

A. Následující kód se nepřeloží. Vlnovkou vyznačte kde je problém a (pokud možno) jednou větou napište proč.

```
class Stack {  
    int[] contents = new int[1000];  
    int offset = 0;  
  
    public void push(int number) {  
        contents[offset++] = number;  
    }  
    public int pop() { return contents[--offset]; }  
    public static int top() {  
        return contents[offset - 1];  
    }  
    public static void f() {  
        Stack s = new Stack();  
        s.push(1);  
        System.out.println(s.top());  
    }  
}
```

**B.** Nakreslete objekty v paměti vytvořené voláním metody `h`, vyznačte obsah jejich atributů.

```
class Node {
    int contents;
    Node next;

    public Node(int c) { contents = c; next = null; }
    public void f(Node n) {
        if (n.next == null) n.next = this;
        else f(n.next);
    }
    public void g(Node n) {
        Node temp = next;
        next = n;
        n.next = temp;
    }

    public static void h() {
        Node a = new Node(1);
        Node b = new Node(2);
        Node c = new Node(3);
        Node d = new Node(4);
        Node e = new Node(5);
        a.g(b);
        a.g(c);
        b.f(d);
        a.g(e);
    }
}
```

C. Zlikvidujte duplicity v níže napsaném kódu, dejte si záležet, aby vámi napsaný kód byl syntakticky správně (šel přeložit) a rozhraní tříd zůstalo stejné. Tip: použijte dědičnost, asi budete muset někde přidat další metodu „navíc“.

```
class Rectangle {
    int x, y;
    /* dalsi atributy */

    public void moveLeft(int distance) {
        x -= distance; repaintYourself();
    }
    public void moveRight(int distance) {
        x += distance; repaintYourself();
    }
    public void moveUp(int distance) {
        y -= distance; repaintYourself();
    }
    public void moveDown(int distance) {
        y += distance; repaintYourself();
    }
    void repaintYourself() { /* kod */ }
}

class Triangle {
    int x, y;
    /* dalsi atributy */

    public void moveLeft(int distance) {
        x -= distance; repaint();
    }
    public void moveRight(int distance) {
        x += distance; repaint();
    }
    public void moveUp(int distance) {
        y -= distance; repaint();
    }
    public void moveDown(int distance) {
        y += distance; repaint();
    }
    void repaint() { /* kod */ }
}
```