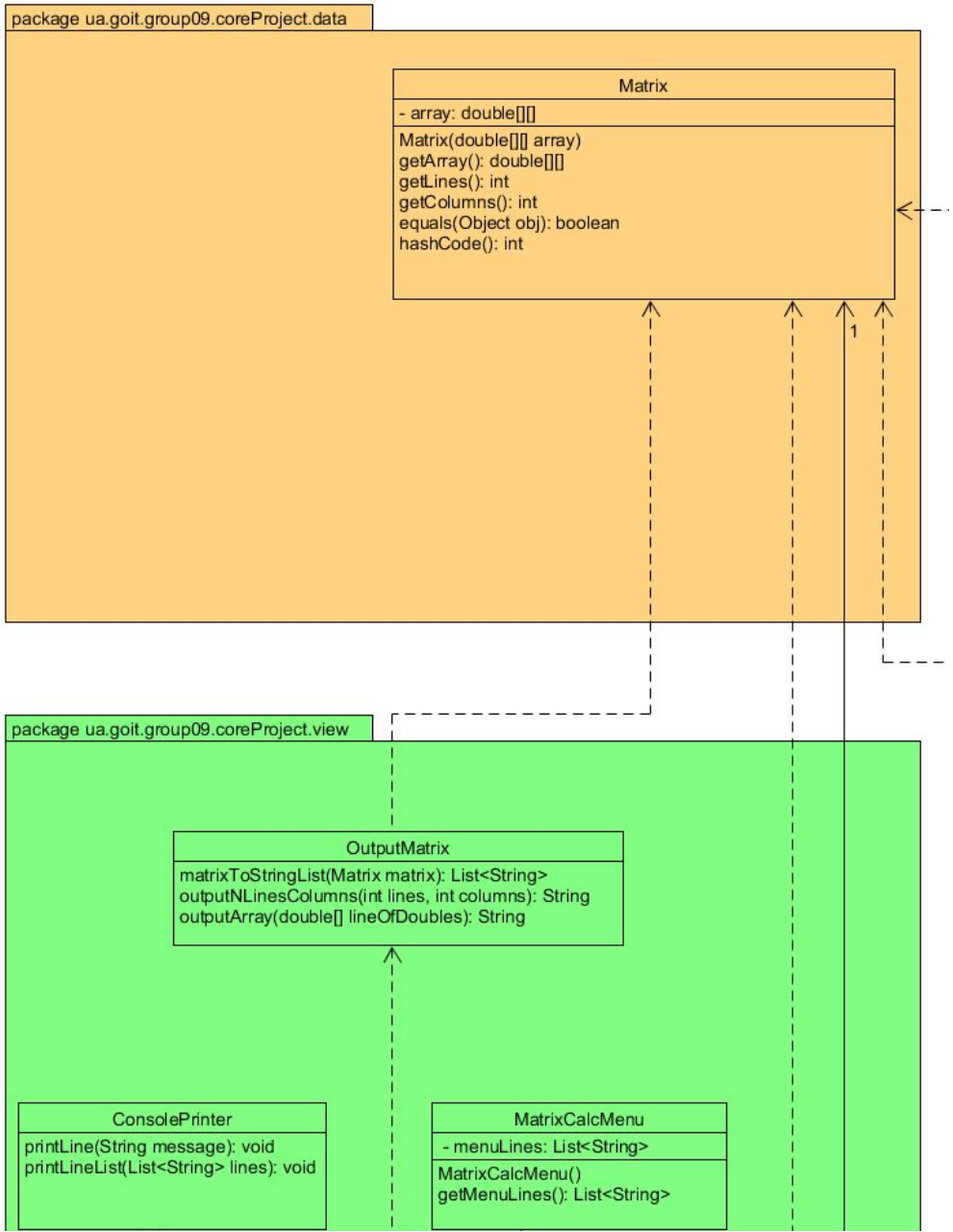
UML Diagram


package ua.goit.group09.coreProject.data
Matrix
- array: double[][]
Matrix(double[][] array)
getArray(): double[][]
getLines(): int
getColumns(): int
equals(Object obj): boolean
hashCode(): int
package ua.goit.group09.coreProject.logic
enum MathOperation
- SUM_MATRICES
- SUBTRACT_MATRICES
- MULTIPLY_MATRICES
- MULTIPLY_NUMBER_AND_MATRIX
- doubleMultiplier: double
getDoubleMultiplier(): double
setDoubleMultiplier(double doubleMultiplier): void
Evaluator
evaluate(MathOperation mathOperation, Matrix matrix1, Matrix matrix2): Matrix
EvaluatorSum
evaluate(MathOperation mathOperation, Matrix matrix1, Matrix matrix2): Matrix
EvaluatorSubtract
evaluate(MathOperation mathOperation, Matrix matrix1, Matrix matrix2): Matrix
EvaluatorMultiply
evaluate(MathOperation mathOperation, Matrix matrix1, Matrix matrix2): Matrix
EvaluatorMultiplyOnNumber
evaluate(MathOperation mathOperation, Matrix matrix1, Matrix matrix2): Matrix
EvaluatorFactory
getEvaluator(MathOperation mathOperation : Evaluator
MatrixCalc
makeOperation(MathOperation mathOperation, Matrix matrix1, Matrix matrix2): Matrix
MatrixCalcDefault
makeOperation(MathOperation mathOperation, Matrix matrix1, Matrix matrix2): Matrix
MatrixCalcValidator
- MatrixCalc origin
MatrixCalcValidator(MatrixCalc matrixCalc)
makeOperation(MathOperation mathOperation, Matrix matrix1, Matrix matrix2, double number): Matrix
makeValidation(Matrix matrix): void
isNeedToCheckSecondMatrix (MathOperation mathOperation): boolean
MatricesCanBeOperatedValidator
- MatrixCalc origin
MatricesCanBeOperatedValidator(MatrixCalc matrixCalc)
makeOperation(MathOperation mathOperation, Matrix matrix1, Matrix matrix2, double number): Matrix
isImpossibleToAddSubtrat (Matrix matrix1, Matrix matrix2): boolean
isImpossibleToMultiply (Matrix multiplier1, Matrix multiplier2): boolean
package ua.goit.group09.coreProject.view
OutputMatrix
matrixToStringList(Matrix matrix): List<String>
outputNLinesColumns(int lines, int columns): String
outputArray(double[] lineOfDoubles): String
ConsolePrinter
printLine(String message): void
printLineList(List<String> lines): void
MatrixCalcMenu
- menuLines: List<String>
MatrixCalcMenu()
getMenuLines(): List<String>
package ua.goit.group09.coreProject.controller
InputMatrix
inputMatrixData(): Matrix
inputNumberLines(): int
inputNumberColumns(): int
inputArray(int lines, int columns): double[][]
inputInt(String message): int
inputDouble(String message): double
MatrixCalcRunner
MatrixCalc matrixCalc
MatrixCalcMenu menu
main(Strings args[]): void
runUserInterface(): void
makeMathOperation (MathOperation mathOperation): void

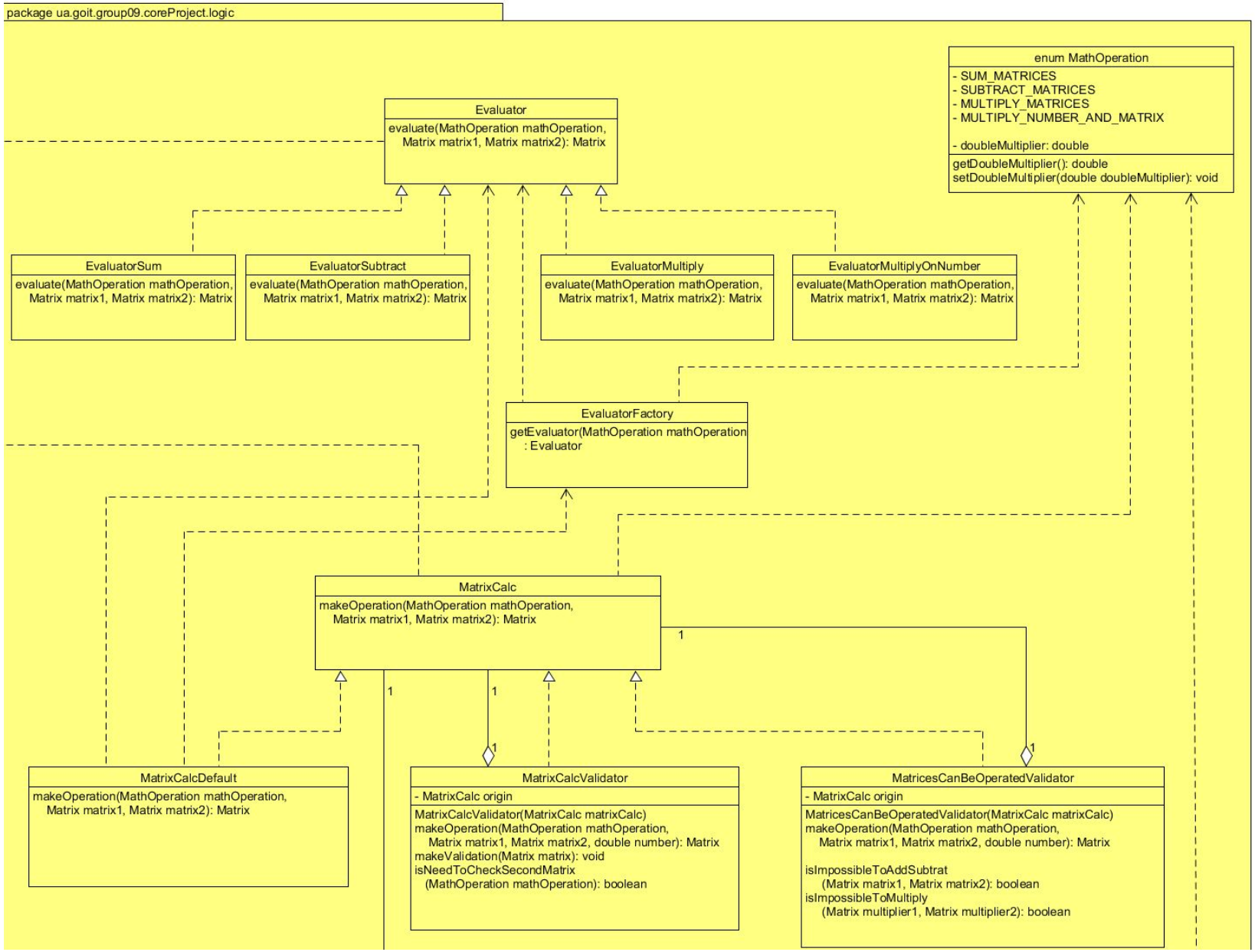
UML Diagram


package ua.goit.group09.coreProject.controller
1
InputMatrix
inputMatrixData(): Matrix
inputNumberLines(): int
inputNumberColumns(): int
inputArray(int lines, int columns): double[][]
inputInt(String message): int
inputDouble(String message): double
1
MatrixCalcRunner
MatrixCalc matrixCalc
MatrixCalcMenu menu
main(Strings args[]): void
runUserInterface(): void
makeMathOperation
(MathOperation mathOperation): void
1

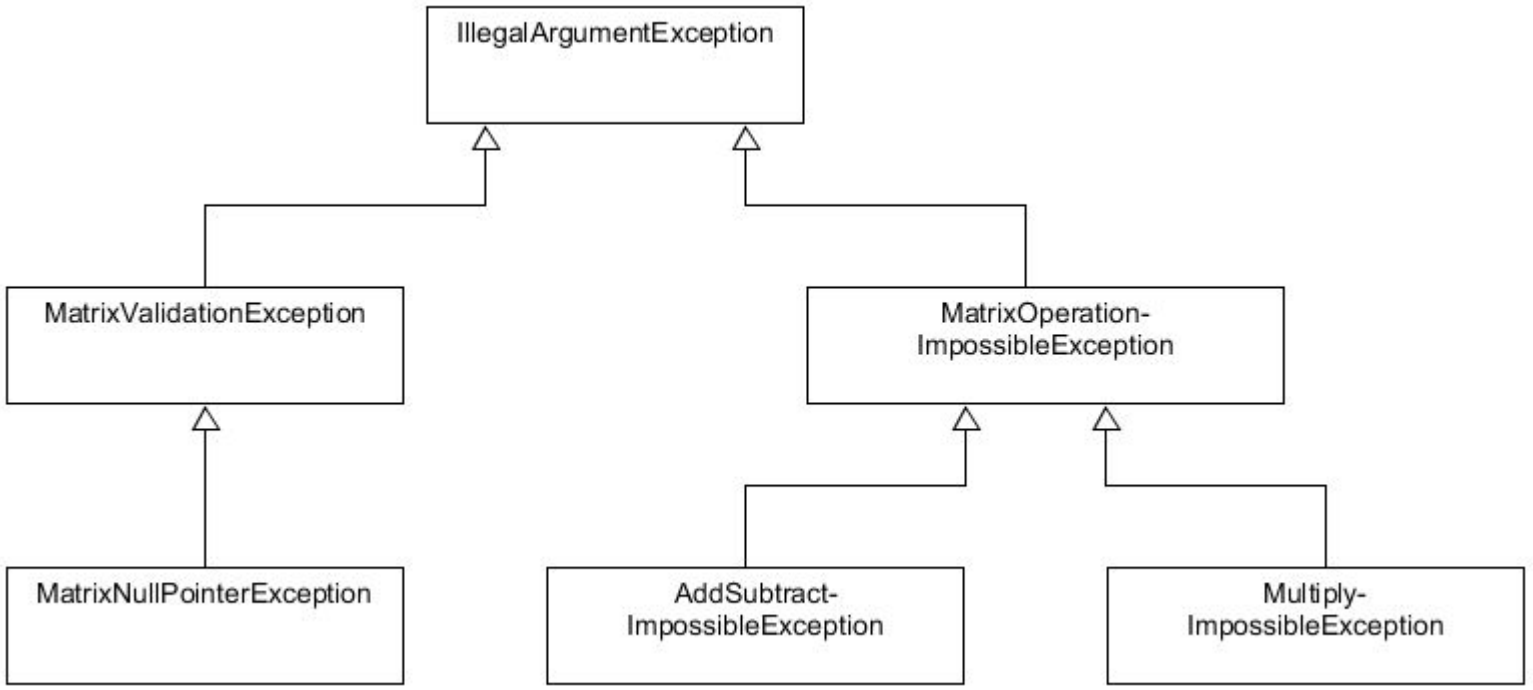
UML Diagram


package ua.goit.group09.coreProject.data
Matrix
- array: double[][]
Matrix(double[][] array)
getArray(): double[][]
getLines(): int
getColumns(): int
equals(Object obj): boolean
hashCode(): int
1
package ua.goit.group09.coreProject.view
OutputMatrix
matrixToStringList(Matrix matrix): List<String>
outputNLinesColumns(int lines, int columns): String
outputArray(double[] lineOfDoubles): String
ConsolePrinter
printLine(String message): void
printLineList(List<String> lines): void
MatrixCalcMenu
- menuLines: List<String>
MatrixCalcMenu()
getMenuLines(): List<String>

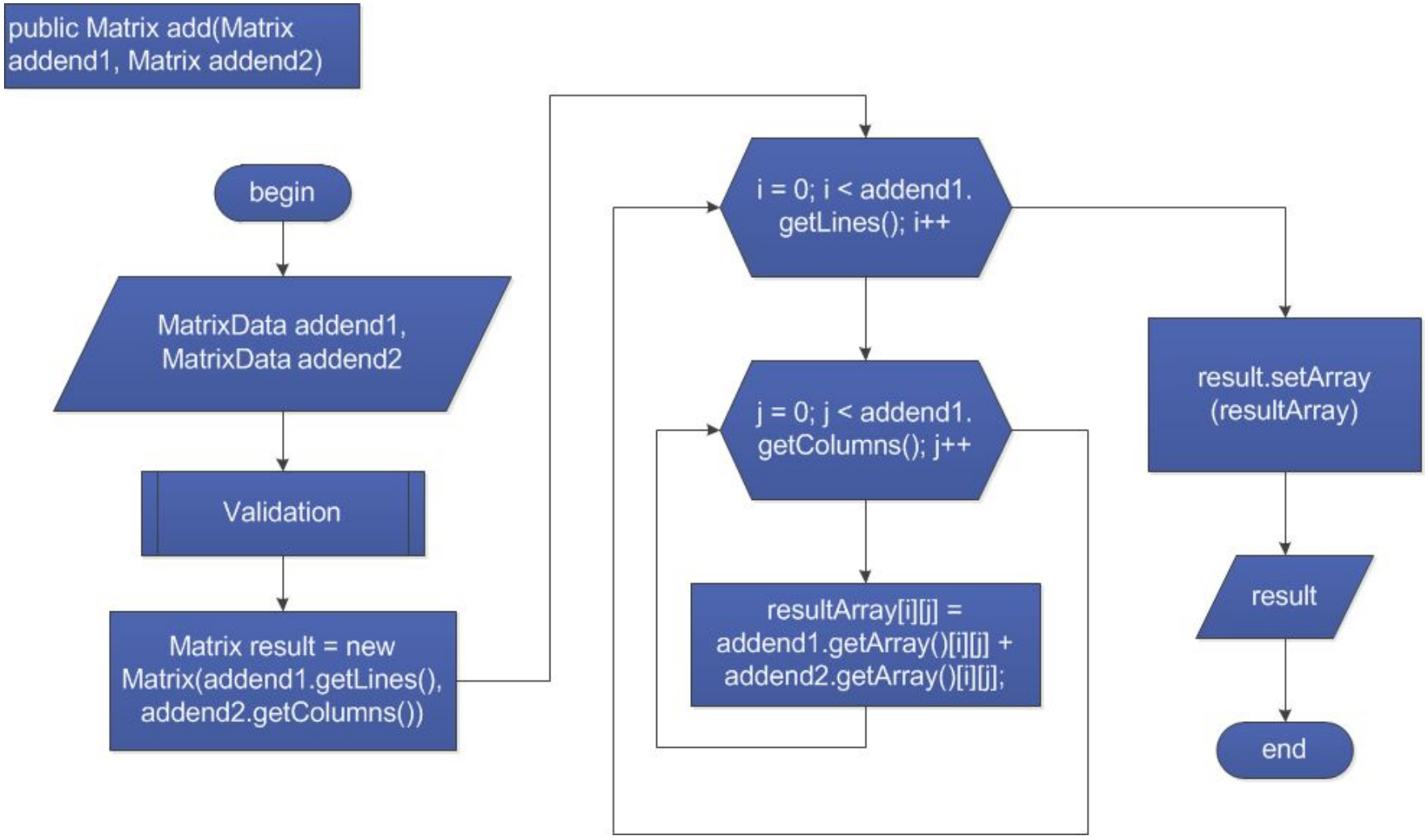
UML Diagram


package ua.goit.group09.coreProject.logic
enum MathOperation
- SUM_MATRICES
- SUBTRACT_MATRICES
- MULTIPLY_MATRICES
- MULTIPLY_NUMBER_AND_MATRIX
- doubleMultiplier: double
getDoubleMultiplier(): double
setDoubleMultiplier(double doubleMultiplier): void
Evaluator
evaluate(MathOperation mathOperation, Matrix matrix1, Matrix matrix2): Matrix
EvaluatorSum
evaluate(MathOperation mathOperation, Matrix matrix1, Matrix matrix2): Matrix
EvaluatorSubtract
evaluate(MathOperation mathOperation, Matrix matrix1, Matrix matrix2): Matrix
EvaluatorMultiply
evaluate(MathOperation mathOperation, Matrix matrix1, Matrix matrix2): Matrix
EvaluatorMultiplyOnNumber
evaluate(MathOperation mathOperation, Matrix matrix1, Matrix matrix2): Matrix
EvaluatorFactory
getEvaluator(MathOperation mathOperation : Evaluator
MatrixCalc
makeOperation(MathOperation mathOperation, Matrix matrix1, Matrix matrix2): Matrix
1
1
1
1
1
MatrixCalcDefault
makeOperation(MathOperation mathOperation, Matrix matrix1, Matrix matrix2): Matrix
MatrixCalcValidator
- MatrixCalc origin
MatrixCalcValidator(MatrixCalc matrixCalc)
makeOperation(MathOperation mathOperation, Matrix matrix1, Matrix matrix2, double number): Matrix
makeValidation(Matrix matrix): void
isNeedToCheckSecondMatrix (MathOperation mathOperation): boolean
MatricesCanBeOperatedValidator
- MatrixCalc origin
MatricesCanBeOperatedValidator(MatrixCalc matrixCalc)
makeOperation(MathOperation mathOperation, Matrix matrix1, Matrix matrix2, double number): Matrix
isImpossibleToAddSubtrat (Matrix matrix1, Matrix matrix2): boolean
isImpossibleToMultiply (Matrix multiplier1, Matrix multiplier2): boolean

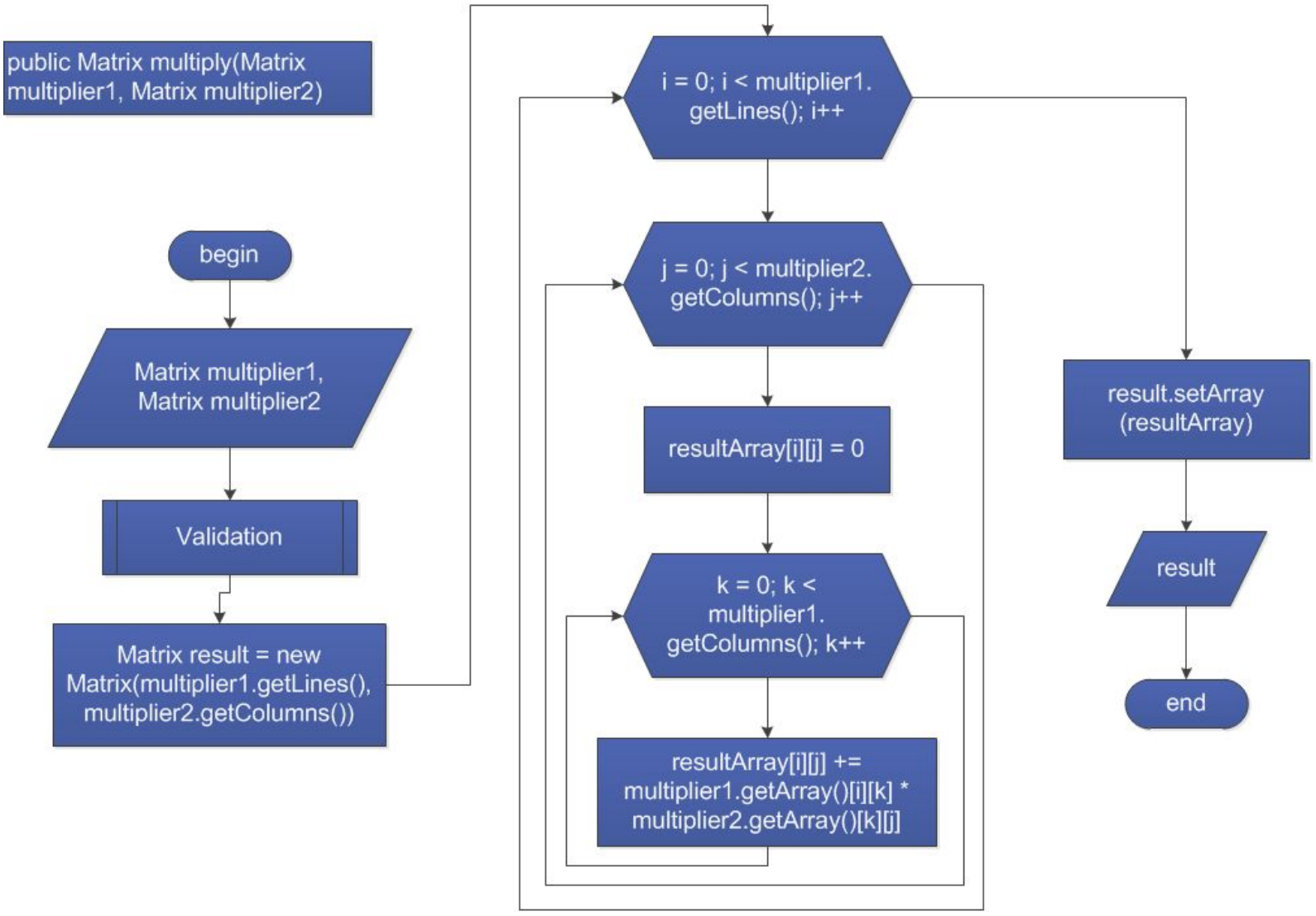
Exceptions UML Diagram



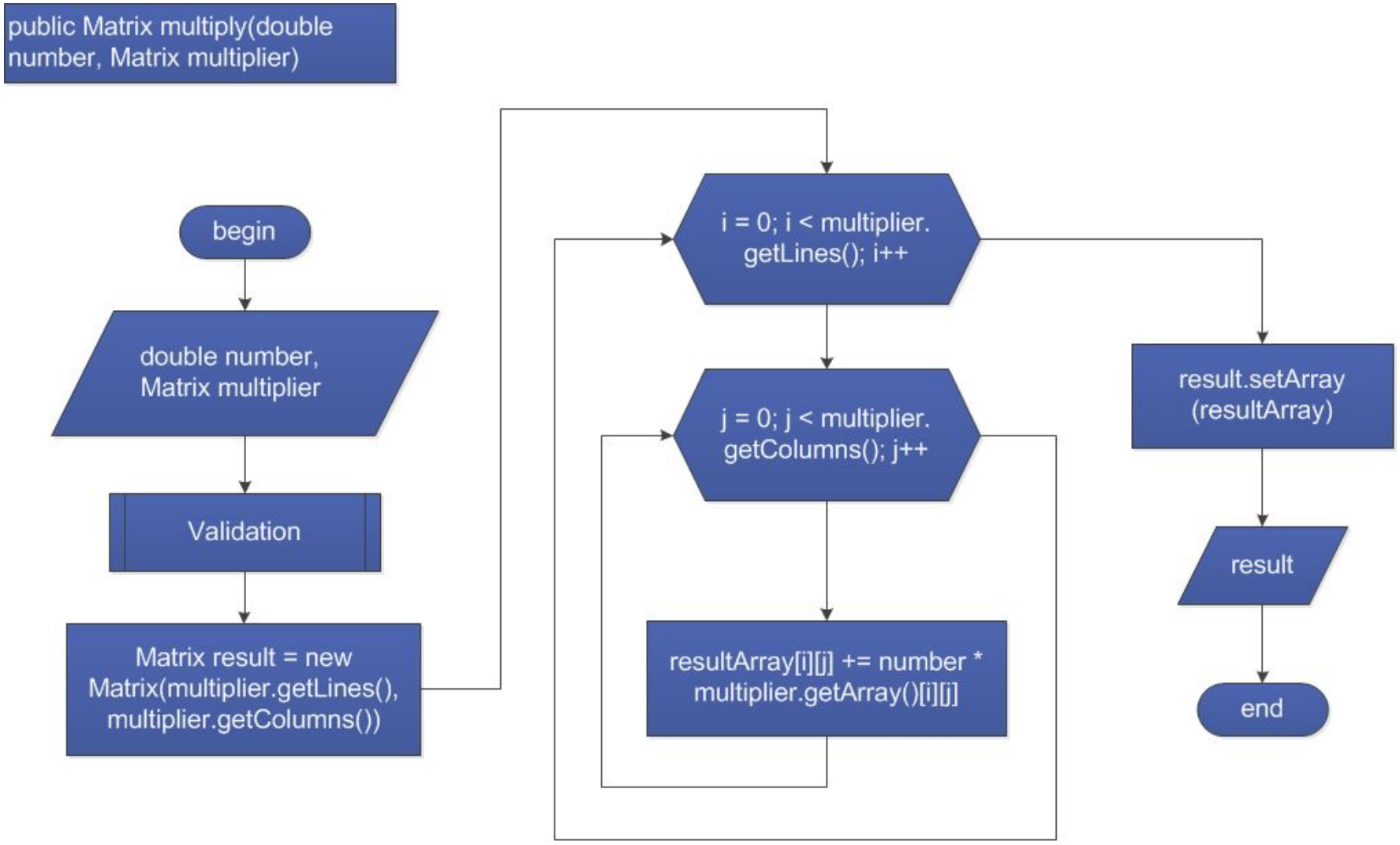
Algorithm of addition matrices



Algorithm of multiplication matrices



Algorithm of multiplying the number and the matrix



Information resources

1. Java 8. The Complete Reference, Ninth Edition. Herbert Schildt.
2. Java. Industrial programming. A Practical Guide. IN Blinov, VS Affair. Minsk 2007
3. Java. Industrial programming. A Practical Guide. IN Blinov, VS Affair. Minsk 2007
4. Java Code Conventions
5. Operations on matrices http://vmatematika.ru/algebra/ linejnaya-algebra/dejstviya-nad-matricami.html
6. Validating Decorators http://www.yegor256.com/2016/01/26/ defensive-programming.html
7. How to Make any Java Class Immutable http://howtodoinjava.com/ core-java/related-concepts/how-to-make-a-java-class-immutable/

Thank you for attention!

Until we meet again!