Lösningsförslag till tentamen för TDA540 Objektorienterad Programmering

Institutionen för Datavetenskap
CTH HT-15, TDA540

Dag: 2016-04-09, *Tid:* 14.00-18.00

Uppgift 1

Metoden `mystery` vänder om ett fält. Utskriften är:

```
[4, 9, 3, 8, 5]
```

Uppgift 2

Utskriften blir:

```
Johan / 13 / 7
Anna / 11 / 5
```

Uppgift 3

Vid kompilering av metoden erhålls följande kompileringsfel:

```
missing return statement
```

Eftersom sista `if`-satsen saknas en `else`-gren med en `return` sats, dvs om `n` är större eller lika med 1000 kommer metoden inte att returnerar något. Vi kan korrigerar det med att lägga till en `else`-gren eller genom att ta bort sista if statsen:

```
public static String storlek(int n) {
  if (n < 0)
    return "Negative";
  else if (n < 100)
    return "Small";
  else
    return "Large";
}
```

Uppgift 4

```
public static int[] bin(int n, int d) {
  final int BASE = 2;
  int[] bits = new int[d];

  for (int i = d-1; i >= 0; i--) {
    bits[i] = n % BASE;
    n /= BASE;
  }

  return bits;
}

public static int[] parity(int n) {
  final int BITS = 7;
  int[] xs = bin(n, BITS);
  int[] ys = new int[BITS + 1];
  boolean even = true;

  for (int i = 0; i < xs.length; i++) {
    ys[i+1] = xs[i];
    if (xs[i] == 1) even = !even;
  }

  if (!even) ys[0] = 1;

  return ys;
}
```

Uppgift 5

```
public class Parity {
  public static void main(String[] args) {
    dialogParity();
  }

  // Check binary format
  private static boolean isBin(String b) {
    for (char c : b.toCharArray())
      if (!(c == '0' || c == '1'))
        return false;

    return true;
  }

  private static boolean checkParity(String n) {
    char[] bits = n.toCharArray();
    boolean even = true;

    for (int i = 1; i < bits.length; i++)
```

```
      if (bits[i] == '1') even = !even;

    return even == (bits[0] == 1);
  }

  // Kommandofönster version
  public static void cmdParity() {
    boolean done = false;
    Scanner sc = new Scanner(System.in);

    while (!done) {
      System.out.print("Give a binary number: ");
      if (sc.hasNext()) {
        String n = sc.next();
        if (isBin(n))
          if (checkParity(n))
            System.out.println("Parity check succeeded!");
          else
            System.out.println("Parity check failed!");
        else
          System.out.println("The input should be a binary number!");
      } else done = true;
    }
  }

  // Dialogrutor version
  public static void dialogParity() {
    boolean done = false;

    while (!done) {
      String input = JOptionPane.showInputDialog("Give a binary number: ");
      if (input != null) {
        Scanner sc = new Scanner(input);
        String n = sc.next();
        if (isBin(n)) {
          String msg = "Parity check " + (checkParity(n) ? "succeeded" : "failed");
          JOptionPane.showMessageDialog(null, msg);
        } else
          JOptionPane.showMessageDialog(null, "The input should be a binary number!");
      } else done = true;
    }
  }
}
```

Uppgift 6

```
public static List<Integer> difference(List<Integer> xs, List<Integer> ys) {
  List<Integer> res = new ArrayList<>();

  for (Integer x : xs)
    if (!ys.remove(x)) res.add(x);
```

```
  return res;
}
```

Uppgift 7

```
public static int[][][] scale(int[][][] samples, int n) {
  int[][][] newSamples = new int[samples.length * n][samples[0].length * n][3];

  for (int row = 0; row < samples.length; row++) {
    for (int col = 0; col < samples[row].length; col++) {
      for (int c = 0; c < samples[row][col].length; c++) {
        int rn = row * n;
        int cn = col * n;
        for (int i = 0; i < n; i++) {
          for (int j = 0; j < n; j++) {
            newSamples[rn + i][cn + j][c] = samples[row][col][c];;
          }
        }
      }
    }
  }

  return newSamples;
}
```

Uppgift 8

```
public static class Encode {
  public static String escape(char ch) {
    switch (ch) {
      case ':' : return "%3A";
      case '/' : return "%2F";
      case '?' : return "%3F";
      case '=' : return "%3D";
      default  : return ch + "";
    }
  }

  public static String encodeURL(String url) {
    String res = "";

    for (char ch : url.toCharArray())
      res += escape(ch);

    return res;
  }
}
```

Uppgift 9

```
public static String copy(int n, String x) {
  String res = "";

  for (int i = 0; i < n; i++)
    res += x;

  return res;
}

public static String julgran(int n) {
  String res = "", margin, stripe;

  for (int i = 0; i < n; i++) {
    margin = copy(i, ".");
    stripe = copy(n - 1 - i, "*-") + "*";
    res    = margin + stripe + margin + "\n" + res;
  }
  return res;
}
```

Uppgift 11

Exempel på olika tänkbara klasser och attribut (finns fler)

```
public class Album {
  private String name;
  private List<Song> songs;
  private Artist artist;
  private int totalDuration;

  public Album(String name, List<Song> songs, Artist artist) {
    this.name     = name;
    this.songs    = songs;
    this.artist   = artist;
    totalDuration = addDurations();
  }

  private int addDurations() {
    int l = 0;

    for (Song s : songs) l += s.getDuration();

    return l;
  }
}

public enum Genre {
  CLASSICAL, ROCK, POP, SOUL
}
```

```
public class Artist {
  private String name;
  private Genre genre;
  private int age;

  public Artist(String name, Genre genre, int age) {
    this.name  = name;
    this.genre = genre;
    this.age   = age;
  }
}

public class Song {
  private String title;
  private int duration;
  private Genre genre;

  public Song(String title, int duration, Genre genre) {
    this.title    = title;
    this.duration = duration;
    this.genre    = genre;
  }

  public int getDuration() {
    return duration;
  }
}

public class Collection {
  private List<Album> albums;

  public Collection() {
    albums = new ArrayList<Album>();
  }
}
```