

# Задачи по Обектно ориентирано програмиране

## Седмица V

### група 4

#### Задача 1.

Да се дефинира клас **IntArrayContainer**, който има следните член-данни:

- *[подходящ тип] array* - реализира същинският контейнер - масив с променлива дължина;
- *int size* - променлива, указваща текущият размер на контейнера;
- *int capacity* - променлива, указваща (конкретния) максимален размер на контейнера - тя трябва да е съобразена с по-горното ограничение за размерността; По подразбиране контейнерът има място за 4 елемента.

Да се реализират следните методи:

- *Канонично представяне, ако е необходимо;*
- *Начин за инициализиране на контейнера с предварително указан капацитет;*
- *Селектори и мутатори, ако е необходимо;*
- *bool empty();*
- *bool full();*
- *void add(аргументи, ако има нужда);*
- *void pop(аргументи, ако има нужда)* - премахва последно добавеният елемент;
- *void remove (аргументи, ако има нужда)* - премахва даден елемент от контейнера. В случай, че той се среща повече от веднъж - премахва се само първото му срещане;
- *bool member(аргументи, ако има нужда);*
- *[подходящ тип] find (bool (\*pred)(аргументи, ако има нужда))* - намира и връща първия срещнат елемент, който отговаря на предиката. В случай че няма такъв - връща свободно създаден елемент от същия тип;
- *[подходящ тип] filter(bool (\*pred)(аргументи, ако има нужда))* - реализира филтър над контейнера и връща нов със филтрираните данни;
- *void map([подходящ тип](\*mod)(аргументи, ако има нужда))* - реализира функция map над контейнера;
- *[подходящ тип] intersection(аргументи, ако има нужда)* - връща нов контейнер, който съдържа сечението на текущия и аргумента;
- *[подходящ тип] complement(аргументи, ако има нужда)* - връща нов контейнер, който съдържа разликата на текущия и аргумента;
- *[подходящ тип] union(аргументи, ако има нужда)* - връща нов контейнер, който съдържа обединението на текущия и аргумента без повторение;
- *void print()* - извежда на екрана всички елементи на масива;

- *int count()* - връща текущия брой на елементите в масива;

Да се реализират следните оператори:

- *[подходящ тип] operator + (аргументи, ако има нужда)* - добавя елемент към контейнера;
- *[подходящ тип] operator += (аргументи, ако има нужда)* - аналог на горния оператор;
- *[подходящ тип] operator + (аргументи, ако има нужда)* - конкатенира елементите на подаден контейнер към тези на текущия; Да се реализира и съответстващ оператор *+=*;
- *[подходящ тип] operator -- (аргументи, ако има нужда)* - премахва елемент последно добавеният елемент от контейнера;
- *\* [подходящ тип] operator bool (аргументи, ако има нужда)* - указва дали текущия контейнер е празен или не;