

# Домашна работа 11

## Държавна администрация v.2

### Въведение

Държавната администрация в кралството на чудовищата остана много доволна от работата на студентите от невидимия университет, както и от софтуера, който те разработили. Вече не било нужно служителите да лекуват мазолите си и да чистят ръцете си изцапани с мастило, в следствие от работата с лист и химикал, а доволно „тропали“ по клавиатурите с пръстчета.

Както всички ние знаем, обаче, всеки софтуерен продукт има нужда от обновяване и поддръжка, затова един слънчев ден, кралят, миризливо и зловонно чудовище, се събудил и решил да отново да потърси студентите от невидимия университет и да ги помоли да разработят втора версия на софтуерния продукт.

***Запратайте ръкави, ще се пише код!***

### Спецификация

Да се разработи конзолно приложение, което трябва да обработва числа, да ги съхранява в подходяща структура (масив) и да извършва операции с тях. При стартиране на приложението трябва да се въведе броят на числата, които ще се въвеждат, а след това и да се въведат тези числа, като те не трябва да са по-малки от нула и по-големи от 100.

#### Примерен вход / изход

Въведете колко на брой числа ще въвеждате.

10

След това трябва да предоставите възможност за въвеждане на числата, като сте свободни да ги прочетете едно по едно от конзолата или всички заедно. След като се въведе броят на числата и самите числа, трябва да се изведе меню с опции, от които държавните служители трябва да могат да избират.

## Меню с опции:

1. Сортиране на въведените числа във възходящ ред
2. Сортиране на въведените числа в низходящ ред
3. Търсене на позиция на конкретно число
4. Разбъркване на числата
5. Изчисляване на сбора на всички числа
6. Намиране на най-голямото число
7. Намиране на най-малкото число
8. Намиране средно-аритметично на числата
9. Проверка за симетричност на масива от числа
10. Обръщане на масива от числа
11. Визуализирай въведените числа
12. Изход

### Функционалност 1. Сортиране на въведени числа във възходящ ред

Въведените от потребителя числа трябва да се сортират във възходящ ред и да се визуализират в конзолата. Например: [1, 2, 3, 4, 5, 6, 7, 8]. След това трябва да се предостави възможност на потребителя да се върне към основното меню или да излезе от програмата. Използвайте Bubble sort, Quick sort или Selection sort. **Да НЕ СЕ използва класът Arrays и принадлежащите му методи.**

### Функционалност 2. Сортиране на въведените числа в низходящ ред

Въведените от потребителя числа трябва да се сортират в низходящ ред и да се визуализират в конзолата. Например: [8, 7, 6, 5, 4, 3, 2, 1]. След това трябва да се предостави възможност на потребителя да се върне към основното меню или да излезе от програмата. Използвайте Bubble sort, Quick sort или Selection sort. **Да НЕ СЕ използва класът Arrays и принадлежащите му методи.**

### Функционалност 3. Търсене на позицията на конкретно число

След като се избере тази опция, трябва да подканите потребителя да въведе числото, чиято позиция иска да намери. Ако числото бъде намерено се визуализира в конзолата самото число и позицията в масива, на която е намерено, а в противен случай се изписва съобщение, че числото не е намерено. След това трябва да се предостави възможност на потребителя да се върне към основното меню или да излезе от програмата. **Да НЕ СЕ използва класът Arrays и принадлежащите му методи.**

### Функционалност 4. Разбъркване на числата

Трябва да се имплементира функционалност, която разбърква (*размества*) числата в масива на произволни позиции и ги визуализира в конзолата. За целта, използвайте класа Random за генериране на случайни числа. След това трябва да се предостави възможност на потребителя да се върне към основното меню или да излезе от програмата. **Да НЕ СЕ използва класът Arrays и принадлежащите му методи.**

## **Функционалност 5. Изчисляване на сбора на всички числа**

Трябва да се имплементира функционалност, която сумира всички въведени от потребителя числа и визуализира сборът им в конзолата. След това трябва да се предостави възможност на потребителя да се върне към основното меню или да излезе от програмата. **Да НЕ СЕ използва класът Arrays и принадлежащите му методи.**

## **Функционалност 6. Намиране на най-голямото число**

Трябва да се имплементира функционалност, която намира най-голямото число в масива и го визуализира в конзолата. След това трябва да се предостави възможност на потребителя да се върне към основното меню или да излезе от програмата. **Да НЕ СЕ използва класът Arrays и принадлежащите му методи.**

## **Функционалност 7. Намиране на най-малкото число**

Трябва да се имплементира функционалност, която намира най-малкото число в масива и го визуализира в конзолата. След това трябва да се предостави възможност на потребителя да се върне към основното меню или да излезе от програмата. **Да НЕ СЕ използва класът Arrays и принадлежащите му методи.**

## **Функционалност 8. Намиране средно-аритметично на числата**

Трябва да се имплементира функционалност, която намира средно-аритметичното на числата от масива и го визуализира в конзолата. След това трябва да се предостави възможност на потребителя да се върне към основното меню или да излезе от програмата. **Да НЕ СЕ използва класът Arrays и принадлежащите му методи.**

## **Функционалност 9. Проверка за симетричност на масива от числа**

Трябва да се имплементира проверка за симетричност на елементите от масива. Какво означава това? Един масив е симетричен, ако първият и последният му елемент са еднакви и същевременно вторият и предпоследният му елемент също са еднакви и т.н.

Ако масивът е симетричен изведете в конзолата съобщението „Масивът е симетричен“ и изведете елементите от масива. Ако масивът не е симетричен изведете в конзолата съобщението „Масивът не е симетричен“ и изведете елементите от масива. **Да НЕ СЕ използва класът Arrays и принадлежащите му методи.** След това трябва да се предостави възможност на потребителя да се върне към основното меню или да излезе от програмата.

## **Функционалност 10. Обръщане на масива от числа**

Трябва да се имплементира функционалност, която обръща числата в масива и ги визуализира в конзолата. Какво означава да се обърне масивът? Например, ако елементите са били в първоначалния си вид [1, 2, 3, 4, 5], то след обръщането масивът да има следния вид [5, 4, 3, 2, 1]. След това трябва да се предостави възможност на

потребителя да се върне към основното меню или да излезе от програмата. **Да НЕ СЕ използва класът *Arrays* и принадлежащите му методи.**

### **Функционалност 11. Визуализирай въведените числа**

Трябва да се имплементира функционалност, която визуализира в конзолата въведените от потребителя числа. След това трябва да се предостави възможност на потребителя да се върне към основното меню или да излезе от програмата. **Да НЕ СЕ използва класът *Arrays* и принадлежащите му методи.**

### **Функционалност 12. Изход от менюто**

Трябва да се имплементира функционалност, която **приключва изпълнението на програмата.**

## Критерии за оценяване

Домашната работа ще се оценява в точкова система. Максималното количество точки, които можете да получите от решението на поставените ви условия е **100 точки**.

<b>Функционалност</b>	<b>Точки</b>
Реализация на меню с опции	5
Реализация на функционалност 1	10
Реализация на функционалност 2	10
Реализация на функционалност 3	10
Реализация на функционалност 4	10
Реализация на функционалност 5	5
Реализация на функционалност 6	5
Реализация на функционалност 7	5
Реализация на функционалност 8	10
Реализация на функционалност 9	10
Реализация на функционалност 10	10
Реализация на функционалност 11	5
Реализация на функционалност 12	5
<b>Общо (точки)</b>	<b>100</b>

Повече информация за предаването на вашите решения можете да намерите в секция **“Начини и срокове за предаване”**.

## ***Начини и срокове за предаване***

Разработката на домашната работа се прикачва към съответното задание като прикачена хипервръзка (*линк*) към GitLab проекта, който сте създали, и след това се предава. За целта, трябва да го предадете като натиснете бутон “**Предай**”.

Предаването на заданието трябва да се случи не по-късно от определения краен срок, упоменат в заданието на класната стая (*Google Classroom*). **Стандартният краен срок за предаване е до вторник 20:30 ч. на следващата седмица.**

Задължително е да си направите нов проект в GitLab със заглавие **pu-fmi-java-intro-w11**

## ***Начини за комуникация с преподавателския колектив***

При въпроси от ваша страна, използвайте частните коментари към заданието в класната стая (*ако въпросите са относно вашата имплементация и проблеми с нея*) и публичните коментари (*ако имате въпроси относно самото задание*).