**Näppäimistön demoaminen**

**Tietorakenteet ja algoritmit**

Marja Jämbeck

Laura Pekkanen

Miikkael Puikkonen

Huhtikuu 2018

Tekniikan ala

Insinööri (AMK), tieto- ja viestintätekniikan koulutusohjelma

Mediatekniikka

Sisältö

[1 Johdanto 2](#_Toc510002938)

[2 Esittely 2](#_Toc510002939)

[3 Toteutus 2](#_Toc510002940)

[4 Testausaineiston generointi, testaus 3](#_Toc510002941)

[5 Yhteenveto 4](#_Toc510002942)

[Lähteet 5](#_Toc510002943)

[Liitteet 5](#_Toc510002944)

Kuviot

[Kuvio 1 Jonon toiminta (Häkkinen 2018, Tietorakenteet ja algoritmit luentomateriaali) 2](#_Toc510005289)

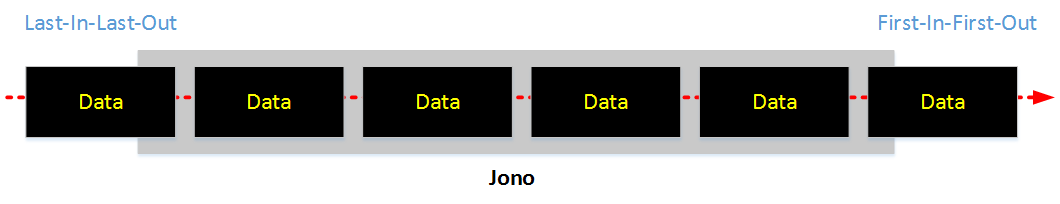
[Kuvio 2 Ohjelman toiminta 4](#_Toc510005290)

1. Johdanto

Harjoitustyön tarkoituksena oli esittää jonon toimintaa näppäimistön syötteen avulla. Jonon tilaa ja näppäinten painalluksia visualisoimme reaaliaikaisesti JavaScriptin avulla.

1. Esittely

Jono on abstrakti tietotyyppi. Sillä ei ole järjestystä, vaan se on auki molemmista päistä. Poistettaessa tieto lähtee joko jonon alusta eli first-in-first-out menetelmällä tai jonon lopusta eli last-in-last-out-menetelmällä. (Ks. Kuvio1).



Kuvio 1. Jonon toiminta (Häkkinen 2018, Abstraktit tietotyypit)

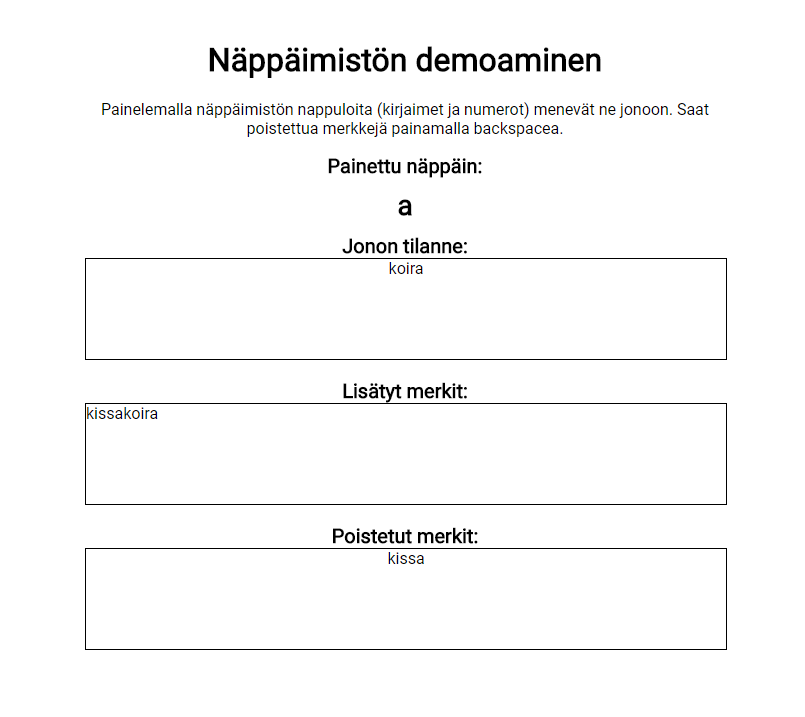
1. Toteutus

Harjoitustyössä toteutimme jonon implementoimalla JavaScriptillä. Jonon rakenne esitettiin usean taulukon eli arrayn avulla. Näppäillessä näppäimistöä merkit tallennettiin kahteen taulukkoon, ’lisatyt’ ja ’tilanne’ taulukkoihin. Näiden sisältö näytetään reaaliaikaisesti sivulla. Merkkejä voi poistaa painamalla backspace-näppäintä, jolloin ne lisätään ’poistetut’ tauluun ja samalla päivittyy ’tilanne’-kentän sisältö. Poiketen tavallisen jonon toiminnasta meidän jonollamme ei ole ollenkaan kokoa. Merkkejä poistuu vain niitä poistettaessa. Tilanteen päivittyessä käytetään shift()-metodia, joka poistaa taulukon ensimmäisen alkion. Kentän sisältö tyhjennetään ja sisältö haetaan uudestaan. Näppäimistön syöte on rajattu keycode-merkkien 46 ja 90 välille, eli syötteeseen on sallittu vain kaikki kirjaimet ja numerot muttei erikoisnäppäimiä.

Totesimme, että tehokkuuden mittaaminen on hankalaa ja turhaa. Nykyisissä tietokoneissa näppäimistön syötteellä ei ole huomattavaa merkitystä CPUn kuormituksen kannalta. Näppäimistön syötteen havainnoimiseen käytimme ensin jQueryn keypress()-metodia, mutta päädyimme vaihtamaan sen keyup()-metodiin, koska keypress huomioi myös erikoisnäppäimet ja ne näkyivät kentissä.

1. Testausaineiston generointi, testaus

Testasimme kirjoittaa ”kissakoira” ja poistaa jonosta sanan ”kissa”. Poistetut merkit näkyvät poistetut merkit kentässä ja jonon nykyinen tila on ”koira”. Tästä voimme todeta, että jono toimii halutulla tavalla (Ks. Kuvio2).



Kuvio 2. Ohjelman toiminta

1. Yhteenveto

Toimeksianto tuntui varsinkin aluksi haastavalta. Saimme kuitenkin visualisoitua näppäimistön toimintaa ja onnistuimme siinä omasta mielestämme hyvin. Harjoitustyöstä jäi kuitenkin pois kuormittavuuden visualisointi, mutta mielestämme se ei ollut harjoitustyölle oleellinen.

Lähteet

Häkkinen, A. Abstraktit tietotyypit. Viitattu 4.4.2018. <https://firesite.duckdns.org/index_TTZC0700.php>

Liitteet

Ohjelman koodi: <http://student.labranet.jamk.fi/~K8966/tietorakenne/>

index.html

<!DOCTYPE HTML>

<html>

<!--https://javascript.info/keyboard-events-->

<head>

<meta charset="utf-8">

<link rel="stylesheet" href="style.css">

<link href="https://fonts.googleapis.com/css?family=Roboto" rel="stylesheet">

</head>

<body onkeypress="return myKeyPress(event)">

<h1>Näppäimistön demoaminen</h1>

<p>Painelemalla näppäimistön nappuloita (kirjaimet ja numerot) menevät ne jonoon. Saat poistettua merkkejä painamalla backspacea.</p>

<div class="painettu">

<h2>Painettu näppäin:</h2>

<div id="painettu"></div>

</div>

<h2>Jonon tilanne:</h2>

<div class="data" id="tilanne"></div><br>

<h2>Lisätyt merkit:</h2>

<div class="data" id="lisatyt"></div><br>

<h2>Poistetut merkit:</h2>

<div class="data" id="poistetut"></div>

<script src="jquery-3.3.1.min.js"></script>

<script src="script.js"></script>

</body>

</html>

style.css

body {

width: 50%;

margin: 50px auto;

font-family: 'Roboto', sans-serif;

text-align: center;

}

.data {

width: 100%;

min-height: 100px;

border: 1px solid #000;

word-break: break-all;

}

.painettu {

height: 80px;

}

h2 {

margin: 0;

font-size: 20px;

}

#painettu {

font-size: 28px;

font-weight: bold;

margin: 10px;

}

#lisatyt, #poisteut {

text-align: left;

}

script.js

(function () {

function contentLoaded () {

var lisatyt = [];

var poistetut = [];

var tilanne = [];

var kaikki = [];

a = 0;

lisatytdiv = document.getElementById('lisatyt');

poistetutdiv = document.getElementById('poistetut');

tilannediv = document.getElementById('tilanne');

window.addEventListener('keyup', function (e) {

//lisÃ¤tyt taulukko

if (e.keyCode >= 46 && e.keyCode <= 90) {

lisatyt.push(e.key);

lisatytdiv.textContent = lisatyt.join('');

//console.log(chars);

//poistetut

}else if (e.keyCode === 8){

for(i = 0; i < 1 ; i++){

char = lisatyt.join('');

var res = char.charAt(a);

poistetut.push(res);

poistetutdiv.textContent = poistetut.join('');

//console.log(res);

kaikki.shift();

$("tilanne").empty();

tilannediv.textContent = kaikki.join('');

a++;

}

//tilanne

}if(e.keyCode >= 46 && e.keyCode <= 90){

kaikki.push(e.key);

if(e.keyCode === 8) {

}

tilannediv.textContent = kaikki.join('');

}

}, false);

}

window.addEventListener('DOMContentLoaded', contentLoaded, false);

}());

//painettu nÃ¤ppÃ¤in

function myKeyPress(e){

var keynum;

if(window.event) { // IE

keynum = e.keyCode;

} else if(e.which){ // Netscape/Firefox/Opera

keynum = e.which;

}

painettu = document.getElementById('painettu');

painettu.textContent = (String.fromCharCode(keynum));

}