

```

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;
import java.util.Random;
/*
All the code comes from me, no cooperation
 */

/**
 * define enum type: +, -, *, and /
 */
enum ops {
    add, sub, mul, div;
}

/**
 * This class generate question list with question and answer
 */
public class ExpGen {
    private final ops op;
    private final int range;
    private final int number;
    private final boolean sign;

    /**
     * construct expression generator with operator, range, total number of question, sign
     */
    public ExpGen(ops op, int range, int number, boolean sign) {
        this.op = op;
        this.range = range;
        this.number = number;
        this.sign = sign;
    }
}

```

```

/**
 * Generate question list
 *
 * @return a map with key as the expression and value as the answer
 */
public Map<String, Integer> getQuestions() {
    Map<String, Integer> questionMap = new HashMap<>();
    for (int i = 0; i < number; i++) {
        String question = "";
        int result = 0;
        Random rand = new Random();
        int operand1 = rand.nextInt(range);
        int operand2 = rand.nextInt(range);
        if (sign) {
            if (rand.nextBoolean()) {
                operand1 *= -1;
            }
            if (rand.nextBoolean()) {
                operand2 *= -1;
            }
        }
        String operand2String = Integer.toString(operand2);
        if (operand2 < 0) {
            operand2String = "(" + operand2String + ")";
        }
        if (op == ops.add) {
            result = operand1 + operand2;
            question = Integer.toString(operand1) + " + " + operand2String;
        } else if (op == ops.sub) {
            result = operand1 - operand2;
            question = Integer.toString(operand1) + " - " + operand2String;
        } else if (op == ops.mul) {

```

```

        result = operand1 * operand2;
        question = Integer.toString(operand1) + " * " + operand2String;
    } else if (op == ops.div) {
        result = operand1 / operand2;
        question = Integer.toString(operand1) + " / " + operand2String;
    }
    questionMap.put(question, result);
}
return questionMap;
}

/**
 * Quick test to see if we generate question and answer correctly
 */
public static void main(String[] args) {
    ExpGen expGen = new ExpGen(ops.add, range: 100, number: 5, sign: true);
    Map<String, Integer> mp = expGen.getQuestions();
    for (Map.Entry<String, Integer> entry : mp.entrySet()) {
        System.out.println(entry.getKey() + " " + entry.getValue());
    }
}
}

```

```

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;
import java.util.Random;
/*
All the code comes from me, no cooperation
 */

/**
 * define enum type: +, -, *, and /
 */
enum ops {
    add, sub, mul, div;
}

/**
 * This class generate question list with question and answer
 */
public class ExpGen {
    private final ops op;
    private final int range;
    private final int number;
    private final boolean sign;

    /**
     * construct expression generator with operator, range, total number of question, sign
     */
    public ExpGen(ops op, int range, int number, boolean sign) {
        this.op = op;
        this.range = range;
        this.number = number;
        this.sign = sign;
    }
}

```

```

/**
 * Generate question list
 *
 * @return a map with key as the expression and value as the answer
 */
public Map<String, Integer> getQuestions() {
    Map<String, Integer> questionMap = new HashMap<>();
    for (int i = 0; i < number; i++) {
        String question = "";
        int result = 0;
        Random rand = new Random();
        int operand1 = rand.nextInt(range);
        int operand2 = rand.nextInt(range);
        if (sign) {
            if (rand.nextBoolean()) {
                operand1 *= -1;
            }
            if (rand.nextBoolean()) {
                operand2 *= -1;
            }
        }
        String operand2String = Integer.toString(operand2);
        if (operand2 < 0) {
            operand2String = "(" + operand2String + ")";
        }
        if (op == ops.add) {
            result = operand1 + operand2;
            question = Integer.toString(operand1) + " + " + operand2String;
        } else if (op == ops.sub) {
            result = operand1 - operand2;
            question = Integer.toString(operand1) + " - " + operand2String;
        } else if (op == ops.mul) {

```

```

        result = operand1 * operand2;
        question = Integer.toString(operand1) + " * " + operand2String;
    } else if (op == ops.div) {
        result = operand1 / operand2;
        question = Integer.toString(operand1) + " / " + operand2String;
    }
    questionMap.put(question, result);
}
return questionMap;
}

/**
 * Quick test to see if we generate question and answer correctly
 */
public static void main(String[] args) {
    ExpGen expGen = new ExpGen(ops.add, range: 100, number: 5, sign: true);
    Map<String, Integer> mp = expGen.getQuestions();
    for (Map.Entry<String, Integer> entry : mp.entrySet()) {
        System.out.println(entry.getKey() + " " + entry.getValue());
    }
}
}

```