Lösningsförslag till tentamen för TDA540 Objektorienterad Programmering

Institutionen för Datavetenskap

CTH HT-17, TDA540

Dag: 2018-08-30, *Tid:* 14.00-18.00

Uppgift 1

a) `public` en modifierare som antyder att enheten (instansvariabel, metod, etc.) är tillgängligt för alla

`class` används för en klassdeklaration som är ett mall för att skapa objekt

`static` en modifierare som anger att en enhet (variabel eller metod) tillhör själva klassen och är gemensamt för alla klassens objekt

`void` returtyp som antyder att en metod har inget returvärde

`int` används för att deklarera en variabel som kan spara en heltal

`for` används för att skapa en räknareloop

`if` används för selektering mellan två satser beroende på en boolesk uttryck

`break` slutar exekvera nuvarande loop

b)

Klass	*Signatur*	*Returtyp*	*Statisk*
`Uppgift1`	`main(String[] args)`	`void`	ja
`Uppgift1`	`maf(int[] xs, int[] ys)`	`void`	ja
`PrintStream`	`println(int i)`	`void`	nej

c)
```
3 int[] xs = {3, 3};
4 int[] ys = {2, 3, 3, 3, 1, 4, 5, 2, 3, 3};
6 maf(xs, ys); // xs och ys är referensvärden
10 int i, j;
12 for (i = 0; i <= ys.length - xs.length; i++) // i = 0 och mindre än 10 - 2
13 for (j = 0; j < xs.length; j++) // j = 0 och mindre än 2
```

```
14 if (xs[j] != ys[i + j]) // xs[0] = 3 och y[0] = 3, så boolesk uttryck är sant
15 break;
```

d) Utskriften:

```
1
2
8
```

Metoden `maf` skriver ut alla start-indices av förekomster av `xs` i `ys`.

e)
```
public static List<Integer> maf(int[] xs, int[] ys) {
  List<Integer> res = new ArrayList<>();

  for (int i = 0; i <= ys.length - xs.length; i++) {
    int j = 0;

    while (j < xs.length && xs[j] == ys[j + i])
      j++;

    if (j == xs.length)
      res.add(i);
  }

  return res;
}
```

Uppgift 2

- Det saknas ett 'större än'-tecken bakom `List<Character`.
- `List` är ett interface och det går inte att skapa objekt med ett interface. Det resulterar i en kompileringsfel.
- Det saknas en semikolon bakom `res.add(c)`.
- Det saknas en `return`-sats.

Förbättrade versionen:

```
import java.util.ArrayList;
import java.util.List;

class Error {
  public static List<Character> fun(Integer size, Character c) {
    List<Character> res = new ArrayList<Character>();

    for (int i = 0; i < size; ++i)
      res.add(c);
```

```
    return res;
  }
}
```

Uppgift 3

a) Utskriften:

```
42 is in range!
java.lang.NumberFormatException: Way too small!
Hmm...
1000 is in range!
```

b) Det är klassen `Throwable`.

c) Vi kommer att få ett kompileringsfel för vi tar inte hand om (checked) undantag med typen `Exception` som kastas av metoden `inRange`. Typen `NumberFormatException` är en subtyp till `Exception`, den kan inte 'fånga' `Exception`.

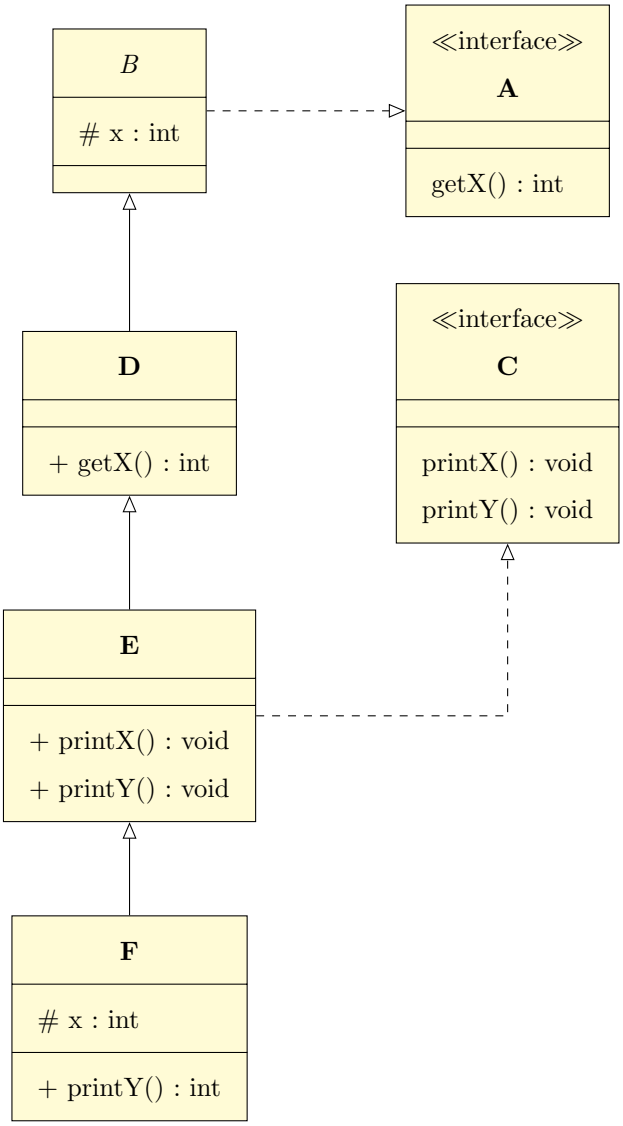
Uppgift 4

```
public class Circle {
  private final Point  center;
  private final double radius;
  private final double EPS = 0.001;

  Circle(Point center, double radius) {
    this.center = new Point(center.getX(), center.getY());

    if (radius > 0)
      this.radius = radius;
    else
      throw new IllegalArgumentException("The radius must be greater than 0.");
  }

  public Point getCenter() {
    return new Point(center.getX(), center.getY());
  }

  public double getRadius() {
    return radius;
  }

  public double area() {
    return Math.pow(radius, 2.0) * Math.PI;
  }

  public double circumference() {
    return 2 * radius * Math.PI;
```

```
  }

  public boolean contains(Circle c) {
    return radius - distanceTo(c) - c.getRadius() > -EPS;
  }

  public boolean overlaps(Circle c) {
    return distanceTo(c) - radius - c.getRadius() < EPS;
  }

  private double distanceTo(Circle c) {
    Point p = c.getCenter();
    int dx  = p.getX() - center.getX();
    int dy  = p.getY() - center.getY();
    return Math.sqrt(Math.pow(dx, 2) + Math.pow(dy, 2));
  }
}
```

Uppgift 5

a) Klassdiagrammet ser ut så här:



b) 1. 0
2. 0
3. 0
4. 0
5. 4
6. 0
7. 0
8. 0

c) 1. 0

2. skriver inte ut något
3. 0
4. 0

d) 1. Inkorrekt: ett interface kan inte instansieras
2. Korrekt
3. Inkorrekt: man får en typfel för `D` är ingen subtyp av `C`
4. Korrekt
5. Korrekt
6. Inkorrekt: man får en typfel för `D` är ingen subtyp av `F`
7. Korrekt
8. Inkorrekt: man får en typfeil för `ArrayList<D>` är ingen subtyp av `ArrayList<E>` (even if `E` är subtyp av `D`)
9. Inkorrect: en abstrakt klass kan inte instansieras

Uppgift 6

```java
public class Keith {
  public static List<Integer> factorize(int n) {
    List<Integer> factors = new ArrayList<Integer>();
    final int BASE = 10;

    while (n > 0) {
      factors.add(0, n % BASE);
      n /= BASE;
    }

    return factors;
  }

  public static int sum(List<Integer> xs) {
    int res = 0;
    for (int x : xs)
      res += x;
    return res;
  }

  public static boolean isKeithNumber(int n) {
    List<Integer> numbers = factorize(n);
    int m = 0;

    do {
      m = sum(numbers);

      if (n == m)
        return true;
```

```
      else {
        numbers.remove(0);
        numbers.add(m);
      }
    } while (n > m);

    return false;
  }
}
```