Praktikum 3

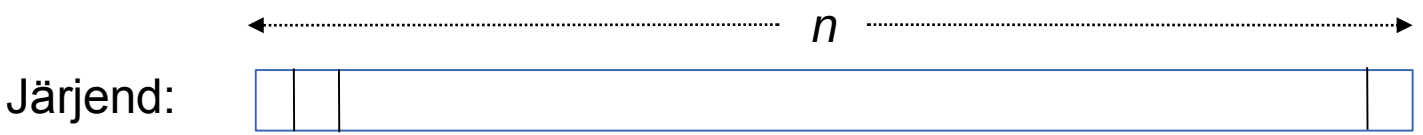
Kombinatoorika. Rekursioon (I)

Kombinatsioonide mitterekursiivselt leidmise idee
Kombinatsioonide rekursiivselt leidmise idee
Permutatsioonide rekursiivselt leidmise idee
Variatsioonid

Harjutusülesanded

- **KMB_1. Kombinatsioonid mitterekursiivselt**
- **KMB_2. Kombinatsioonid rekursiivselt**
- **KMB_3. Sõne sümbolite permutatsioonid**
- **KMB_4. Kombinatsioonide generaator**
- **KMB_5. Permutatsioonide generaator**
- **KMB_6. Variatsioonid**

Kombinatsioonide mitterekursiivselt leidmise idee



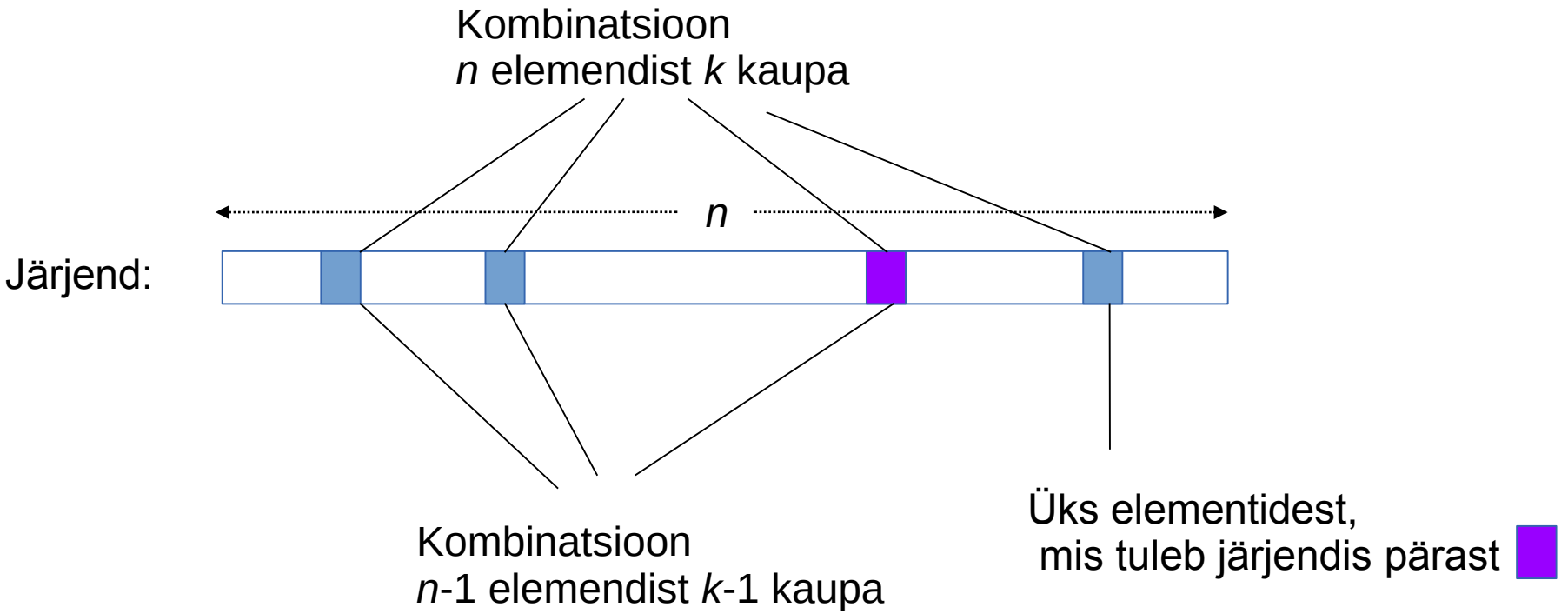
Iga kombinatsioon *n*-elemendilistest järjendist *k* kaupa on määratud järjendi ühe maskiga. Maskideks on bitijärjendid pikkusega *n*, milles igas *k* ühte. Näiteks maskid *n* = 5 ja *k* = 3 korral:

00111
01011
01101
01110
10011
10101
10110
11001
11010
11100

A B C D E	järjend
1 0 1 1 0	mask
A C D	maskile vastav kombinatsioon (5-st 3 kaupa)

Kõikvõimalikud bitijärjendid pikkusega *n* on arvude
0, 1, 2, ..., 2^{n}-1
kahendkujude *n* viimast numbrit; n = 5 korral siis 00000, 00001, 00010, ..., 11111.

Kombinatsioonide rekursiivselt leidmise idee



See kombinatsioon
7 elemendist *3* kaupa (B C E):

A B C D E F G H

annab

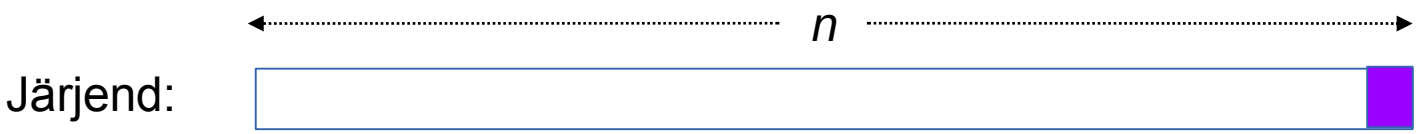
3 kombinatsiooni
8 elemendist *4* kaupa:

B C E F

B C E G

B C E H

Permutatsioonide rekursiivselt leidmise idee



Iga permutatsioon *n* elemendist on saadav sel teel, et mingile permutatsioonile *n*-1 (esimesest) elemendist on mingisse positsiooni lisatud järjendi viimane element.
Näiteks 6-liikmelise järjendi

A B C D E F

esimese 5 liikme üks permutatsioone on B A E D C

Sellest saame järgmised antud 6-liikmelise järjendi permutatsioonid:

F B A E D C
B F A E D C
B A F E D C
B A E F D C
B A E D F C
B A E D C F

Variatsioonid

Variatsioonideks järjendi elementidest *k* kaupa on selle järjendi kõigi *k*-kaupa kombinatsioonidest saadavad permutatsioonid.

Harjutusülesanded

KMB_1. Kombinatsioonid mitterekursiivselt

Kirjutada ja testida mitterekursiivne meetod järgmise ülesande lahendamiseks.
Antud: sõnede järjend *a* (massiivina), arv *k* (0<=*k*<=a.length)
Tulemus: tagastatakse järjendi *a* elementide kombinatsioonid *k* kaupa, maatriksina, mille igas reas üks kombinatsioon

Testi tulemuse näide (järjendis ühesümbolilised sõned):

```
a =  [A, B, C, D, E]
a elementide kombinatsioonid 3 kaupa:

C D E
B D E
B C E
B C D
A D E
A C E
A C D
A B E
A B D
A B C
```

Java

Ühtede arv arvu *m* kahendkujus:
`Integer.bitCount(m)`

Java

Positiivse täisarvu *m* korral on avaldise
`(m>>i) % 2`
väärtuseks arvu *m* kahendkuju *i*-nda järgu bitt (lõpust arvates), nt:
$m = 6 = 110_2$

```
(m>>0) % 2 == 0
(m>>1) % 2 == 1
(m>>2) % 2 == 1
```

Avaldis on kasutatav järjendist elementide valimiseks (maskiga *m*).

Amadeus - workspace2015s\KMB_1.algjava

Fail Toimeti Lisada Minna Teisendus Raam Erisoovid Abi

projekt: AlgJavaWorks2015oCDB

```
preambul
TEST
long kombArv(int N, int R)
String[][] komb(String[] a, int r)»
» Antud: sõnede järjend a
» Tulemus: a elementide kombinatsioonid maatriksina,
» .          iga rida üks kombinatsioon (järjendina)
 int n = a.length;»
 int m = (int)kombArv(n, r);»
 String[][] tulem = new String[m][r];»
 int N = (int)Math.pow(2, n);»
 int tul= 0;  indeks massiivil tulem (rea nr)
* mask = 0 .. N-1»
  if (Integer.bitCount(mask) == r)»
  » mask määrab jrk kombinatsiooni, kui selles on r ühte
   int j = 0;  indeks massiivil tulem[tul] (j on elemendi nr reas)
   a elementide valimine maski järgi
   tul++;»
return tulem;»
```

```
static long kombArv(int N, int R){
// Antud: el arv N, mitme kaupa R
// Tulemus: komb arv N-st R kaupa
    if (N < R) return 0;
    if (N == 0 || R == N) return 1;
    long r = Math.max(R, N-R)+1;
    for (long m = r+1, d = 2; m <= N; m++, d++ ){
        r *= m;
        r /= d;
    }
    return r;
}
```

Idee:

http://stackoverflow.com/questions/2201113/combinatoric-n-choose-r-in-java-math

KMB_2. Kombinatsioonid rekursiivselt

Kirjutada ja testida rekursiivne meetod järgmise ülesande lahendamiseks.

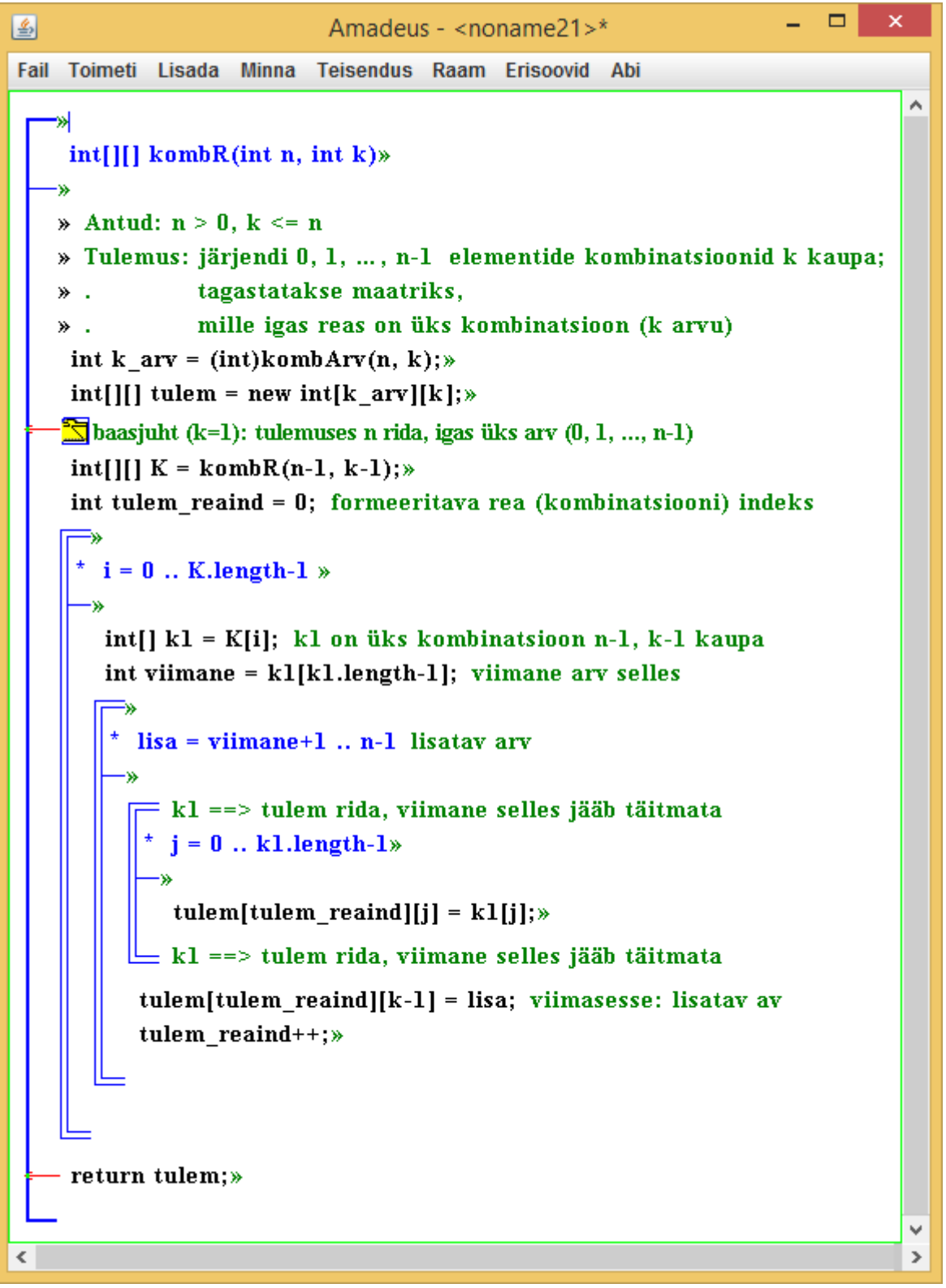
Antud: täisarvud *n* ja *k*, *n*>0, *k*<=*n*

Tulemus: järjendi 0, 1, …, *n*-1 elementide kombinatsioonid *k* kaupa, maatriksina, mille igas reas üks kombinatsioon

Testi tulemuse näide:

```
n = 6  k =4
Järjendi [0, 1, 2, 3, 4, 5]
elementide kombinatsioonid 4 kaupa:

0 1 2 3
0 1 2 4
0 1 2 5
0 1 3 4
0 1 3 5
0 1 4 5
0 2 3 4
0 2 3 5
0 2 4 5
0 3 4 5
1 2 3 4
1 2 3 5
1 2 4 5
1 3 4 5
2 3 4 5
```


Amadeus - <noname21>*
Fail Toimeti Lisada Minna Teisendus Raam Erisoovid Abi
int[][] kombR(int n, int k)»
» Antud: n > 0, k <= n
» Tulemus: järjendi 0, 1, ... , n-1 elementide kombinatsioonid k kaupa;
» . tagastatakse maatriks,
» . mille igas reas on üks kombinatsioon (k arvu)
int k_arv = (int)kombArv(n, k);»
int[][] tulem = new int[k_arv][k];»
baasjuht (k=1): tulemuses n rida, igas üks arv (0, 1, ..., n-1)
int[][] K = kombR(n-1, k-1);»
int tulem_reaind = 0; formeeritava rea (kombinatsiooni) indeks
* i = 0 .. K.length-1 »
int[] k1 = K[i]; k1 on üks kombinatsioon n-1, k-1 kaupa
int viimane = k1[k1.length-1]; viimane arv selles
* lisa = viimane+1 .. n-1 lisatav arv
k1 ==> tulem rida, viimane selles jääb täitmata
* j = 0 .. k1.length-1»
tulem[tulem_reaind][j] = k1[j];»
k1 ==> tulem rida, viimane selles jääb täitmata
tulem[tulem_reaind][k-1] = lisa; viimasesse: lisatav av
tulem_reaind++;»
return tulem;»

KMB_3. Sõne sümbolite permutatsioonid

Kirjutada ja testida rekursiivne meetod järgmise ülesande lahendamiseks.
Antud: sõne *s*
Tulemus: sõne *s* sümbolite permutatsioonid sõnedena

Testi tulemuse näide:

```
Antud: abc
Selle permutatsioonid:
abc
acb
bac
bca
cab
cba
```

Java

Sõne *s* korral on avaldise
`s.substring(0, i) + c + s.substring(i)`
väärtuseks sõne, mis on saadud sõnest *s*
sümboli (muutujast) *c* lisamisel selle *i*-ndasse positsiooni.

KMB_4. Kombinatsioonide generaator
Kirjutada ja testida kombinatsioonide generaator, mis genereerib antud sõnede massiivist antud pikkusega sõnekombinatsioone.

Generaator on klass, mille isendi konstrueerimisel antakse parameetritena sõnede massiiv ja kombinatsioonide pikkus. Klass sisaldab avalikku isendimeetodit *next*(), mille järjekordsel rakendamisel tagastatakse järjekordne sõnede kombinatsioon (või *null*, kui kombinatsioone rohkem ei ole).

Näiteks, kui generaatoriks on klass nimega *Sõnede_komb_gen,* siis testi osa võiks olla:

```
String[] a = {”Tartu”,”Tallinn”,”Tapa”,”Türi”,”Narva”,”Elva”};
...
Sõnede_komb_gen genkomb = new Sõnede_komb_gen(a, 5); // viie kaupa
String[] komb;
while( (komb = genkomb.next()) != null )
          ... väljastada sõnede järjend komb ;
```

KMB_5. Permutatsioonide generaator
Kirjutada ja testida permutatsioonide generaator, mis genereerib antud sõnede listi permutatsioone.

KMB_6. Variatsioonid

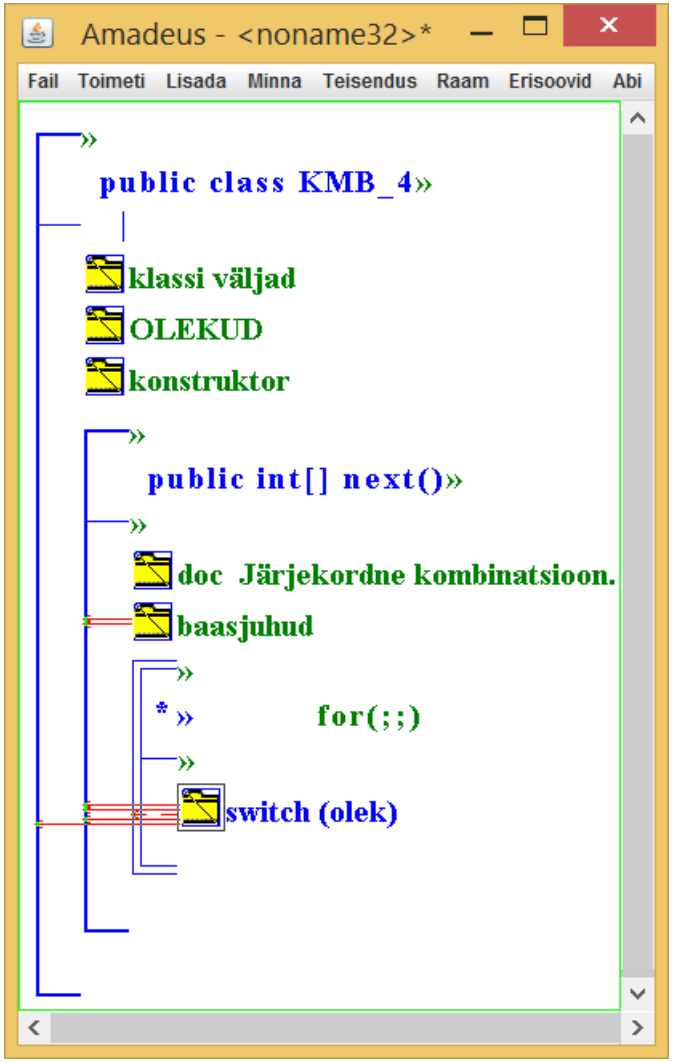
Kirjutada ja testida meetod järgmise ülesande lahendamiseks.

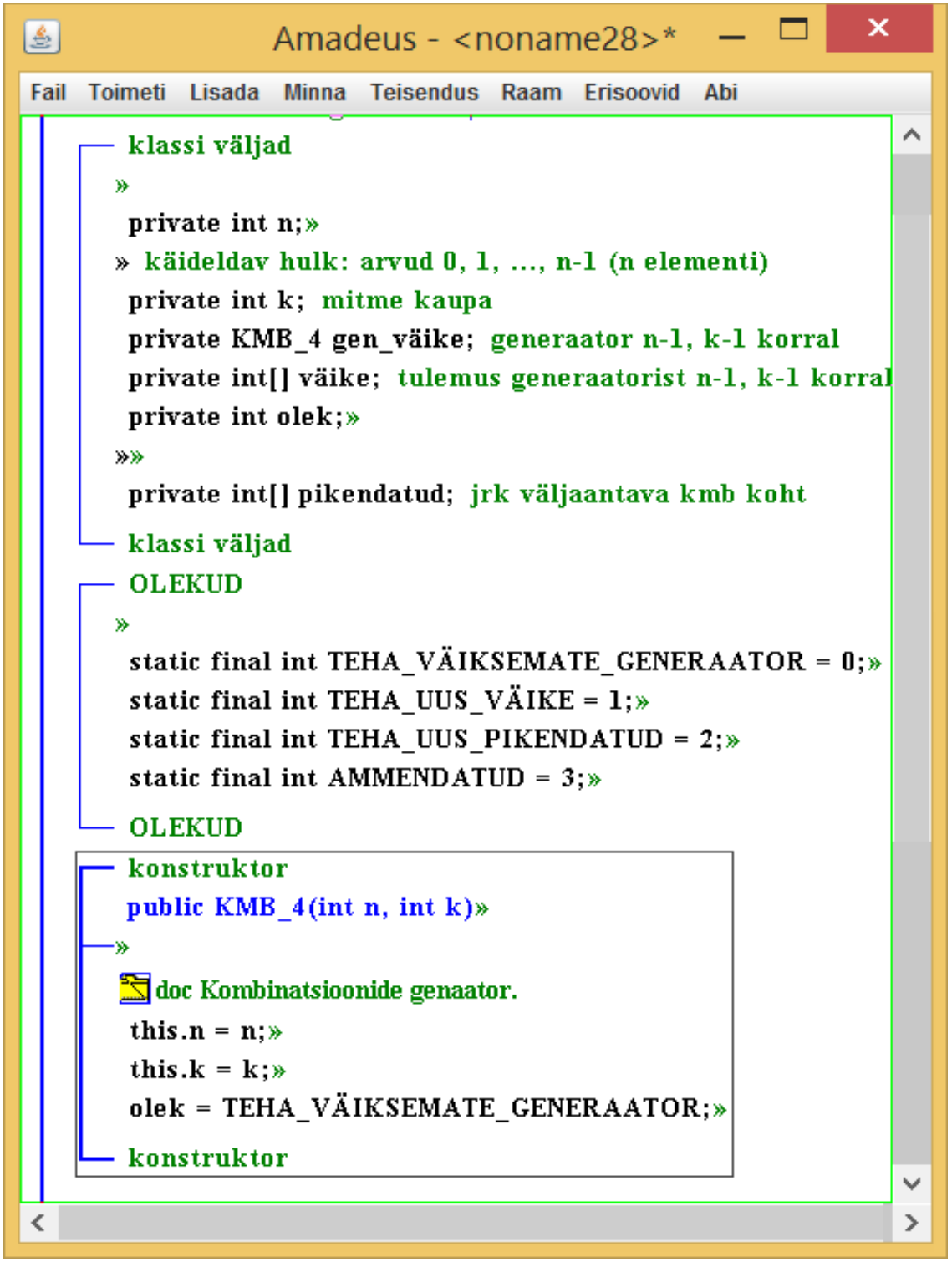
Antud: sõnede massiiv a ja arv *k* (0<=*k*<=a.length)

Tulemus: väljastatud *a* elementide *k* kaupa variatsioonid

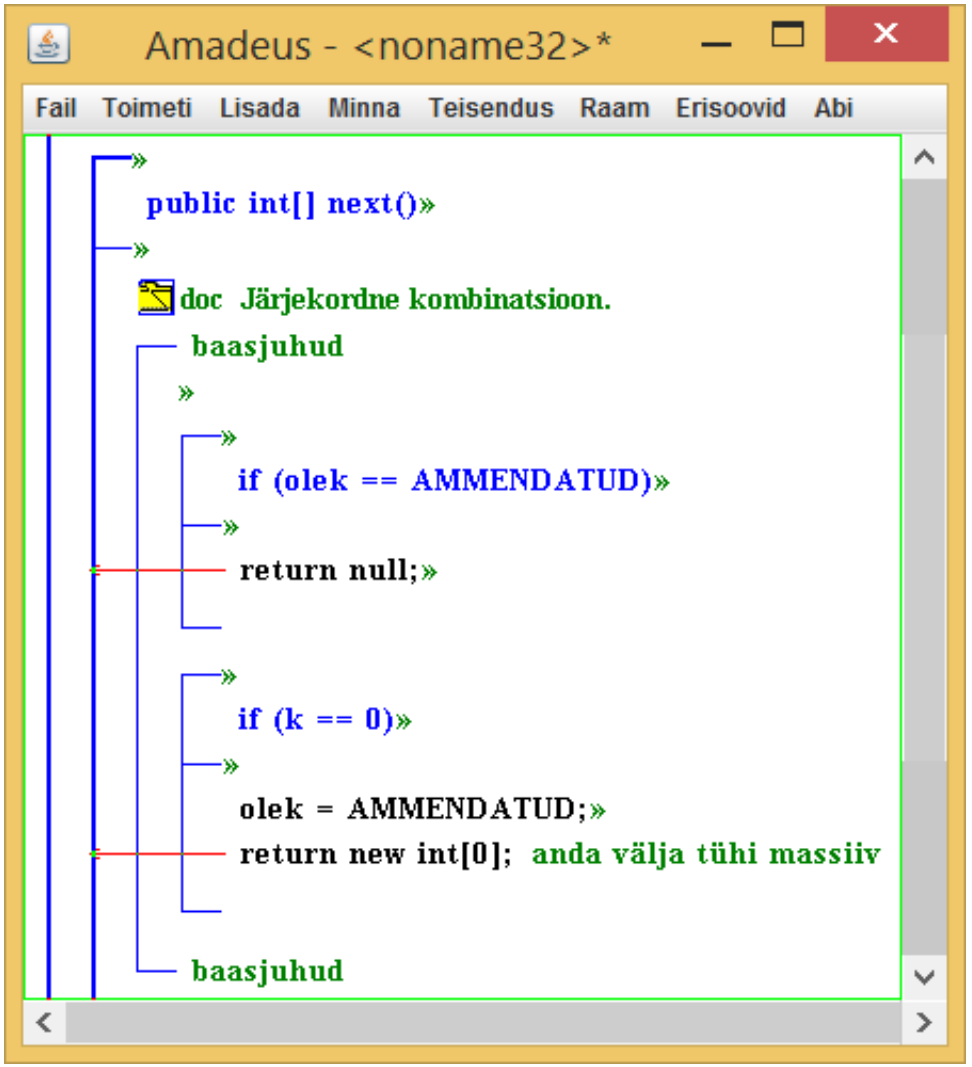
Testi tulemuse näide:

```
Antud:  Ants Bert Cisi Dora Elmo
Variatsioonid 2 kaupa:
[Elmo, Dora]
[Dora, Elmo]
[Elmo, Cisi]
[Cisi, Elmo]
[Dora, Cisi]
[Cisi, Dora]
[Elmo, Bert]
[Bert, Elmo]
[Dora, Bert]
[Bert, Dora]
[Cisi, Bert]
[Bert, Cisi]
[Elmo, Ants]
[Ants, Elmo]
[Dora, Ants]
[Ants, Dora]
[Cisi, Ants]
[Ants, Cisi]
[Bert, Ants]
[Ants, Bert]
```


Amadeus - <noname32>*
Fail Toimeti Lisada Minna Teisendus Raam Erisoovid Abi
public class KMB_4»
klassi väljad
OLEKUD
konstruktor
public int[] next()»
doc Järjekordne kombinatsioon.
baasjuhud
for(;;)
switch (olek)


Amadeus - <noname28>*
Fail Toimeti Lisada Minna Teisendus Raam Erisoovid Abi
klassi väljad
private int n;»
käideldav hulk: arvud 0, 1, ..., n-1 (n elementi)
private int k; mitme kaupa
private KMB_4 gen_väike; generaator n-1, k-1 korral
private int[] väike; tulemus generaatorist n-1, k-1 korral
private int olek;»
private int[] pikendatud; jrk väljaantava kmb koht
klassi väljad
OLEKUD
static final int TEHA_VÄIKSEMATE_GENERAATOR = 0;»
static final int TEHA_UUS_VÄIKE = 1;»
static final int TEHA_UUS_PIKENDATUD = 2;»
static final int AMMENDATUD = 3;»
OLEKUD
konstruktor
public KMB_4(int n, int k)»
doc Kombinatsioonide genaator.
this.n = n;»
this.k = k;»
olek = TEHA_VÄIKSEMATE_GENERAATOR;»
konstruktor


Amadeus - <noname26>*
Fail Toimeti Lisada Minna Teisendus Raam Erisoovid Abi
?? switch (olek)»
case TEHA_VÄIKSEMATE_GENERAATOR :»
gen_väike = new KMB_4(n-1, k-1);»
case TEHA_UUS_VÄIKE :»
väike = gen_väike.next();»
if (väike == null)»
väiksemate pikendused on kõik juba välja antud
olek = AMMENDATUD;»
return null;»
pikendatud = new int[väike.length+1];»
» kopeerida väike ==> pikendatud algusesse:
System.arraycopy(väike, 0, pikendatud, 0, väike.length);»
seada viimane
if (väike.length == 0)»
pikendatud[pikendatud.length-1] = 0;»
else»
pikendatud[pikendatud.length-1] = pikendatud[pikendatud.le
seada viimane
olek = TEHA_UUS_PIKENDATUD;»
return pikendatud;»
case TEHA_UUS_PIKENDATUD:»
if (pikendatud[pikendatud.length-1]++ == n-1)»
olek = TEHA_UUS_VÄIKE;»
continue;»
return pikendatud;»
default :»
println("KMB_4:next(): Olek vale:"+olek);»
System.exit(0);»


Amadeus - <noname32>*
Fail Toimeti Lisada Minna Teisendus Raam Erisoovid Abi
public int[] next()»
doc Järjekordne kombinatsioon.
baasjuhud
if (olek == AMMENDATUD)»
return null;»
if (k == 0)»
olek = AMMENDATUD;»
return new int[0]; anda välja tühi massiiv
baasjuhud