

Курс:  
«Язык программирования Java»

ТЕМА: МЕТОДЫ

### Задание 1

Написать и протестировать методы работы с квадратными матрицами (матрицы представить в виде двумерных массивов).

Должны присутствовать методы:

- создания единичной (диагональной) матрицы;
- создания нулевой матрицы;
- сложение матриц;
- умножения матриц;
- умножение матрицы на скаляр;
- определение детерминанта матрицы;
- вывод матрицы на консоль.

### Задание 2

Написать и протестировать перегруженный метод, выводящий на экран:

- одномерный массив типа *int*;
- одномерный массив типа *String*;
- двумерный массив типа *int*;
- двумерный массив типа *float*.

## Задание 3

В массиве хранится  $n$  явно заданных текстовых строк. Создать метод:

- выводящий содержимое массива в строку через пробел;
- сортирующий массив в обратном порядке (без учета регистра) от  $z$  до  $a$ ;
- сортирующий массив по количеству слов в строке (слова разделены пробелами).

Программа должна вывести строки в начальном и отсортированном порядке.

*Дополнительно:* 1 балл за генерацию случайных уникальных строк реализованных в виде метода.

## Задание 4

На рисунке показан пример треугольника из чисел.



Написать метод:

- выводящий значения треугольника на консоль в таком виде как на рисунке;
- вычисляющий наибольшую сумму чисел, через которые проходит путь, начинающийся на вершине и заканчивающийся где-то на основании.

1. Каждый шаг может идти диагонально вниз-направо или диагонально вниз-налево.
2. Количество строк в треугольнике  $>1$ , но  $<100$ .
3. Числа в треугольнике все целые от 0 до 99 включительно (генерируются случайным образом).

В примере, описанном выше, это путь 7, 3, 8, 7, 5, дающий максимальную сумму 30.

Программа должна выводить на экран треугольник и путь, который даст максимальный результат. Для текущего примера он будет такой – влево, влево, вправо, влево.

## Задание 5

Королю нужно убить дракона, но в его казне мало средств для покупки армии. Нужно создать программу, используя методы, которая поможет **рассчитать минимальное количество копеечников**, которое необходимо, чтобы убить дракона. С клавиатуры вводятся данные:

- здоровья дракона;
- атаки дракона;
- здоровье одного копейщика;
- атака одного копейщика.

Защита, меткость и т. п. не учитывать. Копейщики наносят удар первыми (общий нанесенный урон – это сумма атак всех живых копейщиков). Если атака дракона превышает значение жизни копейщика (например, у копейщика жизни – 10, а атака – 15), то умирает несколько копейщиков, а оставшийся урон идет одному из копейщиков.

Например, жизнь дракона – 500, атака – 55, жизнь одного копейщика – 10, атака – 10, а количество копейщиков при данных условиях – 20.

Лог боя для данного примера должен выглядеть так:

*Итерация 15*

*Копейщики атакуют (урон 200) – у дракона осталось 300 жизней*

*Дракон атакует – осталось 15 копейщиков, один из которых ранен (осталось 5 жизней)*

*Копейщики атакуют – у дракона осталось 150 жизней*

*Дракон атакует – осталось 9 копейщиков*

*Копейщики атакуют – у дракона осталось 60 жизней*

*Дракон атакует – осталось 4 копейщика, один из которых ранен (осталось 5 жизней)*

*Копейщики атакуют – у дракона осталось 20 жизней*

*Дракон атакует и побеждает*

## Задание 6

Напишите метод, проверяющую правильность расстановки скобок в строке, введенной с клавиатуры. При правильной расстановке выполняются условия:

- количество открывающих и закрывающих скобок равно;
- внутри любой пары открывающая–соответствующая закрывающая скобка, скобки расставлены правильно.

В строке могут присутствовать как круглые, так и квадратные скобки (и др. символы). Каждой открывающей скобке соответствует закрывающая того же типа (круглой – круглая, квадратной – квадратная).

Пример неправильной расстановки: ( [ a ) b ]

Пример правильных входных данных: (a[b](f(g)(g))))

Программа должна вывести результат в виде сообщения, примеры:

- Правильная строка
- Ошибка отсутствие (
- Ошибка отсутствие (
- Ошибка отсутствие [
- Ошибка отсутствие ]