

პროგრამირების აბსტრაქციები

სემინარის ამოცანები #18

1 ცვლადი ზომის მასივი

დაწერეთ **double**-ების შემნახველი მასივის კლასი. მომხმარებელს უნდა შეეძლოს პროგრამის მსვლელობის დროს ამ მასივის ზომის შეცვლა.

header file

```
#ifndef RESIZABLE_ARRAY_H_
#define RESIZABLE_ARRAY_H_

class ResizableArray {
private:
    int currentSize;
    double* data;
public:
    ResizableArray(int size = 10);
    ~ResizableArray();
    void resize(int newSize);
    double getAt(int index);
    void setAt(int index, double value);
    int size();
};

#endif
```

source file

```
#include "error.h"
#include "ResizableArray.h"

ResizableArray::ResizableArray(int size) {
    currentSize = size;
    data = new double[currentSize];
    for (int i = 0; i < currentSize; i++)
        data[i] = 0.0;
}

ResizableArray::~ResizableArray() {
    delete[] data;
}

void ResizableArray::resize(int newSize) {
    double* tmpData = new double[newSize];

    for (int i = 0; i < currentSize && i < newSize; i++)
        tmpData[i] = data[i];
    for (int i = currentSize; i < newSize; i++)
        data[i] = 0.0;

    delete[] data;
    data = tmpData;
}

double ResizableArray::getAt(int index) {
    if (0 <= index && index < currentSize) {
        return data[index]; // return *(data + index);
    } else {
        error("index out of bounds");
    }
}

void ResizableArray::setAt(int index, double value) {
    if (0 <= index && index < currentSize) {
        data[index] = value; // *(data + index) = value;
    } else {
        error("index out of bounds");
    }
}

int ResizableArray::size() {
    return currentSize;
}
```

2 ვექტორის იმპლემენტაცია

დაწერეთ **double** ტიპის ელემენტების შემნახველი ვექტორი კლასი ჩვეულებრივი ოპერაციებით. ვექტორში ელემენტის ჩამოღებას და დამატებას საშალოდ დიდი დრო არ უნდა სჭირდებოდეს.

header file

```
#ifndef MY_VECTOR_H_
#define MY_VECTOR_H_

#include "ResizableArray.h"

class MyVector {
private:
    int currentSize;
    ResizableArray data;
public:
    MyVector(int capacity = 10);
    double getAt(int index);
    void setAt(int index, double value);
    int size();
    void add(double value);
    double popLast();
};

#endif
```

source file

```
#include "MyVector.h"

MyVector::MyVector(int capacity) {
    currentSize = 0;
    data.resize(capacity);
}

double MyVector::getAt(int index) {
    return data.getAt(index);
}

void MyVector::setAt(int index, double value) {
    data.setAt(index, value);
}

int MyVector::size() {
    return currentSize;
}

void MyVector::add(double value) {
    if (currentSize >= data.size())
        data.resize(2*data.size() + 1);

    data.setAt(currentSize, value);
    currentSize++;
}

double MyVector::popLast() {
    if (currentSize == 0)
        error("no more elements");

    double result = data.getAt(currentSize - 1);

    currentSize--;
    if (currentSize*4 < data.size())
        data.resize(data.size()/2);

    return result;
}
```