Lösningsförslag till tentamen för TDA540 Objektorienterad Programmering

Institutionen för Datavetenskap
CTH HT-15, TDA540

Dag: 2016-08-25, *Tid:* 14.00-18.00

Uppgift 1

Utskriften blir:

```
x = 42 och y = 1337
x = 42 och y = 1337
```

Uppgift 2

Utskriften blir:

```
1 2 3 4 5
1 2 5 4 3
```

Uppgift 3

Utskriften blir:

```
321
```

Uppgift 4

Vid kompilering av metoden erhålls två kompileringsfel:

1. det saknas en `;` efter `int m = 1`
2. det saknas en `return` sats

Vi kan korrigera det så här:

```java
public static int power(int x, int n) {
  int m = 1;
  for (int i = 0; i < n; i++)
    m *= x;
  return m;
}
```

Uppgift 5

```java
public static int[][][] cut(int[][][] samples, int x, int y, int n, int m) {
  int[][][] newSamples = new int[n][m][3];

  for (int row = 0; row < n; row++)
    for (int col = 0; col < m; col++)
      for (int c = 0; c < 3; c++)
        newSamples[row][col][c] = samples[row + x][col + y][c];

  return newSamples;
}
```

Uppgift 6

```java
public class Loan {
  public static void main(String[] args) {
    cmdLoan();
  }

  private static double calculatePayment(double loan, double rate, int years) {
    int n = 12 * years;
    double c = rate / 12.0 / 100.0;
    return loan * c * Math.pow(1 + c, n) / (Math.pow(1 + c, n) - 1);
  }

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

    while (!done) {
      System.out.print("Give loan amount, interest, and number of years: ");
      if (sc.hasNext()) {
        double loan  = sc.nextDouble();
        double rate  = sc.nextDouble();
        int    years = sc.nextInt();
        if (loan >= 0.0 && rate >= 0.0 && years >= 0) {
          double amount = calculatePayment(loan, rate, years);
          System.out.println("Monthly amount: " + Math.round(amount));
        } else
          System.out.println("All inputs should be positive!");
```

```
      } else done = true;
    }
  }

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

    while (!done) {
      String input = JOptionPane.showInputDialog("Give loan amount, interest, and number of years: ");
      if (input != null) {
        Scanner sc = new Scanner(input);
        double loan  = sc.nextDouble();
        double rate  = sc.nextDouble();
        int    years = sc.nextInt();
        if (loan >= 0.0 && rate >= 0.0 && years >= 0) {
          double amount = calculatePayment(loan, rate, years);
          JOptionPane.showMessageDialog(null, "Monthly amount: " + Math.round(amount));
        } else
          JOptionPane.showMessageDialog(null, "All inputs should be positive!");
      } else done = true;
    }
  }
}
```

Uppgift 7

```
public static boolean checkPersonNumber(String number) {
  int m     = 0;
  boolean b = true;
  char[] cs = number.toCharArray();
  int check = Character.getNumericValue(cs[cs.length - 1]);

  for (int i = 0; i < cs.length - 1; i++) {
    int n = Character.getNumericValue(cs[i]) * (b ? 2 : 1);
    b = !b;
    m += n % 10 + n / 10;
  }
  return check == (10 - (m % 10));
}
```

Uppgift 8

```
public class RLE {
  public static String compress(String input) {
    if (input.length() > 0) {
      char c = input.charAt(0);
      int  i = 1;

      while (i < input.length() && input.charAt(i) == c) i++;
```

```
      return Integer.toString(i) + c + compress(input.substring(i));
    }

    return "";
  }

  public static String decompress(String input) {
    String res   = "";
    String count = "";

    for (char c : input.toCharArray())
      if (Character.isAlphabetic(c)) {
        res += replicate(c, Integer.parseInt(count));
        count = "";
      } else
        count += c;

    return res;
  }

  private static String replicate(char c, int n) {
    String res = "";

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

    return res;
  }
}
```

Uppgift 9

1.
```
public interface Coordinate {
  public void setX(int x);
  public int  getX();
  public void setY(int y);
  public int  getY();
}

public static class Point implements Coordinate {
  private int x, y;

  public Point(int x, int y) {
    setX(x);
    setY(y);
  }

  public void setX(int x) {
    this.x = x;
  }
```

```
  public void setY(int y) {
    this.y = y;
  }

  public int getX() {
    return x;
  }

  public int getY() {
    return y;
  }
}
```

2. Om `p` har typen `Coordinate` då kan vi bara anropa metoderna som är definierat i gränssnittet `Coordinate`. Om `p` har typen `Point` kan vi anropa metoderna som är definierat i gränssnittet `Coordinate` men också `Point` klassens egna publika metoder. Dessutom kan har vi tillgång till publika instansvariabler.

3.
```
public interface Color {
  public void setColor(int color);
  public int  getColor();
}

public static class Pixel extends Point implements Color {
  private int color;

  public Pixel(int x, int y, int color) {
    super(x, y);
    this.color = color;
  }

  public void setColor(int color) {
    this.color = color;
  }

  public int getColor() {
    return color;
  }
}
```

Uppgift 10

Exempel på olika tänkbara klasser och attribut (finns fler)

```
public class GameOfLife {
  private Grid grid;
  private int ticks;

  public void setStartGrid() {}
  public void startGame() {}
```

```
  public void stopGame() {}
}

public class Grid {
  private Cell[][] cells;
  private int width;
  private int height;

  public Cell getCell(int x, int y) {}
  public void resize(int w, int h) {}
  public boolean checkCell(int x, int y) {}
}

public class Cell {
  private boolean state;

  public void changeState(boolean state) {}
}
```