**The report of the task**

1. **Task definition**

Program is console application that verifies if wall of some configuration can be constructed from some set of bricks.

By the task definition, length of brick should be from 1 to 8, and a height equals 1. A configuration of the wall may be any, but it should be described by a matrix that is composed of «0» and «1». If the wall can be built from a given set of bricks, then program should print to standard output «yes», if the wall cannot be built – «no».

The description of wall and bricks is presented in simple text format and can be read from file.

1. **Description of implementation**

The task was divide into two subtasks. First, it was necessary get from a text file structure of the wall that is necessary to build and set of the bricks of which the wall will be built. Secondly, it was necessary to verify whether it is possible to build a wall of specified configuration from the specified set of bricks.

To solve the first subtask was implemented four methods:

1. getWidthAndHeight () - getting the width and height of the wall
2. getWallStructure () - getting the wall structure
3. getCountOfBricksSorts () - getting the count of different types of bricks
4. getBricks () - getting bricks

The wall structure was save to a one-dimensional array «wallStructure» with elements «0» and «1»

* «1» - part of the wall, which should be filled with bricks;
* «0» - a hole in the wall.

All bricks were save as a map with name «bricks», where:

* key - the length of a brick;
* value - the of bricks with this length.

Directly for solving the main task I implemented a method isPossibleConstructWall(), where:

* from the map «bricks» taken the type of bricks with the largest length.
* in the array, which describes the shape of the wall expended in one "floor", for every brick of this type looking for a place.
* if such place is found - brick is removed from the set of available bricks and elements of wall's array, that match the brick, filled with the value "-1".
* if there is another bricks of the same size in the set, then looking for a place for them in the wall's array
* if there are no bricks of the same size in the set, we take smaller bricks.
* operation of looking for a place is repeated for all the bricks in the set.
* after the all the bricks have been tested for the possibility of their use, the array representing the wall analyzed:
  + if the array contains the value "1" (the cells in the wall that were not filled with bricks), the method returns the string "no" (wall can not be constructed);
  + if the array contains no value "1" (all cells are filled with bricks), the method returns the string "yes" (wall can be constructed);

To develop the program was used **Java SE Development Kit 6u45** and **Intellij IDEA 13.0**

1. **Test instructions**

For the testing of code you have to have simply text data file with the following format:

6 3

101101

111111

111111

3

1 4

2 6

3 1

Line 1: This is width and height of wall matrix - two positive integers W and H separated by space.

Line 2-4: The configuration of wall. «1» - this is unit of wall, that is need to cover. «0» is a hole in the wall.

Line 5: Count of sorts (types) of bricks.

Line 6-8: First digit in line is length of brick (must be from 1 to 8), second digit is number of bricks of the concrete type.

1. **Installation instructions**

To run the program, you should:

1. Open system console and go into the directory with file **Wall.class**
2. Type the command **java Wall data.txt** and press Enter (**data.txt** – text file with a description of structure of the wall and set of the bricks).