Министерство образования и науки, молодежи и спорта Украины

ГВУЗ "Донецкий национальный технический университет"

кафедра Прикладной математики и информатики

Лабораторная работа №1

по курсу " Введение в программирование на Java"

по теме Операторы в Java "

Выполнил студент гр. ІПЗ-12а Егоров А. А.

Проверил: Середа А.А.

Донецк – 2014

Задание на лабораторную работу

Необходимо в основном классе Java - программы создать:

1. 9 методов (один оператор по варианту задания – один метод).

2. 5 методов для проверки приоритетов (методы, в которых используется одновременно два разных оператора – например, «a+b\*c» - результат вычисления зависит от приоритета).

3. 5 методов для проверки ассоциативности (методы, в которых один и тот же оператор используется дважды, и результат зависит от порядка выполнения операторов. Например: «a-b-c» - результат зависит от того, вычисляется ли сначала a-b или b-c).

4. 10 методов тестирования. Для каждого метода тестирования провести минимум 3 теста.

|  |  |
| --- | --- |
| Вариант | Операторы |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | арифметические | | | битовые | | отношения | логические | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 3 | + | % | -= | | | ^= | > | ^ | ! | ^= |

Ручной расчёт результата работы программы для всех тестов.

a + b % c === 10 + 12 % 8 === 14

a -= b + c === 10 -= 12 + 8 === -10

a ^= b | c === 10 ^= 12 | 8 === 6

a + b % c === 11 + 13 % 9 === 15

a -= b + c === 11 -= 13 + 9 === -11

a ^= b | c === 11 ^= 13 | 9 === 6

a + b % c === 12 + 14 % 10 === 16

a -= b + c === 12 -= 14 + 10 === -12

a ^= b | c === 12 ^= 14 | 10 === 2

a + b % c === 13 + 15 % 11 === 17

a -= b + c === 13 -= 15 + 11 === -13

a ^= b | c === 13 ^= 15 | 11 === 2

a + b % c === 14 + 16 % 12 =18

a -= b + c === 14 -= 16 + 12 === -14

a ^= b | c === 14 ^= 16 | 12 === 18

a + b % c === 15 + 17 % 13 === 19

a -= b + c === 15 -= 17 + 13 === -15

a ^= b | c === 15 ^= 17 | 13 === 18

a ^= b ^ c === 1 ^= 0 ^ 0 === true

a ^= !b ^ !c === 1 ^= !0 ^ !0 === true

a ^= b ^ c === 0 ^= 0 ^ 1 === true

a ^= !b ^ !c === 0 ^= !0 ^ !1 === true

a ^= b ^ c === 1 ^= 1 ^ 0 === false

a ^= !b ^ !c === 1 ^= !1 ^ !0 === false

a ^= b ^ c === 1 ^= 1 ^ 1 === true

a ^= !b ^ !c === 1 ^= !1 ^ !1 === true

a ^= b ^ c === 0 ^= 0 ^ 0 === false

a ^= !b ^ !c === 0 ^= !0 ^ !0 === false

a % b % c === 10 % 12 % 5 === 0

a -= b -= c === 10 -= 12 -= 5 === 3

a ^= b ^= c === 10 ^= 12 ^= 5 === 3

a % b % c === 11 % 13 % 6 === 5

a -= b -= c === 11 -= 13 -= 6 === 4

a ^= b ^= c === 11 ^= 13 ^= 6 === 0

a % b % c === 12 % 14 % 7 === 5

a -= b -= c === 12 -= 14 -= 7 === 5

a ^= b ^= c === 12 ^= 14 ^= 7 === 5

a % b % c === 13 % 15 % 8 === 5

a -= b -= c === 13 -= 15 -= 8 === 6

a ^= b ^= c === 13 ^= 15 ^= 8 === 10

a % b % c === 14 % 16 % 9 === 5

a -= b -= c === 14 -= 16 -= 9 === 7

a ^= b ^= c === 14 ^= 16 ^= 9 === 23

a % b % c === 15 % 17 % 10 === 5

a -= b -= c === 15 -= 17 -= 10 === 8

a ^= b ^= c === 15 ^= 17 ^= 10 === 20

a ^= b ^ c === 1 ^= 0 ^ 0 === true

a ^= b ^= c === 1 ^= 0 ^= 0 === true

a ^= b ^ c === 0 ^= 0 ^ 1 === true

a ^= b ^= c === 0 ^= 0 ^= 1 === true

a ^= b ^ c === 1 ^= 1 ^ 0 === false

a ^= b ^= c === 1 ^= 1 ^= 0 === false

a ^= b ^ c === 1 ^= 1 ^ 1 true

a ^= b ^= c === 1 ^= 1 ^= 1 === true

a ^= b ^ c === 0 ^= 0 ^ 0 === false

a ^= b ^= c === 0 ^= 0 ^= 0 === false

5+8 === 13

5%2 === 1

5-=8 === -3

5|8 === 13

5^=8 === 13

5 > 8 === false

6+9 === 15

6%3 === 0

6-=9 === -3

6|9 === 15

6 ^=9 === 15

6 > 9 === false

7 + 10 === 17

7 % 4 === 3

7-=10 === -3

7|10 === 15

7 ^=10 === 13

7 > 10 === false

8+11 === 19

8%5 === 3

8-=11 === -3

8|11 === 11

8^=11 === 3

8>11 === false

1 ^ 1 === false

1 ^=1 === false

1 ^=1 === false

1 ^ 0 === true

1 ^= 0 === true

1 ^= 0 === true

0 ^ 1 === true

0 ^= 1 === true

0 ^= 1 === true

0 ^ 0 === false

0 ^= 0 === false

0 ^= 0 === false

1 === false

0 === true

Распечатка программы

package lab.yegorov;

public class Main {

public static void main(String args[]) {

for(int a = 10; a < 16; ++a) {

priorityTest1(a, a + 2, a - 2);

priorityTest2(a, a + 2, a - 2);

priorityTest3(a, a + 2, a - 2);

}

priorityTest4(true, false, false);

priorityTest5(true, false, false);

priorityTest4(false, false, true);

priorityTest5(false, false, true);

priorityTest4(true, true, false);

priorityTest5(true, true, false);

priorityTest4(true, true, true);

priorityTest5(true, true, true);

priorityTest4(false, false, false);

priorityTest5(false, false, false);

for(int a = 10; a < 16; ++a) {

associativityTest1(a, a + 2, a - 5);

associativityTest2(a, a + 2, a - 5);

associativityTest3(a, a + 2, a - 5);

}

associativityTest4(true, false, false);

associativityTest5(true, false, false);

associativityTest4(false, false, true);

associativityTest5(false, false, true);

associativityTest4(true, true, false);

associativityTest5(true, true, false);

associativityTest4(true, true, true);

associativityTest5(true, true, true);

associativityTest4(false, false, false);

associativityTest5(false, false, false);

for(int a = 5; a < 9; ++a) {

addTest(a, a+3);

modTest(a, a-3);

assignmentWithSubtractionTest(a, a+3);

orTest(a, a+3);

xorWithAssignmentTest(a, a+3);

moreTest(a, a+3);

}

boolXorTest(true, true);

boolXorWithAssignmentTest(true, true);

boolXorWithAssignmentTest2(true, true);

boolXorTest(true, false);

boolXorWithAssignmentTest(true, false);

boolXorWithAssignmentTest2(true, false);

boolXorTest(false, true);

boolXorWithAssignmentTest(false, true);

boolXorWithAssignmentTest2(false, true);

boolXorTest(false, false);

boolXorWithAssignmentTest(false, false);

boolXorWithAssignmentTest2(false, false);

boolNotTest(true);

boolNotTest(false);

}

public static int add(int a, int b) {

return a + b;

}

public static int mod(int a, int b) {

return a % b;

}

public static int assignmentWithSubtraction(int a, int b) {

return a -= b;

}

public static int or(int a, int b) {

return a | b;

}

public static int xorWithAssignment(int a, int b) {

return a ^= b;

}

public static boolean more(int a, int b) {

return a > b;

}

public static boolean boolXor(boolean a, boolean b) {

return a ^ b;

}

public static boolean boolNot(boolean a) {

return !a;

}

public static boolean boolXorWithAssignment(boolean a, boolean b) {

return a ^= b;

}

//-------//

public static void priorityTest1(int a, int b, int c) {

int t;

if((t = add(a,mod(b,c))) == (a + b % c))

System.out.println("Priority operators \"a + b % c\" "+ t +" is work!");

else

System.out.println("Priority operators \"a + b % c\" not working correctly!");

}

public static void priorityTest2(int a, int b, int c) {

int t;

if((t = assignmentWithSubtraction(a, add(b, c))) == (a -= b + c))

System.out.println("Priority operators \"a -= b + c\" "+ t +" is work!");

else

System.out.println("Priority operators \"a -= b + c\" not working correctly!");

}

public static void priorityTest3(int a, int b, int c) {

int t;

if((t = xorWithAssignment(a, or(b, c))) == (a ^= b | c))

System.out.println("Priority operators \"a ^= b | c\" "+ t +" is work!");

else

System.out.println("Priority operators \"a ^= b | c\" not working correctly!");

}

public static void priorityTest4(boolean a, boolean b, boolean c) {

boolean t;

if((t = boolXorWithAssignment(a, boolXor(b, c))) == (a ^= b ^ c))

System.out.println("Priority operators \"a ^= b ^ c\" "+ t +" is work!");

else

System.out.println("Priority operators \"a ^= b ^ c\" not working correctly!");

}

public static void priorityTest5(boolean a, boolean b, boolean c) {

boolean t;

if((t = boolXorWithAssignment(a, boolXor(boolNot(b), boolNot(c)))) == (a ^= !b ^ !c))

System.out.println("Priority operators \"a ^= !b ^ !c\" "+ t +" is work!");

else

System.out.println("Priority operators \"a ^= !b ^ !c\" not working correctly!");

}

//-----//

public static void associativityTest1(int a, int b, int c) {

int t;

if((t = mod(mod(a, b), c)) == (a % b % c))

System.out.println("Associativity operators \"a % b % c\" "+ t +" is work!");

else

System.out.println("Associativity operators \"a % b % c\" not working correctly!");

}

public static void associativityTest2(int a, int b, int c) {

int t;

if((t = assignmentWithSubtraction(a, assignmentWithSubtraction(b, c))) == (a -= b -= c))

System.out.println("Associativity operators \"a -= b -= c\" "+ t +" is work!");

else

System.out.println("Associativity operators \"a -= b -= c\" not working correctly!");

}

public static void associativityTest3(int a, int b, int c) {

int t;

if((t = xorWithAssignment(a, xorWithAssignment(b, c))) == (a ^= b ^= c))

System.out.println("Associativity operators \"a ^= b ^= c\" "+ t +" is work!");

else

System.out.println("Associativity operators \"a ^= b ^= c\" not working correctly!");

}

public static void associativityTest4(boolean a, boolean b, boolean c) {

boolean t;

if((t = boolXor(boolXor(a, b), c)) == (a ^ b ^ c))

System.out.println("Associativity operators \"a ^= b ^ c\" "+ t +" is work!");

else

System.out.println("Associativity operators \"a ^= b ^ c\" not working correctly!");

}

public static void associativityTest5(boolean a, boolean b, boolean c) {

boolean t;

if((t = boolXorWithAssignment(a, boolXorWithAssignment(b, c))) == (a ^= b ^= c))

System.out.println("Associativity operators \"a ^= b ^= c\" "+ t +" is work!");

else

System.out.println("Associativity operators \"a ^= b ^= c\" not working correctly!");

}

//-----//

public static void addTest(int a, int b) {

int t;

if((t = add(a, b)) == (a + b))

System.out.println("Method \"add\" "+ t +" is work!");

else

System.out.println("Method \"add\" not working correctly!");

}

public static void modTest(int a, int b) {

int t;

if((t = mod(a, b)) == (a % b))

System.out.println("Method \"mod\" "+ t +" is work!");

else

System.out.println("Method \"mod\" not working correctly!");

}

public static void assignmentWithSubtractionTest(int a, int b) {

int t;

if((t = assignmentWithSubtraction(a,b)) == (a -= b))

System.out.println("Method \"assignmentWithSubtraction\" "+ t +" is work!");

else

System.out.println("Method \"assignmentWithSubtraction\" "+ t +" is work!");

}

public static void orTest(int a, int b) {

int t;

if((t = or(a, b)) == (a | b))

System.out.println("Method \"or\" "+ t +" is work!");

else

System.out.println("Method \"or\" not working correctly!");

}

public static void xorWithAssignmentTest(int a, int b) {

int t;

if((t = xorWithAssignment(a, b)) == (a ^= b))

System.out.println("Method \"xorWithAssignment\" "+ t +" is work!");

else

System.out.println("Method \"xorWithAssignment\" not working correctly!");

}

public static void moreTest(int a, int b) {

boolean t;

if((t = more(a, b)) == (a > b))

System.out.println("Method \"more\" "+ t +" is work!");

else

System.out.println("Method \"more\" not working correctly!");

}

public static void boolXorTest(boolean a, boolean b) {

boolean t;

if((t = boolXor(a, b)) == (a ^ b))

System.out.println("Method \"boolXor\" "+ t +" is work!");

else

System.out.println("Method \"boolXor\" not working correctly!");

}

public static void boolNotTest(boolean a) {

boolean t;

if((t = boolNot(a)) == (!a))

System.out.println("Method \"boolNot\" "+ t +" is work!");

else

System.out.println("Method \"boolNot\" not working correctly!");

}

public static void boolXorWithAssignmentTest(boolean a, boolean b) {

boolean t;

if((t = boolXorWithAssignment(a, b)) == (a ^= b))

System.out.println("Method \"boolXorWithAssignment\" "+ t +" is work!");

else

System.out.println("Method \"boolXorWithAssignment\" not working correctly!");

}

public static void boolXorWithAssignmentTest2(boolean a, boolean b) {

boolean t;

if((t = boolXorWithAssignment(b, a)) == (b ^= a))

System.out.println("Method \"boolXorWithAssignment\" "+ t +" is work!");

else

System.out.println("Method \"boolXorWithAssignment\" not working correctly!");

}

}

Экранные формы

