

Ülesanne 1

Seikluspargis koosneb rada rippildadest. Selles ülesandes on raja läbimist simuleeritud.

```
public class Rippsild {
    private char sillaTähis;
    private int läbimiseAeg;
    public Rippsild(char sillaTähis, int läbimiseAeg) {
        this.läbimiseAeg = läbimiseAeg;
        this.sillaTähis = sillaTähis;
    }
    public void ületa(Seikleja seikleja) throws InterruptedException {
// public synchronized void ületa(Seikleja seikleja) throws InterruptedException {
        System.out.println(seikleja.getNimi() + " astus sillale " + sillaTähis + " "
            + ((System.currentTimeMillis() - seikleja.getAlgusAeg()) / 1000));
        Thread.sleep(1000 * läbimiseAeg);
        System.out.println(seikleja.getNimi() + " lahkus sillalt " + sillaTähis + " "
            + (System.currentTimeMillis() - seikleja.getAlgusAeg()) / 1000);
    }
}

public class Seikleja implements Runnable {
    private String nimi;
    private boolean kasAlgusest;
    private Rippsild[] rippisillad;
    private long algusAeg;
    public Seikleja(String nimi, Rippsild[] rippisillad, boolean kasAlgusest) {
        this.nimi = nimi;
        this.rippisillad = rippisillad;
        this.kasAlgusest = kasAlgusest;
    }
    public String getNimi() {
        return nimi;
    }
    public long getAlgusAeg() {
        return algusAeg;
    }
    public void run() {
        this.algusAeg = System.currentTimeMillis();
        try {
            if (kasAlgusest) {
                for (int i = 0; i < rippisillad.length; i++)
                    rippisillad[i].ületa(this);
            } else {
                for (int i = rippisillad.length - 1; i >= 0; i--)
                    rippisillad[i].ületa(this);
            }
        } catch (InterruptedException e) {
            throw new RuntimeException(e);
        }
    }
}
```

Peaklass on järgmine.

```
public class Seikluspark {
    public static void main(String[] args) {
        Rippsild[] sillad = {new Rippsild('A', 10), new Rippsild('B', 20)};
        Seikleja seikleja1 = new Seikleja("Rasmus", sillad, true);
        Seikleja seikleja2 = new Seikleja("Grete", sillad, false);
        Thread t1 = new Thread(seikleja1);
        Thread t2 = new Thread(seikleja2);
        t1.start();
        t2.start();
    }
}
```

Ülesanne 1.1

Mis ilmub ekraanile. Miks?

Ülesanne 1.2

Mis ilmuks ekraanile, kui meetod ületa ees oleks lisaks võtmesõna **synchronized**? Miks?

Teavet klassi `Thread` kohta

```
static void sleep(long millis)
```

Causes the currently executing thread to sleep (temporarily cease execution) for the specified number of milliseconds, subject to the precision and accuracy of system timers and schedulers.

Teavet klassi `System` kohta

```
static long currentTimeMillis()
```

Returns the current time in milliseconds.

Ülesanne 2

Taheti koostada programm, mis simuleeriks laste mängimist mänguasjadega mängutoas. Programm pidi tagama järgmiste tingimuste kehtimise:

- Mängutoas saab korraga mitu last mängida.
- Laps saab korraga võtta mänguasjariiulist ainult ühe mänguasja.
- Konkreetse mänguasjaga saab korraga mängida vaid üks laps.
- Laps eelistab alati mänguasja, millel on kõige pikem nimi.
- Pärast mänguasjaga mängimist paneb laps selle riiulile tagasi.

Paraku see päriselt **ei õnnestunud**. Märkida järgnevast loetelust, mis põhjus(t)el see programm korrektselt ei tööta. Selgitada iga vastusevariandi korral, kuivõrd see võib olla põhjuseks.

1. Laps ei saa kasutada mängutuba, sest tema lõim võib enne tööd alustada, kui Mängutuba on loodud.
2. Meetodit `run` klassist `Laps` ei ole üldse välja kutsutud.
3. Meetod `leiaMänguasi` klassis `Mängutuba` peab olema sünkroniseeritud.
4. Meetod `tagastaMänguasi` klassis `Mängutuba` peab olema sünkroniseeritud.
5. Meetod `run` klassis `Laps` peab olema sünkroniseeritud.
6. Klassi `Laps` isenditele ei saa üldse konstruktori parameetreid anda, sest `Laps` realiseerib liidest `Runnable`.
7. Teine laps ei saa kunagi mängimist alustada, sest esimese lapse tegevus on lõpmatus tsükliis.
8. Isend `laps1` on loodud valesti, talle tüübiks on pandud `Runnable`, seda ei tohi teha.
9. Isend `laps2` on loodud valesti, talle tüübiks on pandud `Laps`, aga peab olema kindlasti `Runnable`.

```

public class Laps implements Runnable {
    private Mängutuba mængutuba;
    private String nimi;

    public Laps(Mängutuba mængutuba, String nimi) {
        this.mængutuba = mængutuba;
        this.nimi = nimi;
    }

    public void run() {
        while (true) {
            String mænguasi = mængutuba.leiaMænguasi();
            System.out.println(nimi + " mængib mænguasjaga " + mænguasi);
            mængutuba.tagastaMænguasi(mænguasi);
        }
    }
}

public class Mængutuba {
    private String[] riiul;

    public Mængutuba(String[] mænguasjad) {
        this.riiul = mænguasjad;
    }

    public String leiaMænguasi() {
        int parim = -1;
        for (int i = 0; i < riiul.length; i++) {
            if (riiul[i] != null
                && (parim == -1 || riiul[i].length() > riiul[parim].length())) {
                parim = i;
            }
        }
        String valik = riiul[parim];
        riiul[parim] = null;
        return valik;
    }

    public void tagastaMænguasi(String mænguasi) {
        for (int i = 0; i < riiul.length; i++) {
            if (riiul[i] == null) {
                riiul[i] = mænguasi;
                break;
            }
        }
    }

    public static void main(String[] args) {
        Mængutuba mængutuba = new Mængutuba(new String[]{"rong", "puzzle", "nukk"});
        Runnable laps1 = new Laps(mængutuba, "Volli");
        Laps laps2 = new Laps(mængutuba, "Martin");
        Thread t1 = new Thread(laps1);
        Thread t2 = new Thread(laps2);
        t1.start();
        t2.start();
    }
}

```