

Praktikum 4

Kombinatorika. Rekursioon (II)

- Hulga kõikide alamhulkade hulk

Harjutusülesanded

- KMB_7. Alamhulkade hulk mitterekursiivselt
 - KMB_8. Alamhulkade hulk rekursiivselt
 - KMB_9. Osasummad
 - KMB_10. Seljakoti komplekteerimine
- Hulga tõkestatud alamhulkade hulk
 - KMB_11. Seljakoti kiirem komplekteerimine

Iseseisev töö nr 2

Hulga kõikide alamhulkade hulk

Tähistus: $P(A)$ – hulga A kõigi alamhulkade hulk.

Variant I. Mitterekursiivne

$P(A)$ elemendid saame, valides alamhulgad hulgast A maskide abil,
 $mask = 0, 1, \dots, 2^n - 1$, kus $n = |A|$

Variant II. Rekursiivne või mitterekursiivne

$$P(B \cup \{x\}) = P(B) \cup X,$$

kus X on hulk, mille iga element on saadud elemendi x lisamisel ühele hulga $P(B)$ elemendile.

Näide:

$$A = \{2, -1, 3\}$$

$$B = \{2, -1\}$$

$$A = B \cup \{3\}$$

$$P(B) = \{\{\}, \{2\}, \{-1\}, \{2, -1\}\}$$

$$P(A) = P(B \cup \{3\}) = P(B) \cup X =$$

$$= \{\{\}, \{2\}, \{-1\}, \{2, -1\}\} \cup \{\{3\}, \{2, 3\}, \{-1, 3\}, \{2, -1, 3\}\} =$$

$$= \{\{\}, \{2\}, \{-1\}, \{2, -1\}, \{3\}, \{2, 3\}, \{-1, 3\}, \{2, -1, 3\}\}$$

Variant III. Mitterekursiivne

Hulga A kõigi alamhulkade hulk hulga A kombinatsioonide ühendina:

$$P(A) = \{komb(A, 0)\} \cup$$

$$\cup \{komb(A, 1)\} \cup$$

...

$$\cup \{komb(A, |A|)\}$$

Näide:

$$P(\{2, -1, 3\}) = \{\{\}\} \cup$$

$$\cup \{\{2\}, \{-1\}, \{3\}\} \cup$$

$$\cup \{\{2, -1\}, \{2, 3\}, \{-1, 3\}\} \cup$$

$$\cup \{\{2, -1, 3\}\} =$$

$$= \{\{\}, \{2\}, \{-1\}, \{3\}, \{2, -1\}, \{2, 3\}, \{-1, 3\}, \{2, -1, 3\}\}$$

Harjutusülesanded

KMB_7. Alamhulkade hulk mitterekursiivselt

Kirjutada ja testida mitterekursiivne generaator järgmise ülesande lahendamiseks.

Antud: hulk a (listina)

Tulemus: järjekordsel pöördumisel ($next()$) antakse välja a järjekordne alamhulk (listina)

Testi tulemuse näide:

```
Antud hulk: [-1, 3, 4, 2]
Selle alamhulkade hulk:
[]
[-1]
[3]
[4]
[2]
[-1, 3]
[-1, 4]
[3, 4]
[-1, 2]
[3, 2]
[4, 2]
[-1, 3, 4]
[-1, 3, 2]
[-1, 4, 2]
[3, 4, 2]
[-1, 3, 4, 2]
```

KMB_8. Alamhulkade hulk rekursiivselt

Kirjutada ja testida rekursiivne generaator järgmise ülesande lahendamiseks.

Antud: hulk a (massiivina)

Tulemus: järjekordsel pöördumisel ($next()$) antakse välja a järjekordne alamhulk (massiivina).

KMB_9. Osasummad

Kirjutada ja testida meetod järgmise ülesande lahendamiseks.

Antud: täisarvude hulk a listina

Tulemus: hulga a kõik osasummad (alamhulkade summad) listina, sorteeritult, võrdsed ühekordselt

Tulemuse näide:

```
Antud: [2, -3, 3, 3]
Selle osasummadeks on:
[-3, -1, 0, 2, 3, 5, 6, 8]
```

```
import java.util.LinkedHashSet;
```

Java (I)

Korduvate eemaldamine listist s:

```
LinkedHashSet<Integer> h = new LinkedHashSet<Integer>(s);
s.clear();
s.addAll(h);
```

```
import java.util.TreeSet;
```

Java (II)

Korduvaid elemente ei ole. Elemendid hoitakse loomulikus järjestuses.
Soovitav kasutada osasummade hoidmiseks.

KMB_10. Seljakoti komplekteerimine

Koostada programm seljakotiülesande lahendamiseks “jõumeetodil” – läbi vaadatakse kõikvõimalikud esemete komplektid, vt ka [Kiho 2003, lk 17].

Väljundi näide

```
Esemete kaalud: [10, 25, 9, 13, 7]
Esemete hinnad: [30, 62, 98, 29, 41]
Seljakoti lubatud kaal: 50
-----
Parim komplekt: [1, 2, 4]
                 kaaludega [25, 9, 7]
                 hindadega [62, 98, 41]
Selle komplekti hind: 201
Selle komplekti kaal: 41
```

Hulga tõkestatud alamhulkade hulk

Ülesannetes, kus tuleb leida vaid teatavat kriteeriumi rahuldavad alamhulgad saab tihtipeale vältida kõikvõimalike alamhulkade läbivaatamist lahendamise käigus.

Kasulikuks võib osutada kontrollifunktsioon *veel(ah)*, mis argumendiks oleva alamhulga *ah* korral on väärtusega *tõene* parajasti siis, kui vaatlusele „tasub“ võtta veel ka alamhulk *ah'*, mis on saadud alamhulgast *ah* sellele veel ühe elemendi lisamise teel.

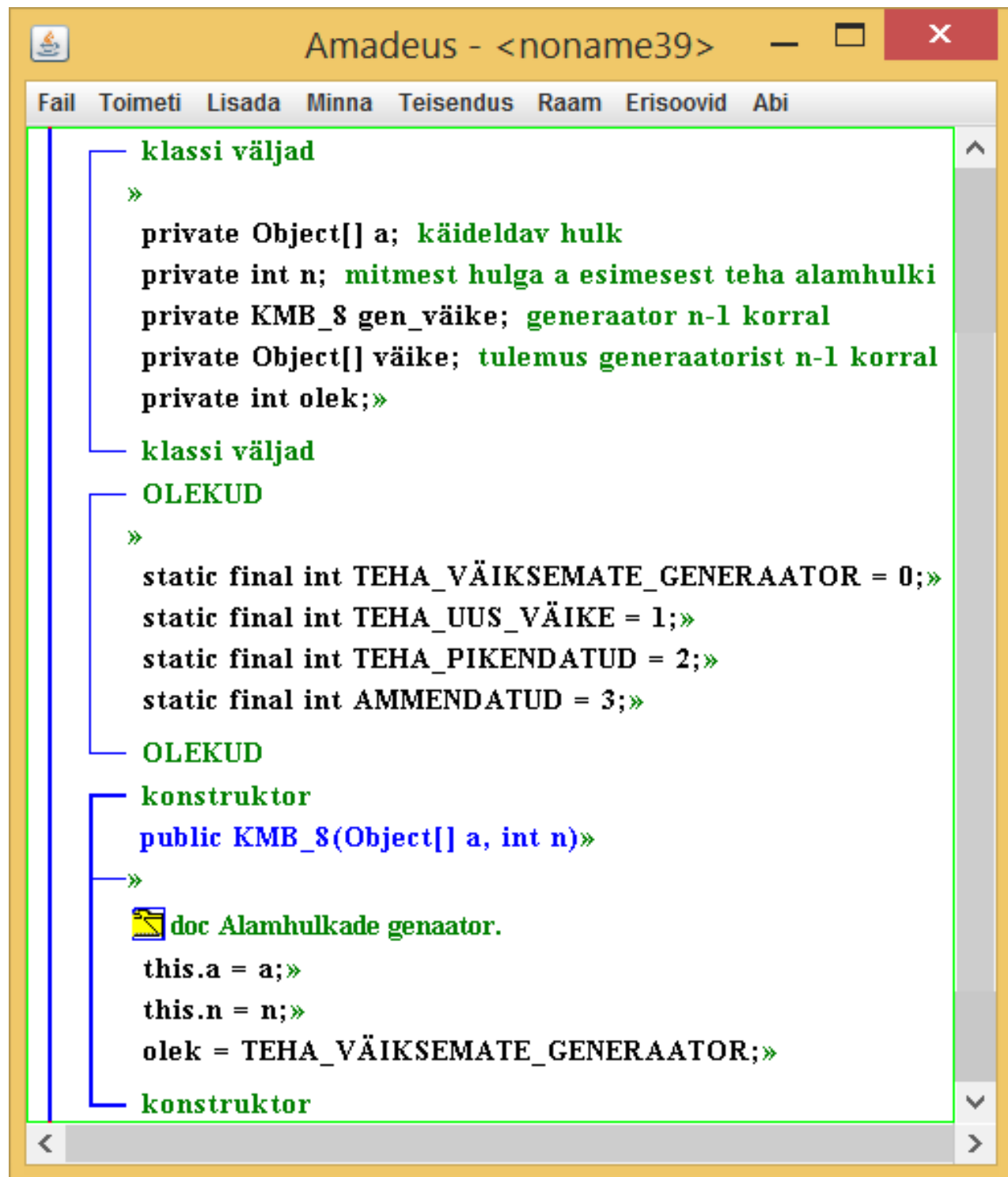
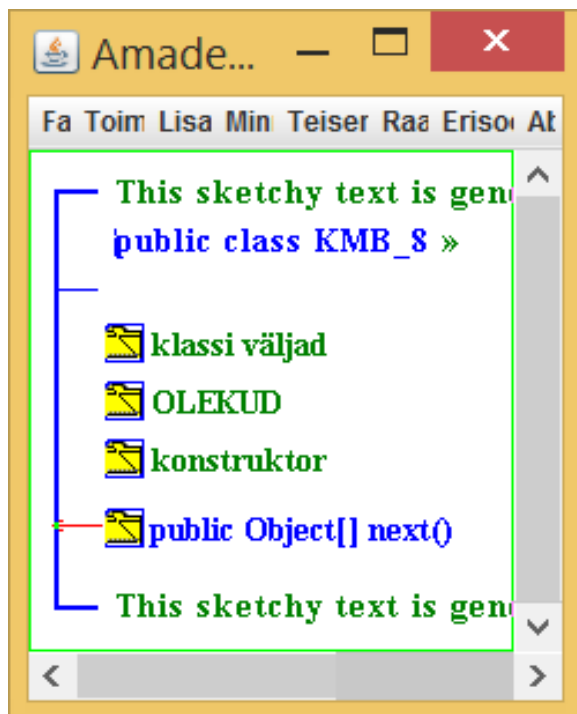
KMB_11. Seljakoti kiirem komplekteerimine

Koostada programm seljakotiülesande lahendamiseks, milles on tõkestatud „ülekaaluliste“ esemekomplektide läbivaatamine.

Iseseisev töö nr 2

Esitamise tähtaeg rühmal J. Liivi 2-206, E12 -14: **10. oktoober 2014, kell 12.15**

Esitamise tähtaeg rühmal J. Liivi 2-207, T 14 -16: **4. oktoober 2014, kell 14.15**




```
Amadeus - <noname40>
Fail Toimeti Lisada Minna Teisendus Raam Erisoovid Abi

public Object[] next()»
  doc Järjekordne alamhulk.
  baasjuhud
  for(;;)
    ?? switch (olek)»
      case TEHA_VÄIKSEMATE_GENERAATOR :»
        gen_väike = new KMB_S(a, n-1);»
      case TEHA_UUS_VÄIKE :»
        väike = gen_väike.next();»
        if (väike == null)»
          väiksemad (ja nende pikendused) kõik juba välja antud
          olek = AMMENDATUD;»
          return null;»
        olek = TEHA_PIKENDATUD;»
        return väike;»
      case TEHA_PIKENDATUD:»
        Object[] pikendatud = new Object[väike.length+1]; pikendatu k
        » kopeerida väike ==> pikendatud algusesse:
        System.arraycopy(väike, 0, pikendatud, 0, väike.length);»
        » a viimane element ==> pikendatud viimasele kohale:
        pikendatud[pikendatud.length-1] = a[n-1];»
        olek = TEHA_UUS_VÄIKE;»
        return pikendatud;»
      default :»
        println("KMB_S:next(): Olek vale:"+olek);»
        System.exit(0);»
```

```
Amadeus - <noname41>
Fail Toimeti Lisada Minna Teisendus Raam Erisoovid Abi

public Object[] next()»
  doc Järjekordne alamhulk.
  <p>
  @return (antakse välja) jrk hulga a alamhulk (massiivina)
  doc Järjekordne alamhulk.
  baasjuhud
  »
  if (olek == AMMENDATUD)»
    return null;»
  if (n == 0)»
    olek = AMMENDATUD;»
    return new Object[0]; anda välja tühihulk
  baasjuhud
  This sketchy text is generated by AMADEUS
```

väike juba liiga „raske“?
olek = TEHA_UUS_VÄIKE;
continue;