Praktikum 7

Kahendpuud

- **Abivahendid kahendpuu loomiseks, kuvamiseks ja töötlemiseks**

- **Harjutusülesanded**

  KP_1. Täieliku kahendpuu valmistamine
  KP_2. Kahendpuu kõrgus
  KP_3. Tasakaalu kontrollimine
  KP_4. Kahendpuu ja puu läbimine
  KP_5. Aritmeetilise avaldise puu
  KP_6. Tipu järglaste koguarv
  KP_7. Tippudele kõrguse- ja tasemeväljad
  KP_8. Tasemete generaator
  KP_9. Kompaktse kahendpuu valmistamine
  KP_10. Kahendpuu peegeldus

Abivahendid kahendpuu loomiseks, kuvamiseks ja töötlemiseks

Pakett *ee.ut.kiho.aa.graaf* **(** `import ee.ut.kiho.aa.graaf.*;` **)**

sisaldab klasside

Graaf
GraafiJoonistaja
Kaar
Kahendpuu
Puu
Tipp

kirjeldusi.

Kahendpuud käsitletakse kui graafi/puu erijuhtu. Seetõttu on rakendatavad graafi- ja puutöötlemise meetodid klassidest *Graaf/Puu*. Kahendpuu/puu juureks on graafi esimene tipp (tipp indeksiga 0).

Vasak-parem alluvussuhe on määratud tippude *x*-koordinaatidega:
kui tipp *t* on tipu *u* alluv (st on olemas kaar *u*-->*t*), siis
t on tipu u vasak alluv, kui *t.x* < *u.x*,
t on tipu u parem alluv, kui *t.x* > *u.x.*

n-tipulise juhu-kahendpuu *kp* valmistamine:

```
Kahendpuu kp = new Kahendpuu(Puu.juhupuu(n, 2));
```

Harjutusülesanded

KP_1. Täieliku kahendpuu valmistamine [Õpik, lk 28]

Kirjutada ja testida rekursiivne meetod järgmise ülesande lahendamiseks.
Antud: arv *m* – loodava kahendpuu tasemete arv.
Tulemus: täielik *m*-tasemeline kahendpuu.

Java

Ühetipulise (ainult juur) kahendpuu *kp* tegemine:

```
Graaf g =  Graaf();          -- tühi graaf
g.lisada(new Tipp());        -- lisada üks tühi tipp
Kahendpuu kp = new Kahendpuu(g);
```

Kahendpuus *kp* alluvateta tipule *t* mõlema alluva lisamine:

```
Tipp uus = new Tipp();                  --  tühi tipp
kp.lisadaAlluv(t, uus, true);   --  true: lisamine vasakuks alluvaks
uus = new Tipp();
kp.lisadaAlluv(t, uus, False)   --  false: lisamine paremaks alluvaks
```

Ehk lühemalt:

```
kp.lisadaAlluv(t, new Tipp(), true);
kp.lisadaAlluv(t, new Tipp(), false);
```

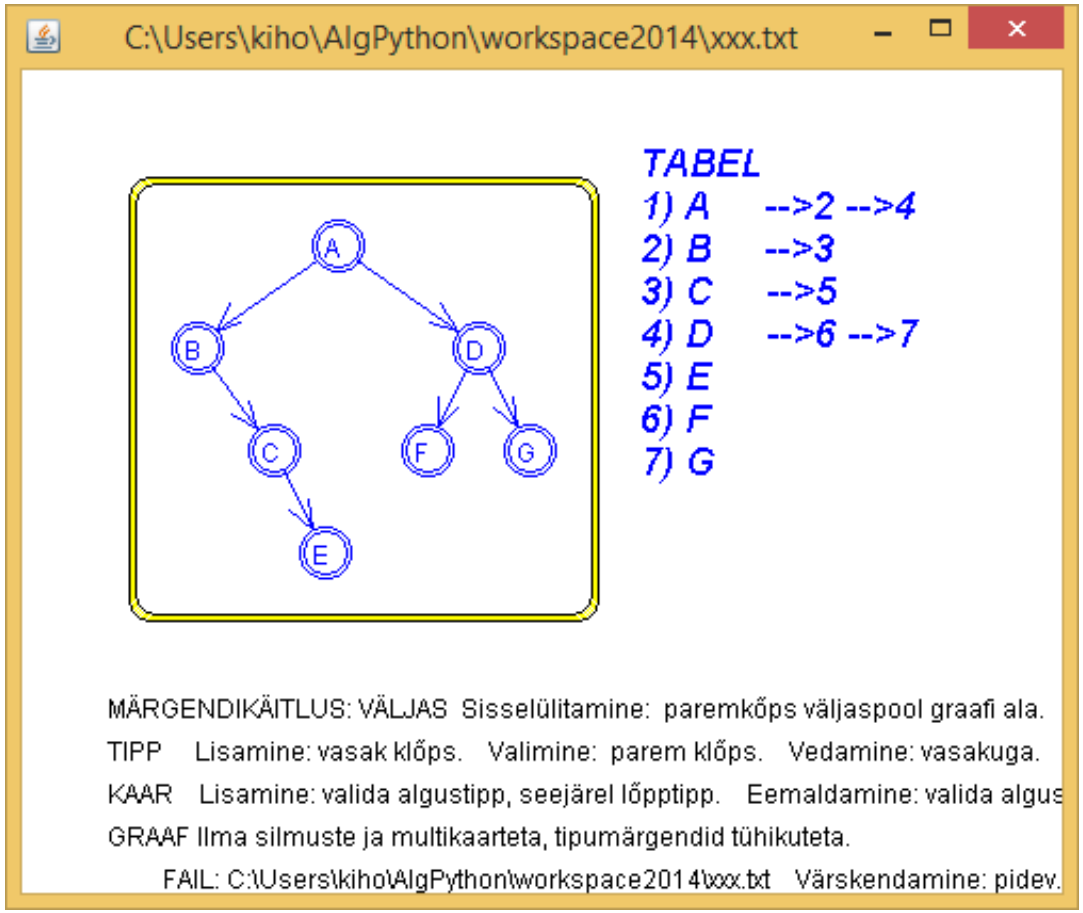
Kahendpuu (ka graafi ja puu) kuvamine:

```
kp.väljastadaTabelina("fail.txt"); -- kahendpuu faili (tabelkujul)
new GraafiJoonistaja("fail.txt", true);-- tippudes märgendid
new GraafiJoonistaja("fail.txt"); -- tippudes numbrid
```

KP_2. Kahendpuu kõrgus

Kirjutada ja testida rekursiivne meetod järgmise ülesande lahendamiseks.
Antud: kahendpuu ja selle tipp.
Tulemus: antud tipust algava (alam)kahendpuu kõrgus.

Testi tulemuse näide




```
 Alampuu
juur kõrgus
 A      4
 B      3
 C      2
 D      2
 E      1
 F      1
 G      1
```

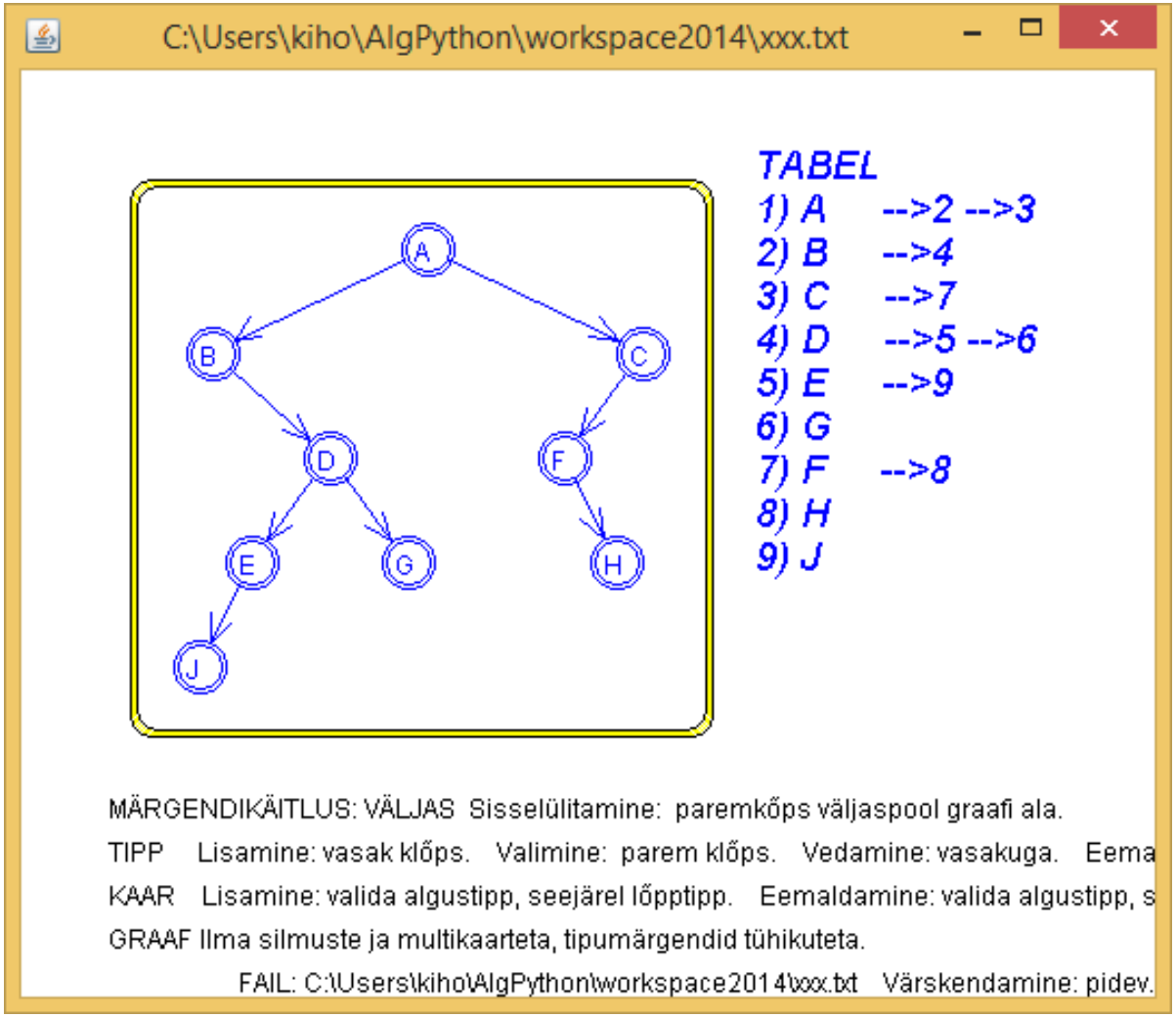
KP_3. Tasakaalu kontrollimine

Kirjutada ja testida meetod järgmise ülesande lahendamiseks.

Antud: kahendpuu.

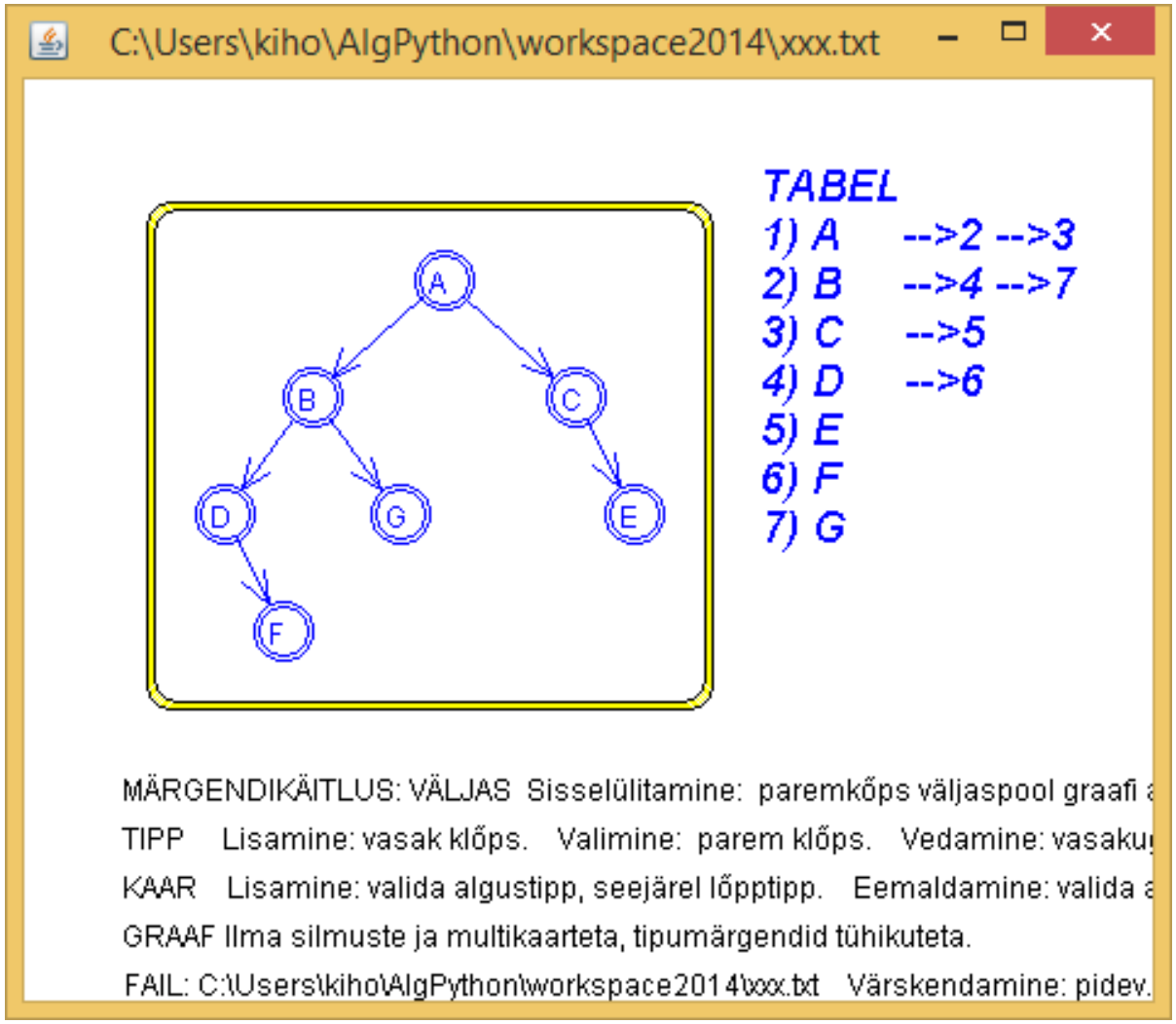
Tulemus: antud kahendpuu nende tippude järjend (listina), kus tasakaal on rikutud, st tipu alluvate kõrgus erineb rohkem kui 1 võrra.

Testi tulemuse näide




```
Tasakaal rikutud
tipus B
tipus C
```

Testi tulemuse näide



On tasakaalus.

KP_4. Kahendpuu ja puu läbimine

KP_4-1. Kahendpuu läbimine [Õpik, lk 29]
Kirjutada ja testida rekursiivsed meetodid järgmiste ülesannete lahendamiseks

KP_4-1E:
Antud: kahendpuu ja selle tipp *t.*
Tulemus: tipust *t* algava kahendpuu tipud eesjärjestuses (listina).

KP_4-1K:
Antud: kahendpuu ja selle tipp *t.*
Tulemus: tipust *t* algava kahendpuu tipud keskjärjestuses (listina).

KP_4-1L:
Antud: kahendpuu ja selle tipp *t.*
Tulemus: tipust *t* algava kahendpuu tipud lõppjärjestuses (listina).

Java

Tagastada (listina) tipust *t* algava *kp* alamkahendpuu tipud eesjärjestuses:

```
ArrayList<Tipp> eesjrst(Kahendpuu kp, Tipp t){
    ArrayList<Tipp> tulem = new ArrayList<Tipp>();
    if (t == null) return tulem;
    tulem.add(t);
    tulem.addAll(eesjrst(kp, kp.vasakAlluv(t)));
    tulem.addAll(eesjrst(kp, kp.paremAlluv(t)));
    return tulem;
 }
```

KP_4-2. Puu läbimine

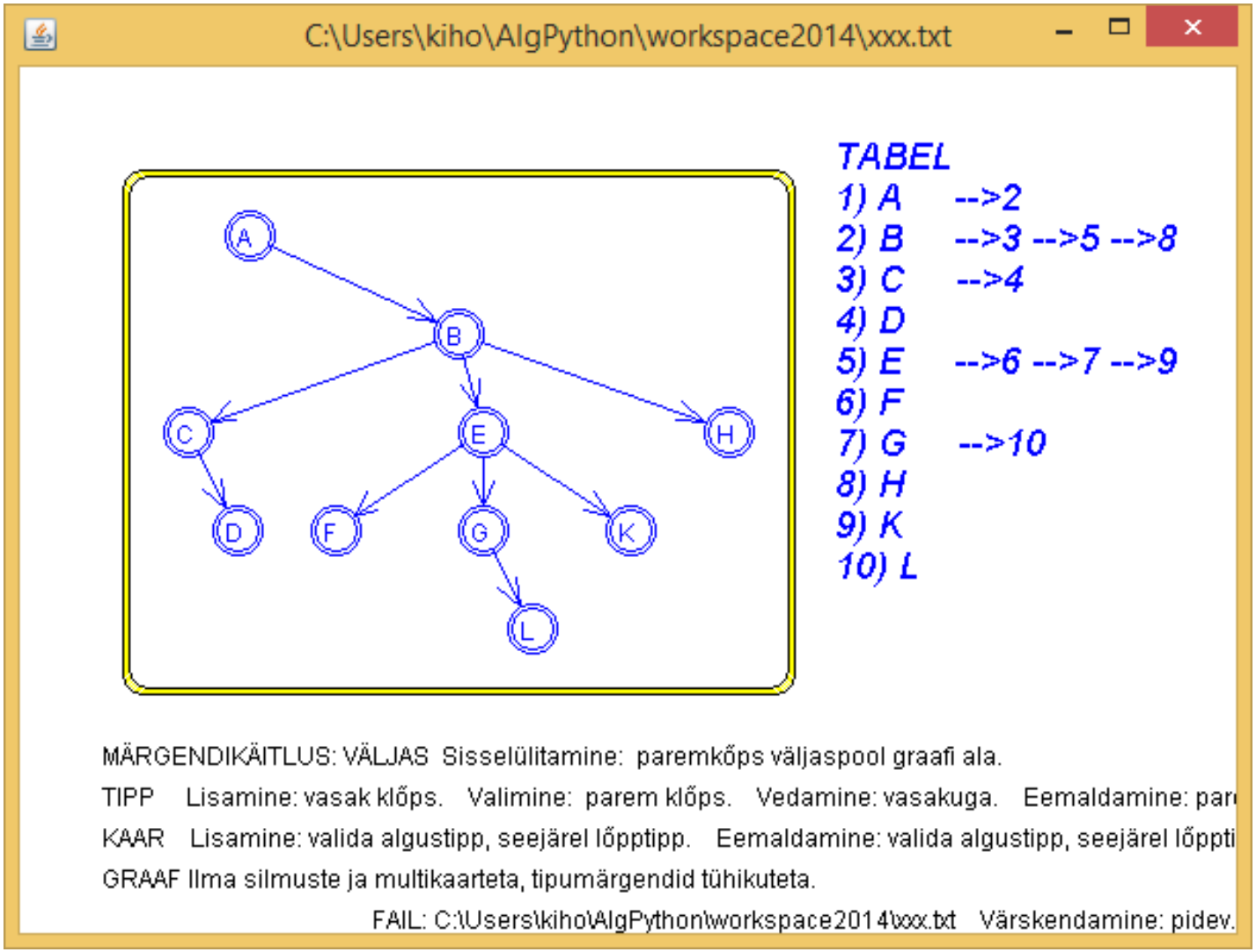
Kirjutada ja testida rekursiivsed funktsioonid järgmiste ülesannete lahendamiseks.

Antud: puu ja selle tipp *t.*.

Tulemus: tipust *t* algava (alam)puu tipud eesjärjestuses (listina).

Antud: puu ja selle tipp *t.*

Tulemus: tipust *t* algava (alam)puu tipud lõppjärjestuses (listina).



Testi tulemuse näide

```
Puu eesjrst:
ABCDEFGLKH

Puu lõppjrst:
DCFLGKEHBA
```

KP_5. Aritmeetilise avaldise puu

Koostada programm, mis valmistab (juhusliku) aritmeetilise avaldise kahendpuu ja arvutab vastava avaldise väärtuse. (Vt ka järgmine slaid.)

Eraldi meetoditena kirjeldada järgmised alamülesanded:

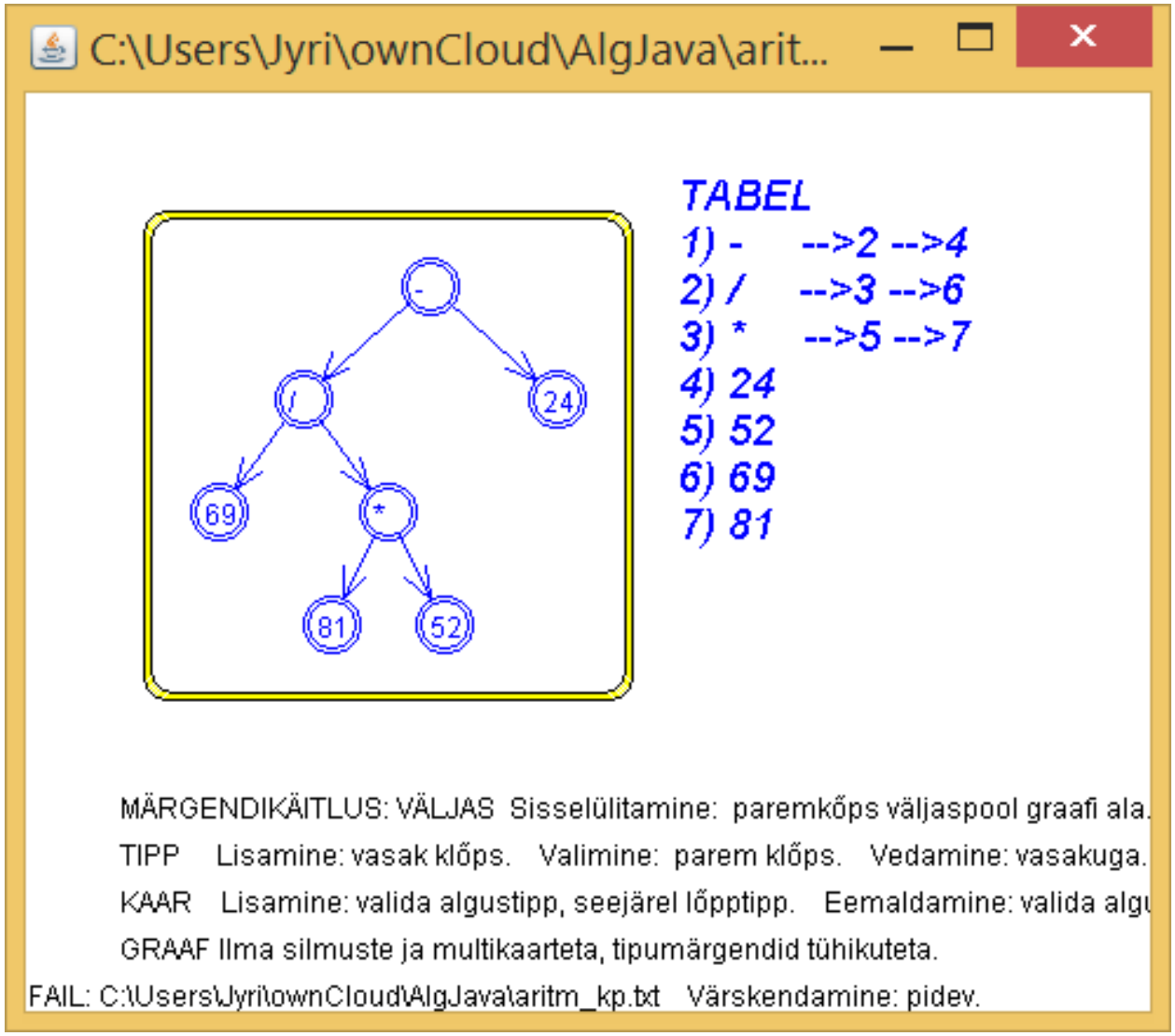
KP_5-1. Aritmeetilise avaldise puu valmistamine

Kirjutada ja testida funktsioon järgmise ülesande lahendamiseks.
Antud: kahendpuu.
Tulemus: antud kahendpuu muudetud 0-2 kahendpuuks (igale ühe alluvaga tipule lisatud teine, puudunud alluv), seejärel
rippuvate e. lehttippude märgenditeks seatud juhuarvud,
ülejäänutele – binaarsete tehete märgid, juhuvalik (+ - * /) seast.

KP_5-2. Aritmeetilise avaldise puust väärtuse arvutamine

Kirjutada ja testida rekursiivne funktsioon järgmise ülesande lahendamiseks.
Antud: aritmeetiline kahendpuu ja selle tipp *t.*
Tulemus: tipust *t* algava aritmeetilisest (alam)puust arvutatud väärtus.

Aritmeetilise avaldise kahendpuu (KP_5 testi tulemuse) näide.



Avaldise väärtus: -23.98.

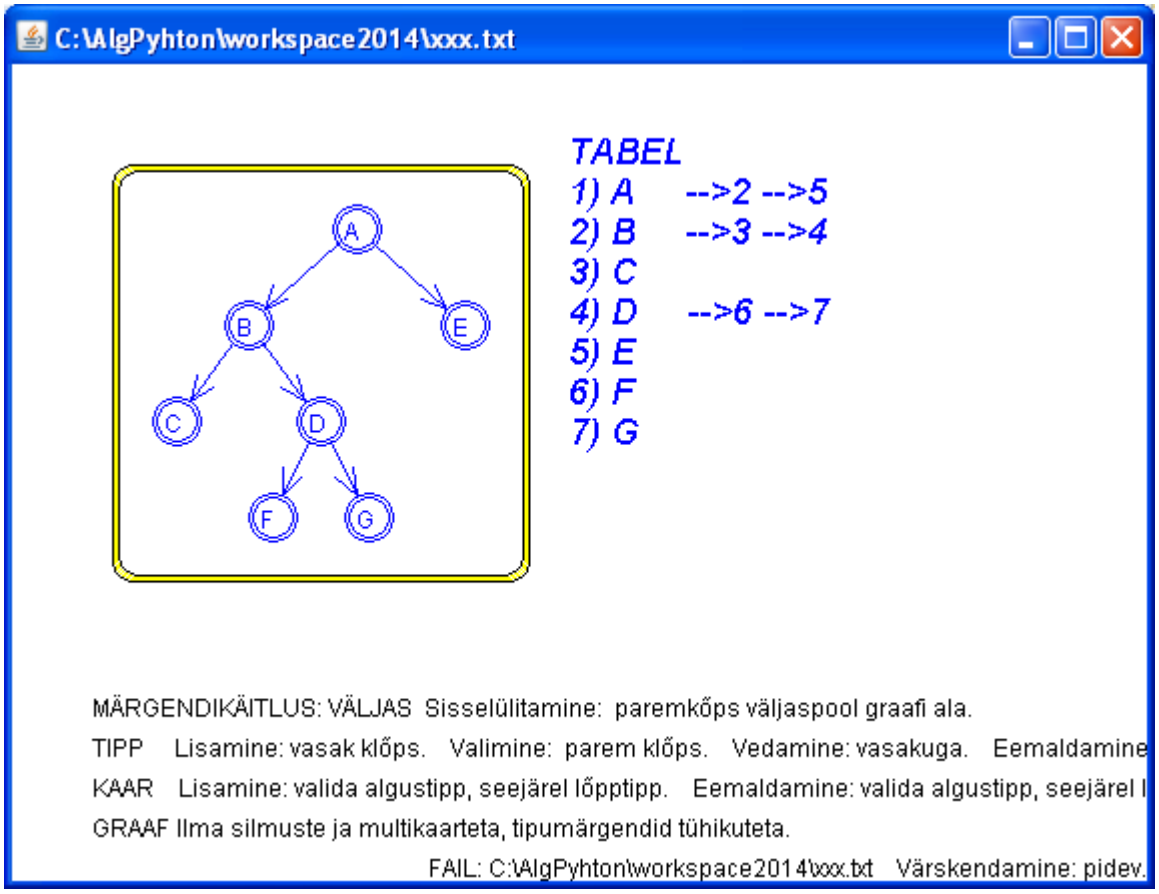
KP_6. Tipu järglaste koguarv

Kirjutada ja testida rekursiivne meetod järgmise ülesande lahendamiseks.

Antud: kahendpuu ja selle tipp.

Tulemus: antud tipu kõigi (vahetute ja kaugemate) järglaste koguarv antud kahendpuus.

Testi tulemuse näide



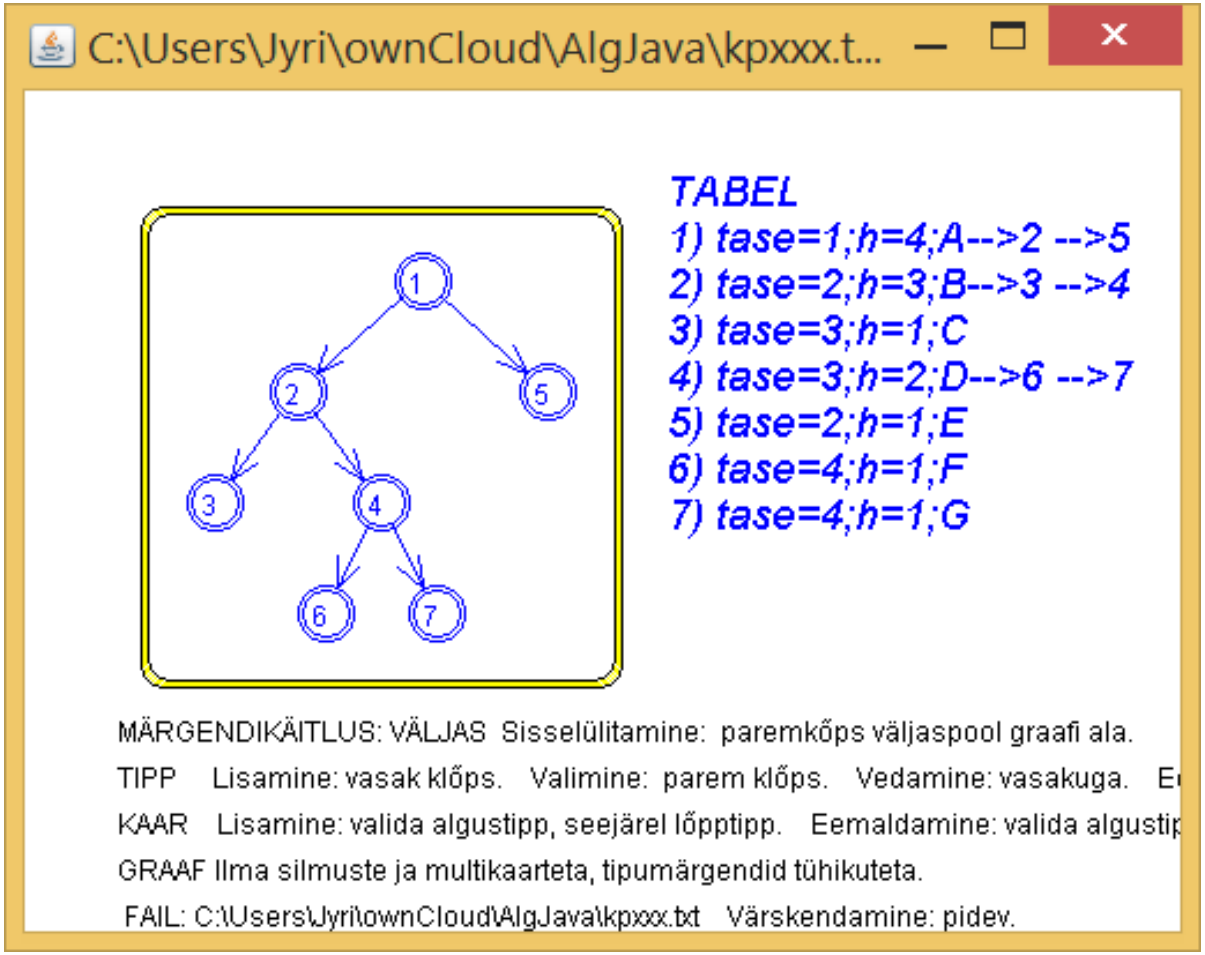
Tipp	Järglasi
A	6
B	4
C	0
D	2
E	0
F	0
G	0

KP_7. Tippudele kõrguse- ja tasemeväljad

KP_7-1. Kirjutada ja testida rekursiivne meetod järgmise ülesande lahendamiseks.
Antud: kahendpuu *kp*, selle tipp *t.*
Tulemus: tipust *t* algavas *kp* alampuus on iga tipu *u* väljale "h" omistatud tipust *u* algava alampuu kõrgus; tagastatakse tipust *t* algava alampuu kõrgus.

KP_7-2. Kirjutada ja testida rekursiivne meetod järgmise ülesande lahendamiseks.
Antud: kahendpuu *kp*, selle tipp *t* ja tipu *t* tasemenumber *m.*
Tulemus: tipust *t* algavas *kp* alampuus on iga tipu väljale "tase" omistatud tasemenumber; antud tipu tasemenumbriks on *m*, selle alluvatel *m*+1 jne.

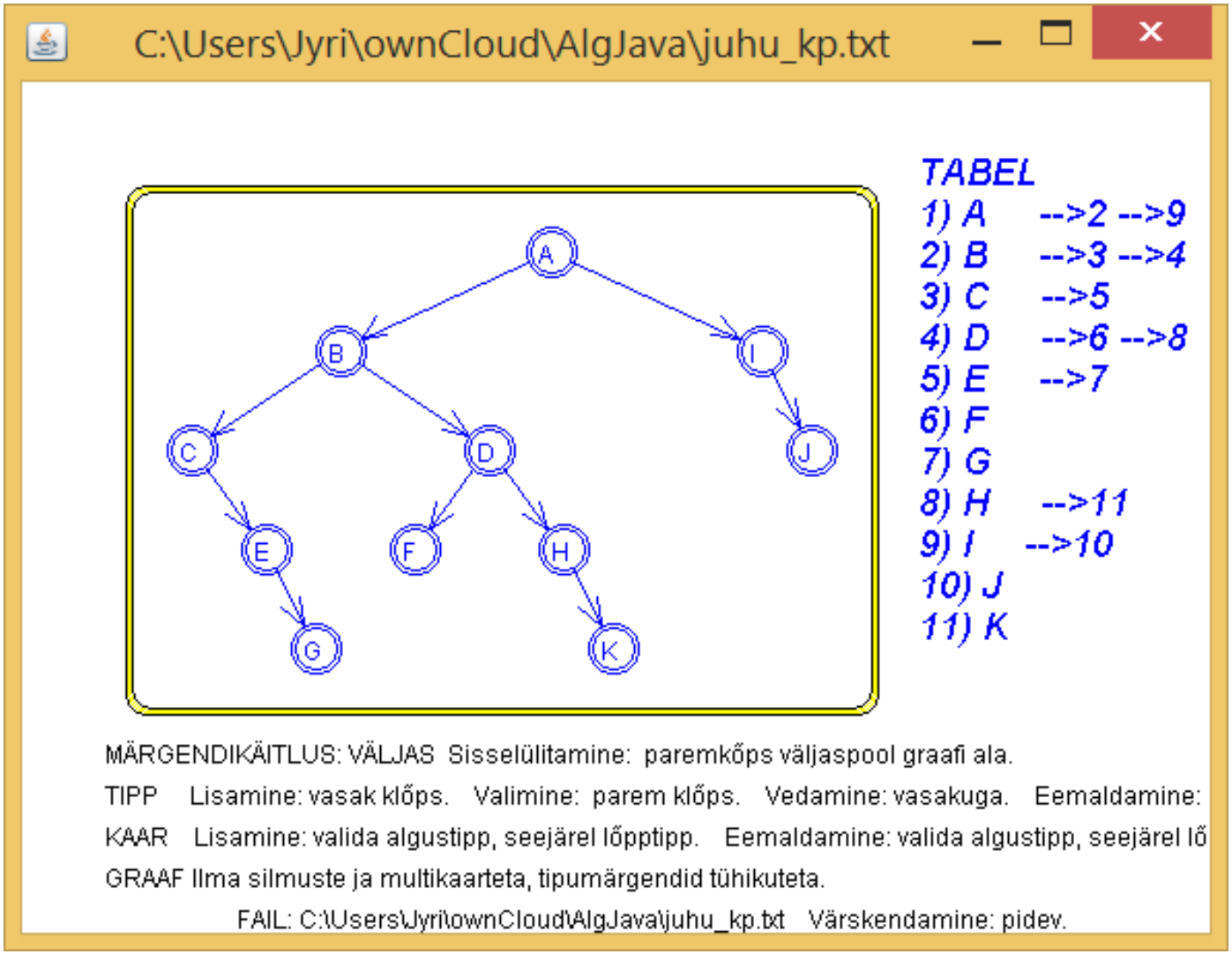
KP_7 testi tulemuse näide



KP_8. Tasemete generaator

Kirjutada ja testida tasemete generaator genereerimaks järjest antud kahendpuu tasemeid (tippude loeteludena).

Testi tulemuse näide




```
Tipumärgendid
tasemeti:
A
B I
C D J
E F H
G K
```

KP_9. Kompaktse kahendpuu valmistamine [Õpik, lk 28]
Kirjutada ja testida meetod järgmise ülesande lahendamiseks.
Antud: arv *n* – loodava kahendpuu tippude arv.
Tulemus: kompaktne *n*-tipuline kahendpuu.

Meetodi üldskeem:

Valmistada (KP_1) täielik kahendpuu, milles on $\log_2(n+1)$ taset.
Leida (KP_8) viimase taseme tipud ja lisada neile alluvateks veel vajalik arv uusi tippe.

KP_10. Kahendpuu peegeldus

Kirjutada ja testida meetod järgmise ülesande lahendamiseks.
Antud: kahendpuu.
Tulemus: antud kahendpuu peegeldus üle vertikaaltelje.

Idee: tippude *x*-koordinaadid asendada vastandarvudega.

Testi tulemuse näide