

Rehmaatika

Koostada rakendus arvutamise harjutamiseks. Kuvatakse tehteid liitmine-lahutamine ja kasutaja peab võimalikult kiiresti sisestama vastuseid.

Arvud (liidetavad) ja tehe tekitatakse juhuarvude abil etteantud piirides min...max. Et rakendus sobiks kasutamiseks algklassiõpilaste jaoks, ei tohiks lahutamistehe tulemus olla negatiivne arv.

Kasutaja peab saama valida taset, millest sõltub tehetes kasutatavate arvude suurus ja punktide arv ühe tehte kohta ning katse aega.

Lisada võib ka sobiva animatsiooni.

Tegevuste üldine skeem on järgnevalt toodud UML tegevusdiagrammidena ja ka algoritmikeeles.

Muutujad:

a ja **b** – juhuarvud vahemikus min...max,

c – a ja b summa (a + b)

tehe – juhuarv: 1 - liitmine, 2 - lahutamine

tulem – õige vastus: c (kui liitmine) või b (kui lahutamine)

vastus – kasutaja vastus

punkte – punktide jooksev ja lõplik summa

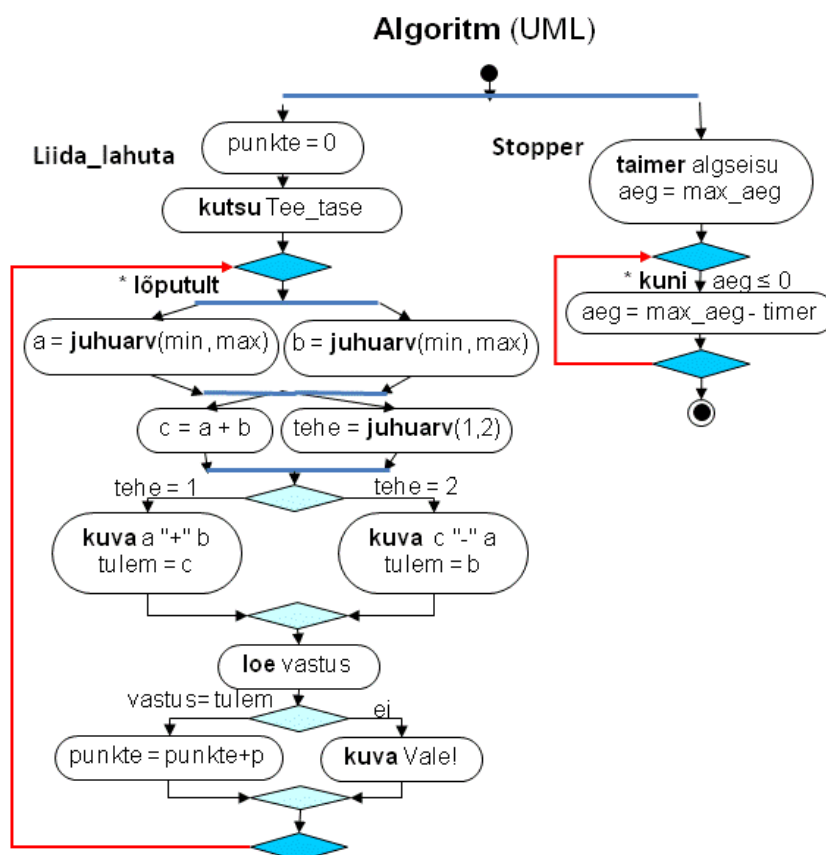
tase – ülesannete tase, võtame 3 taset

min ja **max** – min ja maks väärtus, sõltuvad tasemest näiteks: 1. tase: 1...10, 2. tase: 5...20, 3. tase: 10...50

p – punkte ühe tehte eest, sõltub tasemest, näiteks: 1. tase: p = 2, 2. tase: p = 5; 3. tase: p = 10

max_aeg – katse maksimaalne aeg, laseme valida kasutajal, näiteks vahemikus 20 ... 60 sek

aeg – jooksev aeg katse lõpuni: max_aeg – taimer



Algoritm (algoritmikeel)

protseduur Liida_lahuta

punkte = 0

kutsu Tee_tase

kordus lõputult

a = juhuarv(min, max)

b = juhuarv(min, max)

c = a + b

tehe = juhuarv(1, 2)

kui tehe = 1 **siis**

kuva a & "+" & b

tulem = c

muidu

kuva c & "-" & a

tulem = b

lõpp kui

loe vastus

kui vastus = tulem **siis**

punkte = punkte + p

muidu

kuva "Vale!"

lõpp kui

lõpp kordus

Rakendust võiks hakata ehitama järk-järgult:

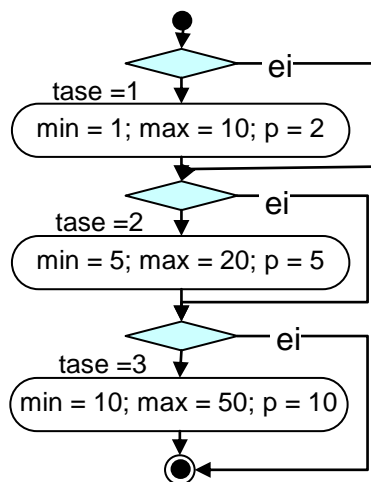
- tehte koostamine
- tehte vastuse küsimine, vastuse kontroll (esialgu ei pruugi punktiarvestust toimuda)
- raskusastme valik ning miinimumi, maksimumi, ühe tehte eest antavate punktide muutmine vastavalt valitud tasemele.

Realiseerimisel võib kasutada lisaks abimuutujaid näiteks küsimuse (tehte teksti) säilitamise jaoks.

Raskusastme kohandamine vastavalt eelnevalt määratud tasemele

Variant1

Tegevuskeem - UML



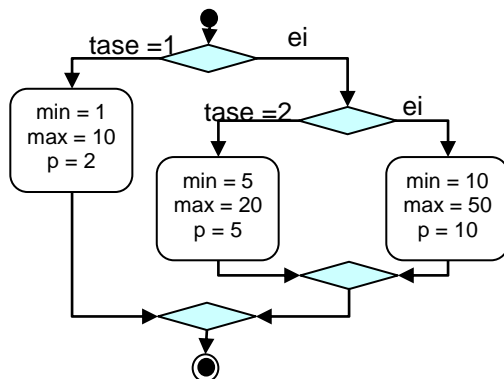
Algoritmikeel

protseduur Tee_Tase

```
kui tase = 1 siis
    min = 1; max = 10
    p = 2
kui tase = 2 siis
    min = 5; max = 20
    p = 5
kui tase = 3 siis
    min = 10; max = 50
    p = 10
```

Kõiki kolme tingimust kontrollitakse ka siis, kui tase on 1. Parem on variant 2.

Variant2



protseduur Tee_Tase

```
kui tase = 1 siis
    min = 1; max = 10; p = 2
muidu
    kui tase = 2 siis
        min = 5; max = 20; p = 5
    muidu
        min = 10; max = 50; p = 10
lõpp kui
lõpp kui
```

Lisaülesandeid:

- Loendada ka tehete ja õigete (ja valede) vastuste arv
- Väikseima ja suurima liidetava asemel võiks olla ette antud väikseim liidetav ja suurim summa.
- Lisada ka korrutamise ja jagamise tehted.
- Vale vastuse puhul võiks sama tehet korrata.
- Raskusaste (tase) võiks punktisumma kasvades suureneada automaatselt.