САНКТ-ПЕТЕРБУРЖСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

*Кафедра Вычислительной техники*

**Лабораторная работа №2**

*«Принципы ООП»*

Выполнил:

студент II курса группы 2125

Припадчев Артём

Проверил:

Харитонова А.Е.

Санкт-Петербург

2013

**Задание:** скомпилировать и запустить исходный код на языке программирования Java, выданный в соответствии с вариантом. Разобраться том, как реализуются принципы объектно-ориентированного программирования в получившейся программе, и том, почему она выдаёт такой результат. Добавить комментарии в ключевые фрагменты программы, поясняющие её поведение.

# Код программы

// var. 250

**public** **class** Lab2 {

**public** **static** **void** main(String[] args) {

C a = **new** C();

C b = **new** E();

E c = **new** E();

b.m12(); //вызывается метод класса E

a.m6(); //вызывается метод класса C

b.m10(); //вызывается метод класса E

c.m24(); //вызывается метод класса E

c.m1(); //вызывается метод класса E

a.m40();//вызывается метод класса C

a.*m14*();//вызывается метод класса C

b.*m28*();//вызывается метод класса C

a.*m27*();//вызывается метод класса C

c.*m11*();//вызывается метод класса E

a.m20(a); //вызывается метод класса C

b.m20(b); //вызывается метод класса E

a.m20(c); //вызывается метод класса E

}

}

**class** C {

**int** m22;

**int** m9;

**int** m13;

**int** m34;

**int** m18;

**long** m19;

**long** m15;

**long** m32;

**int**[] m29 = {-3, 1, 1, 2, -2};

**int**[] m39 = {-2, -2, -2, 1, 3};

**int**[] m7 = {-1, -1, -2, -2, -2};

**static** **int** *m2*;

**static** **int** *m36*;

**static** **int** *m4*;

**static** **int** *m21*;

**static** **int** *m30*;

**public** C() {

m22 = 0;

m9 = 1;

m13 = 8;

m34 = 4;

m18 = 3;

m19 = 8L; //L – суффикс, обозначающий long

m15 = 4L;

m32 = 9L;

}

**public** **void** m12() {

System.*out*.println("метод m12 в классе C");

System.*out*.println(m9 - 1);

}

**public** **void** m6() {

System.*out*.println("метод m6 в классе C");

System.*out*.println(++*m21*); //префиксный инкремент

}

**public** **void** m10() {

System.*out*.println("метод m10 в классе C");

System.*out*.println(*m4*++); //постфиксный инкремент

}

**public** **void** m24() {

System.*out*.println("метод m24 в классе C");

System.*out*.println(++m22);

}

**public** **void** m1() {

System.*out*.println("метод m1 в классе C");

System.*out*.println(m22 + 3);

}

**public** **void** m40() {

System.*out*.println("метод m40 в классе C");

System.*out*.println(m9 << 2); //сдвиг влево на 2 разряда, эквивалентно умножению на 4

}

**public** **static** **void** m14() {

System.*out*.println("метод m14 в классе C");

System.*out*.println(*m21*);

}

**public** **static** **void** m28() {

System.*out*.println("метод m28 в классе C");

System.*out*.println((*m21* - 4));

}

**public** **static** **void** m27() {

System.*out*.println("метод m27 в классе C");

System.*out*.println(*m30*);

}

**public** **static** **void** m11() {

System.*out*.println("метод m11 в классе C");

System.*out*.println((*m30* - 4));

}

**public** **void** m20(C r) {

r.m12();

}

**public** **void** m20(E r) {

r.m6();

}

}

**class** E **extends** C { //Ключевое слово extends указывает, что E наследуется от C

**public** E() { //конструктор класса E

m22 = 3;

m9 = 1;

m34 = 3;

m18 = 6;

m19 = 8L; //L – суффикс, обозначающий long

m15 = 8L;

}

**public** **void** m12() {

System.*out*.println("метод m12 в классе E");

System.*out*.println(m18 << 2);

}

**public** **void** m6() {

System.*out*.println("метод m6 в классе E");

System.*out*.println(m13);

}

**public** **void** m10() {

System.*out*.println("метод m10 в классе E");

System.*out*.println(m18);

}

**public** **void** m24() {

System.*out*.println("метод m24 в классе E");

System.*out*.println(m34);

}

**public** **void** m1() {

System.*out*.println("метод m1 в классе E");

System.*out*.println(m9--); //постфиксный декремент

}

**public** **static** **void** m14() {

System.*out*.println("метод m14 в классе E");

System.*out*.println(*m36*);

}

**public** **static** **void** m28() {

System.*out*.println("метод m28 в классе E");

System.*out*.println((*m36* - 4));

}

**public** **static** **void** m27() {

System.*out*.println("метод m27 в классе E");

System.*out*.println(*m36*);

}

**public** **static** **void** m11() {

System.*out*.println("метод m11 в классе E");

System.*out*.println((*m36* - 4));

}

**public** **void** m20(C r) {

r.m10();

}

**public** **void** m20(E r) {

r.m24();

}

}

**Результат программы:**

метод m12 в классе E

24

метод m6 в классе C

1

метод m10 в классе E

6

метод m24 в классе E

3

метод m1 в классе E

1

метод m40 в классе C

4

метод m14 в классе C

1

метод m28 в классе C

-3

метод m27 в классе C

0

метод m11 в классе E

-4

метод m12 в классе C

0

метод m10 в классе E

6

метод m6 в классе E

8

**Вывод:** разобрались с принципом работы, данной в соответствии с вариантом, программы, а также изучили основные принципы ООП