

```
Amadeus - workspace\Tk_1_1_ret.algjava*
Fail Toimeti Lisada Minna Teisendus Raam Erisoovid Abi
projekt: AlgJavaWorks

TEST
»
int n = 7;»
println("Astmete arv n = " + n + "\n-----");»
int kokku = trepp(n, 0, "");»
println("-----\nKokku võimalusi: " + kokku);»

TEST
»
int trepp(int n, int loe, String s)»
»
» Antud: n -- trepi astmete koguarv
» .   loe -- senini ületatud astmete arv ühel käigul
» .   s   -- senini ületatud astmete ületamismoodus:
» .           sõne, mis koosneb sümbolitest 1 ja 2,
» .           1, kui mindud on järgmisele astmele,
» .           2, kui mindud on ülejärgmisele astmele
» Tulemus: printitakse kõik kogu trepi ületamismoodused (s),
» .       tagastatakse ületamismooduste (printimiste) arv

    if (loe > n)»
    »
    » return 0;»

    if (loe == n)»
    »
    » println(s);»
    » return 1;»

    return trepp(n, loe + 2, s + "2") + trepp(n, loe + 1, s + "1");»
```

```
Amadeus - workspace\Tk_1_1_fibo2.al...
Fail Toimeti Lisada Minna Teisendus Raam Erisoovid Abi
projekt: AlgJavaWorks

TEST
»
int[] fibo = {1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233,
377, 610, 987, 1597, 2584, 4181, 6765, 10946, 17711,
28657, 46368, 75025, 121393, 196418, 317811};»
int n = 1; peab olema: 0 < n < 28
println("Astmete arv n = " + n + "\n-----");»
trepp(n, "");»
println("-----\nKokku võimalusi: " + fibo[n]);»
»

TEST
»
void trepp(int n, String s)»
»
» Antud: n -- trepi astmete koguarv
» .   s   -- senini ületatud astmete ületamismoodus:
» .           sõne, mis koosneb sümbolitest 1 ja 0,
» .           1, kui sellele astmele on astutud,
» .           0, kui see aste on vahele jäetud
» Tulemus: printitud kõik kogu trepi ületamismoodused

    if (s.length() == n)»
    »
    » println(s.replaceAll("01", "2"));»

    else if (s.length() < n)»
    »
    » trepp(n, s + "1");»
    » trepp(n, s + "01");»
```

```
Amadeus - workspace\Tk_1_00.algjava*
Fail Toimeti Lisada Minna Teisendus Raam Erisoovid Abi
projekt: AlgJavaWorks

import java.util.ArrayList;»
TEST
ArrayList<String> ast(int n)»
    spetsifikatsioon
    ArrayList<String> tulem = new ArrayList<String>();»
    baasjuht
    if (n == 2)»
        teisele astmele ülesminekumooduseid on 2:
        tulem.add("11");»
        tulem.add("2");»
    return tulem;»
    baasjuht
    ArrayList<String> vtulem = ast(n-1); rekursioon
    » vtulem on kõik astmele n-1 ülesminekumoodused
    * for(String v : vtulem)»
        » v on üks astmele n-1 ülesminekumoodus;
        » jätkata moodust v:
        tulem.add(v+"1"); n-ndale astmele saab astuda ühiksammuga
        if (v.endsWith("1")) kui astmele n-1 tuldud ühiksammuga
        siis saab n-ndale astet n-1 vahele jättes
        » viimane v sümbol ("1") asendada sümboliga "2":
        v = v.substring(0, v.length()-1) + "2";»
        tulem.add(v);»
    return tulem;»
```

```
Amadeus - workspace\Tk_1_00.algjava*
Fail Toimeti Lisada Minna Teisendus Raam Erisoovid Abi
projekt: AlgJavaWorks

spetsifikatsioon
»
» Antud: trepi astmete arv n, n > 1
» Tulemus: tagastatakse sõnede hulk, milles
» . iga sõne on üks ülesminekumoodus,
» . koosnedes sümbolitest 1 ja 2;
» . 1 -- tehtud on samm järgmisele astmele,
» . 2 -- tehtud on samm ülejärgmisele astmele;
» . näiteks n = 3 korral on tulemuseks sõnede hulk
» . 111 (ühtegi pole vahele jäetud)
» . 12 (teine on vahele jäetud)
» . 21 (esimene on vahele jäetud)
spetsifikatsioon
```

```
Amadeus - workspace\Tk_1_00.algjava*
Fail Toimeti Lisada Minna Teisendus Raam Erisoovid Abi
projekt: AlgJavaWorks

TEST
»
int n = 7;»
println("Astmete arv n = " + n + "\n-----");»
ArrayList<String> moodused = ast(n);»
* for(String moodus : moodused)»
    println(moodus);»

// ehk: moodused.forEach(moodus -> System.out.println(moodus));»
println("-----\nKokku võimalusi: " + moodused.size());»
TEST
```

```
Amadeus - workspace\Tk_1_0.algjava*
Fail Toimeti Lisada Minna Teisendus Raam Erisoovid Abi
projekt: AlgJavaWorks

import java.util.HashSet;»
TEST
HashSet<String> ast(int n)»
spetsifikatsioon
HashSet<String> tulem = new HashSet<String>();»
baasjuht
if (n == 2)»
teisele astmele ülesminekumooduseid on 2:
tulem.add("11");»
tulem.add("01");»
return tulem;»
baasjuht
HashSet<String> vtulem = ast(n-1); rekursioon
» vtulem on astmele n-1 ülesminekumoodused
* for(String v : vtulem)»
» v on üks astmele n-1 ülesminekumoodus
tulem.add(v+"1"); igal juhul saab astuda n-ndale astmele
if (v.endsWith("11")) kui astmetele n-2 ja n-1 on astunud
siis saab n-ndale astet n-1 vahele jättes
» v = ".....11"
v = v.substring(0, v.length()-1) + "0";»
» v = ".....10"
tulem.add(v+"1");»
return tulem;»
```

```
Amadeus - workspace\Tk_1_0.algjava*
Fail Toimeti Lisada Minna Teisendus Raam Erisoovid Abi
projekt: AlgJavaWorks

spetsifikatsioon
»
» Antud: trepi astmete arv n, n > 1
» Tulemus: tagastatakse n-sümboliliste sõnade hulk, milles
» . iga sõne koosneb sümbolitest 1 ja 0
» . 1 -- kui vastavale astmele on astunud,
» . 0 -- kui vastav aste on vahele jäetud;
» . näiteks n = 3 korral on tulemuseks sõnade hulk
» . 011 (jõutakse viimasele, vastusesse: 21)
» . 111 (jõutakse viimasele, vastusesse: 111)
» . 101 (jõutakse viimasele, vastusesse: 12)
spetsifikatsioon
```

```
Amadeus - workspace\Tk_1_0.algjava*
Fail Toimeti Lisada Minna Teisendus Raam Erisoovid Abi
projekt: AlgJavaWorks

TEST
»
int n = 7;»
println("Astmete arv n = " + n + "\n-----");»
int loe = 0;»
* for(String s : ast(n))»
loe++;»
s = s.replaceAll("01", "2"); paarid 01 asendada arvuga 2
println(s);»

println("-----\nVõimaluste arv: " + loe);»
TEST
```

Mitterekursiivne

```
Amadeus - workspace\Tk_1_1_mitterek.algjava*
Fail Toimeti Lisada Minna Teisendus Raam Erisoovid Abi
projekt: AlgJavaWorks

import java.util.ArrayList;

TEST

ArrayList<String> astumised(int n)»
» Antud: trepi astmete arv n, n > 1
» Tulemus: tagastatakse sellest trepist ülesmineku mooduste list,
» . mis koosneb sümbolitest 1 ja 2 (mitu astet korraga võetud)
ArrayList<String> eelmisele = new ArrayList<String>(); astmele i-2 jõudmise moodused
ArrayList<String> sellele = new ArrayList<String>(); astmele i-1 jõudmise moodused
» tsükli algatuseks (juhaks kui i = 3):
eelmisele.add("1");»
sellele.add("11");»
sellele.add("2");»

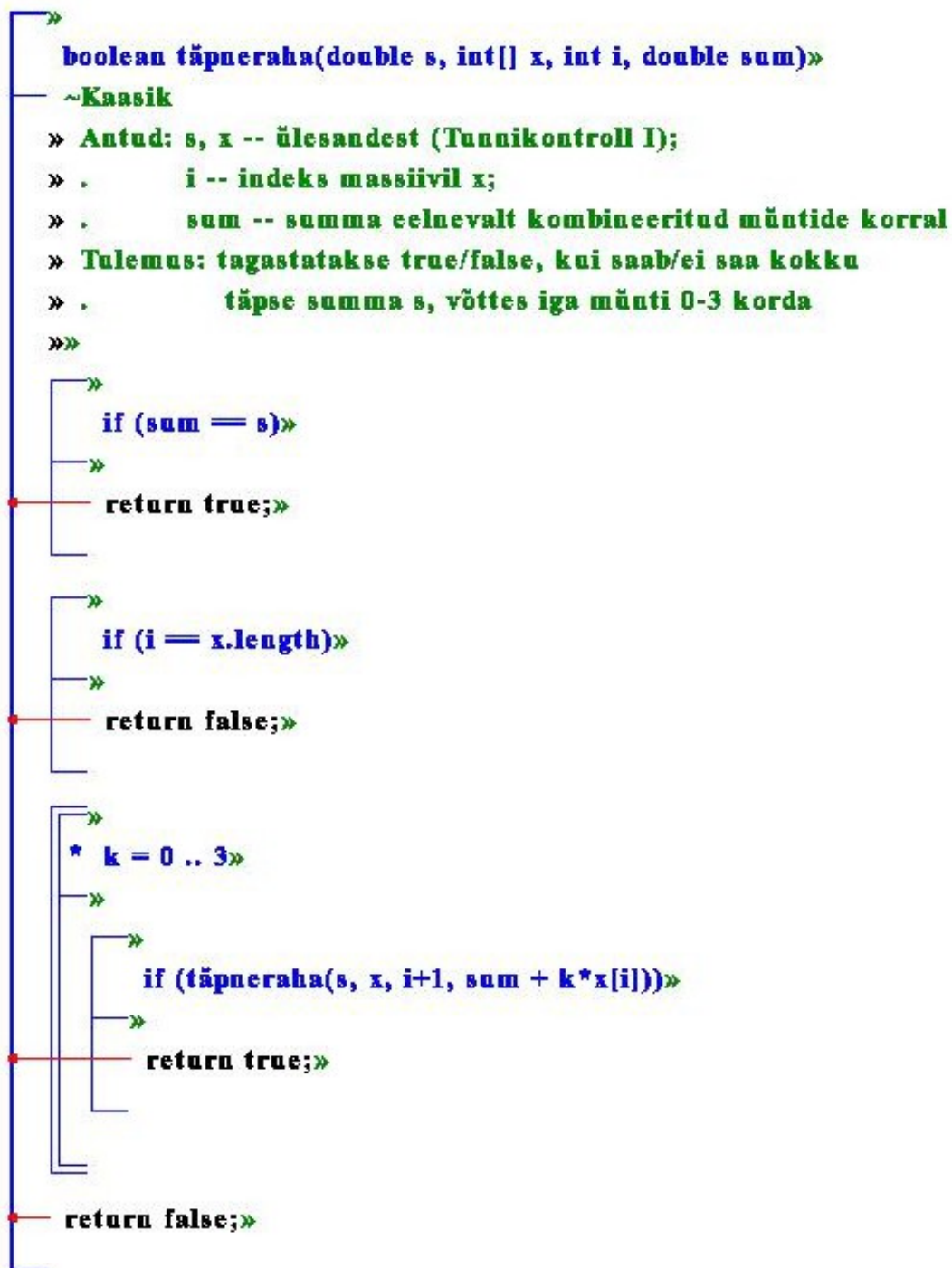
+ i = 3 .. n»
ArrayList<String> järgmisele = new ArrayList<String>();»
+ for(String p : eelmisele)»
+   järgmisele.add(p + "2");»
+ for(String p : sellele)»
+   järgmisele.add(p + "1");»
eelmisele = sellele;»
sellele = järgmisele;»

return sellele;»
```

```
Amadeus - workspace\Tk_1_1_mitterek...
Fail Toimeti Lisada Minna Teisendus Raam Erisoovid Abi
projekt: AlgJavaWorks

TEST
»
int n = 7;»
println("Astmete arv n = " + n);»
println("-----");»
ArrayList<String> see = astumised(n);»
»
+ for(String p: see)»
+   println(p);»

println("-----");»
println("Kokku võimalusi: " + see.size());»
TEST
```



```
boolean raha_klapib(Integer[] x, int s)»
```

```
» Antud: x -- müntide vääringud, mida igaüht 3 tk,
```

```
» .      s -- poes nõutav summa
```

```
» Tulemus: tagastatakse true,
```

```
» .      kui nende müntidega saab kokku täpselt summa s;
```

```
» .      false vastasel korral
```

```
int n = x.length * 3;»
```

```
int[] a = new int[n]; siia - kolmekordistatud x
```

```
int j = 0;»
```

```
for (i = 0 .. x.length-1
```

```
int N = (int)Math.pow(2, n);»
```

```
* mask = 0 .. N-1»
```

```
int sum = 0;»
```

```
for (a elementide valimine maski järgi, summeerimine
```

```
* k = 0 .. n-1 k: indeks originaalil a
```

```
if ( ((mask >> (n-k-1)) % 2) == 1) kas valida
```

```
sum += a[k];»
```

```
for (a elementide valimine maski järgi, summeerimine
```

```
if (sum == s)
```

```
return false;»
```