# Лабораторная работа 6 Шаблон проектирования «MVC», графический интерфейс пользователя (GUI).

# Вариант №1 (Тетрис).

## Постановка задачи

Написать аналог игры «Тетрис» (“Tetris”). Набор фигур стандартный – все возможные вариации связных многоугольников, составленных из 4-х квадратов. Архитектура программы должна быть основана на паттерне MVC (Mode-View-Controller).

## Требования к программе

1. Игра должна поддерживать таблицу рекордов.
2. Пользователю должны быть доступны команды: *Exit*, *About*, *New Game*, *High Scores*.

# Вариант №2 (Сапёр).

## Постановка задачи

Написать аналог игры «Сапер» (“Minesweeper”) из состава стандартных программ для Windows OS. Архитектура программы должна быть основана на паттерне MVC (Mode-View-Controller). Программа должна иметь два интерфейса – текстовый и графический, причем оба интерфейса должны использовать одну и ту же игровую модель. Т.е. классы данных и логики должны быть одинаковые для текстового и графического интерфейсов.

## Пример структуры программы

/ru/nsu/ccfit/ФАМИЛИЯ/minesweeper – основные классы программы.

/ru/nsu/ccfit/ФАМИЛИЯ/minesweeper/text – классы текстового интерфейса пользователя.

/ru/nsu/ccfit/ФАМИЛИЯ/minesweeper/gui – классы графического интерфейса.

/ru/nsu/ccfit/ФАМИЛИЯ/minesweeper/resources – картинки и другие ресурсы.

## Требования к программе

1. Размер поля и количество мин можно изменить. По умолчанию поле размером 9x9 и количество мин 10.
2. Игра должна поддерживать таблицу рекордов.
3. Пользователю должны быть доступны команды: *Exit*, *About*, *New Game*, *High Scores*.
4. Отчет времени должен быть реализован отдельным потоком.

## Реализация текстового UI

1. Команды пользователя вводятся с консоли, ячейки нумеруются от ноля
2. После каждого хода игрока все игровое поле распечатывается на экран целиком

## Реализация графического UI

1. Мины и флажки отображать с помощью картинок.
2. При формировании окна игры использовать класс **LayoutManager**. Для расположения элементов на игровой панели рекомендуется использовать класс **GridBagLayout**. Для расположения ячеек поля рекомендуется использовать класс **GridLayout**.

# Методические указания:

* Шаблон проектирования “MVC”:
  + <http://rsdn.ru/article/patterns/generic-mvc.xml>
  + <http://ru.wikipedia.org/wiki/Model-View-Controller>
* Для реализации пользовательского интерфейса использовать библиотеку Swing (классы из пакета **javax.swing.\***).
* Работа с компонентами пользовательского интерфейса (классами библиотеки Swing) должна проходить только из UI потока.
* Для отображения диалоговых окон рекомендуется использовать класс **JOptionPane**.