Университет ИТМО.

Лабораторная работа №2.

Работу выполнил студент группы 2125
Халанский Дмитрий Владимирович

2014

1. Листинг программы

```java
// 0

public class Lab2 {
  public static void main(String[] args) {

    // Variable initialization
    Crustle daughter = new Crustle();
    Dwebble sister = new Crustle();
    Dwebble mother = new Dwebble();

    // Family battle
    mother.sand_Attack(daughter);
    daughter.sand_Attack(mother);
    daughter.defenseCurl();
    ((Crustle)sister).splash();
    mother.foresight();
    daughter.camouflage();
    mother.swarm();
    sister.focusEnergy();
    daughter.swarm();
    sister.foresight();
    daughter.foresight();
    daughter.sand_Attack(sister);
    daughter.swordsDance();
    mother.sharpen();
    sister.swarm();
    mother.sand_Attack(sister);
  }
}

class Dwebble {
  protected byte underwater;
  protected String iceGhost = "IceGhost";
  int inflatable = 84;
  protected String ghost = "Ghost";
  public static int blender;
  protected String ice = "Ice";
  double weight = 9.4;

  public Dwebble() {
    // 0x84 is 10000100 in binary. Its complementary code is ~(10000100)+1,
    // which is 0111 1100 in binary and 124 in decimal. It means that
    // underwater is -124.
    underwater = (byte) 0x84;
  }

  static {
    blender = 84;
  }
```

```
50
51    public void focusEnergy() {
52      // True
53      // Java compiler replaces "Ice" + "Ghost" with "IceGhost", to which
54      // iceGhost is a reference, too.
55      System.out.println(iceGhost == "Ice"+"Ghost");
56      // True
57      // Here a new String object is built, then it is compared to iceGhost
58      // byte-by-byte using String.equals.
59      System.out.println(iceGhost.equals(ice+ghost));
60      // True
61      // Same as before, only here the string literal "IceGhost" and iceGhost
62      // reference the same memory area, so String.equals doesn't need to
63      // check the equality of strings, only of references.
64      System.out.println(iceGhost.equals("Ice"+"Ghost"));
65      // False
66      // A new String object is built, and a reference to it is not the same
67      // as to iceGhost.
68      System.out.println(iceGhost == ice+ghost);
69      // False
70      // See above.
71      System.out.println(iceGhost == "Ice"+ghost);
72      // True
73      // A new String object is built, then String.equals checks their factual
74      // equality.
75      System.out.println(iceGhost.equals(ice+"Ghost"));
76    }
77
78    public void sharpen() {
79      double accuracy = 3.7;
80
81      // False
82      // We cannot compare two floating-point numbers accurately without
83      // using the machine delta.
84      System.out.println((weight + accuracy) == 5.7);
85    }
86
87    public void sand_Attack(Dwebble p) {
88      System.out.println("Dwebble attacks Dwebble with Sand- Attack");
89    }
90
91    public void sand_Attack(Crustle p) {
92      System.out.println("Dwebble attacks Crustle with Sand- Attack");
93    }
94
95    public static void swarm() {
96      System.out.println("Dwebble casts Swarm");
97    }
98
99    public void swordsDance() {
100     // 208
```

```
101       // blender is 84
102       // underwater is -124
103       System.out.println(blender - underwater);
104       // -40
105       // underwater is -124
106       // inflatable is 84
107       System.out.println(underwater + inflatable);
108       // 0
109       // inflatable is 84
110       // blender is 84
111       System.out.println(inflatable - blender);
112     }
113
114     public void foresight() {
115       System.out.println("Dwebble casts Foresight");
116     }
117 }
118
119 class Crustle extends Dwebble {
120     private String electric = "Electric";
121     private String electricGhost = "ElectricGhost";
122     float length = 3.6f;
123     private int intelligence;
124
125     public Crustle() {
126       // It is actually 61, only written in octal
127       intelligence = 075;
128     }
129
130     // Instance initializer. Its purpose is unclear since the constructor is
131     // called after it, replacing the values with the other ones
132     {
133       intelligence = 42;
134     }
135
136
137     public void camouflage() {
138       float speed = 4.2f;
139
140       // Since two floating-point variables cannot be compared directly and
141       // need a delta, prints ``false''
142       System.out.println((speed + length) == 7.8f);
143     }
144
145     public void defenseCurl() {
146       // -23
147       // intelligence is 61
148       // inflatible is 84
149       System.out.println(intelligence - inflatable);
150       // 145
151       // intelligence is 61
```

```
152     // blender is 84
153     System.out.println(intelligence + blender);
154     // -185
155     // underwater is -124
156     // intelligence is 61
157     System.out.println(underwater - intelligence);
158   }
159
160   public void foresight() {
161     System.out.println("Crustle casts Foresight");
162   }
163
164   public static void swarm() {
165     System.out.println("Crustle casts Swarm");
166   }
167
168   // It's been called sand-Attack, causing a compile-time error.
169   public void sand_Attack(Crustle p) {
170     System.out.println("Crustle attacks Crustle with Sand- Attack");
171   }
172
173   public void sand_Attack(Dwebble p) {
174     System.out.println("Crustle attacks Dwebble with Sand- Attack");
175   }
176
177   public void splash() {
178     // True
179     // A new String object is built, then String.intern() checks whether it
180     // is present in the strings pool. It is since the contents of
181     // electricGhost are the same as those of electric+ghost. A reference to
182     // the object in the pool is returned, so the references are equal.
183     System.out.println(electricGhost == (electric+ghost).intern());
184     // False
185     // Here and below new String objects are constructed, and references to
186     // them are not the same as to electricGhost, so they're not equal.
187     System.out.println(electricGhost == electric+ghost);
188     System.out.println(electricGhost == new String("Electric"+"Ghost"));
189     System.out.println(electricGhost == new String("ElectricGhost"));
190   }
191 }
```

2. Результаты работы

```
Dwebble attacks Crustle with Sand- Attack
Crustle attacks Dwebble with Sand- Attack
-23
145
-185
```

```
true
false
false
false
Dwebble casts Foresight
false
Dwebble casts Swarm
true
true
true
false
false
true
Crustle casts Swarm
Crustle casts Foresight
Crustle casts Foresight
Crustle attacks Dwebble with Sand- Attack
208
-40
0
false
Dwebble casts Swarm
Dwebble attacks Dwebble with Sand- Attack
```

3. Выводы

В результате проделанной работы я узнал порядок инициализации объектов в Java, а также детали представления строк в объектной структуре языка, вспомнил тонкости работы с числами с плавающей точкой и записи целых чисел в памяти и в разных системах счисления.