

самостоятелна работа – седмица 5

Държавна администрация

ВЪВЕДЕНИЕ В УПРАЖНЕНИЕТО

Кралството на чудовищата, както всяко едно друго, има държавна администрация, която обслужва интересите на народа. До сега всички служители, изтормозени от капризите на народа, са работили с лист и химикал.

Един слънчев ден, кралят, миризливо и зловонно чудовище, се събудил и решил да купи компютри за всеки служител в държавната администрация. Понеже бил свидлив, решил да купи компютри втора употреба, които едва успявали да се стартират, вентилаторите им мъркали като 45 годишен трактор, а на всичкото отгоре, нямали графичен потребителски интерфейс и работили в DOS среда.

Компютрите били купени и монтирани по работните бюра на служителите, обаче сега бил нужен софтуер, който да помогне на държавните работници да обработват информацията. Кралят дочул някакъв слух, че студентите от невидимия университет били истински хакери и виртуози в разработката на софтуер, и затова се обърнал към тях за помощ. Запрятайте 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
<p>\$ Въведете колко на брой числа ще въвеждате.</p> <p>\$ 10</p>

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

СОРТИРАНЕ НА ВЪВЕДЕНИТЕ 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.**

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.**

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.

С цел бързодействие, искаме да бъде имплементирано двоично търсене в масив. **Да НЕ СЕ използва класът Arrays и принадлежащите му методи.**

Сценарий 1

Примерен вход / изход
<p>\$ Въведете числото, чиято позиция искате да намерите.</p> <p>\$ 10</p> <p>\$ Числото 10 е намерено на позиция 2.</p>

Сценарий 2

Примерен вход / изход
<p>\$ Въведете числото, чиято позиция искате да намерите.</p> <p>\$ 10</p> <p>\$ Числото 10 не е намерено.</p>

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.**

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.**

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.**

НАМИРАНЕ НА НАЙ-МАЛКОТО ЧИСЛО

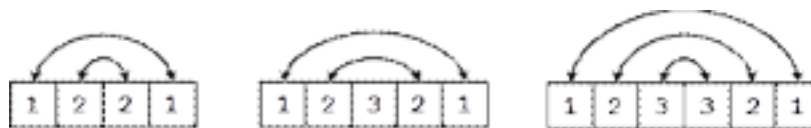
Трябва да се имплементира функционалност, която намира най-малкото число в масива и го визуализира в конзолата. След това трябва да се предостави възможност на потребителя да се върне към основното меню или да излезе от програмата. **Да НЕ СЕ използва класът Arrays и принадлежащите му методи.**

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.**

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. На картинката са дадени няколко примера за симетрични масиви.



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.

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.**

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.**

ИЗХОД ОТ МЕНЬОТО

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

ЗАДАЧИ ЗА ИМПЛЕМЕНТАЦИЯ

Всяка една от разработените задачи ще се оценява в точкова система с арбитрарни тежести. Максималното количество точки, които можете да получите от решението на поставените ви условия е **100 точки**.

В рамките на първото упражнение, ще включим и дистрибутирането на вашето решение като част от крайната оценка. Повече информация за предаването на вашите решения можете да намерите в секция **ПРЕДАВАНЕ НА РЕШЕНИЯТА**.

Имплементация на	Точки
• Визуализация на менюто	15
• Сортиране на въведените числа във възходящ ред	6
• Сортиране на въведените числа в низходящ ред	6
• Търсене на позицията на конкретно число	10
• Разбъркване на числата	7
• Изчисляване на сбора на всички числа	7
• Намиране на най-голямото число	7
• Намиране на най-малкото число	7
• Намиране средно-аритметично на числата	7
• Проверка за симетричност на масива от числа	7
• Обръщане на масива от числа	7
• Визуализирай въведените числа	7
• Изход	7

КРИТЕРИИ ЗА ВАЛИДНА ОЦЕНКА

В тази част са описани критериите, които добавят положителен резултат към крайната оценка. За да бъдат приложени е необходимо да не влязло в сила нито едно от правилата описани в предходната част.

КОГА ДОМАШНОТО СЕ НАКАЗВА С НУЛЕВ РЕЗУЛТАТ?

Една домашна работа няма да бъде проверявана или ще бъде наказана с 0 точки при някое от следните обстоятелства.

Обстоятелство	Наказание
<ul style="list-style-type: none">Програмата не се компилираПрограмата е предадена във вид, който изисква допълнителна намеса от страна на проверяващия, за да се стартира.Изпълними файлове с грешно разширение.	0 точки
<ul style="list-style-type: none">Домашно, предадено извън срока описан в заданието. Без пазарлъци за паднал интернет или гръмнал компютър.	0 точки
<ul style="list-style-type: none">Нагло плагиатство, дословно копиране и погазване на моралните принципи за справедливо предаване на домашна работа	0 точки
<ul style="list-style-type: none">Грубо погазване на описаните правила и тотално неспазване на условията на домашните работи	0 точки

ПРОЦЕНТНО СЪОТНОШЕНИЕ НА КРИТЕРИИТЕ, СПРЯМО ДАДЕНИТЕ ТОЧКИ

Графична имплементация	проценти
<ul style="list-style-type: none">Форматиране на кода	10%
<ul style="list-style-type: none">Структура на кода	20%
<ul style="list-style-type: none">Именуване на променливи и методи	20%
<ul style="list-style-type: none">Коректност на имплементираната функционалност	50%

ФОРМАТИРАНЕ НА КОДА

Когато псувате вашите колеги, за неформатираната гнус, която са ви дали за да правите ревюта, ще се сещате за нас 😊 Напомняме, че повечето среди за разработка (IDE) извършват форматирането автоматично. За това лошо форматиран код вече го приемаме по-скоро за negliжиране, отколкото за затруднение.

СТРУКТУРА НА КОДА

Структурирането на изходния код се извършва чрез правилно разпределение в компонентите, които могат да капсулират функционалност. До момента единствените такива компоненти в Java, които са засегнати на практическите занятия са методите. Затова и изискването, за сега, е функционалността да се разпределя в различни методи, така че всеки метод да съдържа ясна бизнес-логика, която извършва точно определена дейност. Разбира се колегите, които са напред с материала, познават и използват и други компоненти на езика (като класове) могат да не се ограничават само до методите.

ИМЕНУВАНЕ НА ПРОМЕНЛИВИ И МЕТОДИ

Правилното именуване на методите и променливите са ключови за лесното разбиране на кода, което е свързано с предишните два критерия.

КОРЕКТНА ФУНКЦИОНАЛНОСТ

Това е критерият с най-голяма тежест, тъй като целта на всяко задание, в крайна сметка, е коректно работещо приложение.

ПРЕДАВАНЕ НА РЕШЕНИЯТА

ТЕХНИЧЕСКИ ИЗИСКВАНИЯ

Всички участници във входния изпит трябва да предадат задачата си, използвайки съвременна и проверена схема за дистрибуция на софтуерни инструкции.

Задължително е да си направите нов проект в *GitLab-a на курса*, със заглавие [pu-fmi--java-intro-w5](#). Ако не можете да го запомните - копирайте го. **Ако имате нужда от допълнителна помощ, използвайте инструкцията, която е качена в Google Classroom**

Името на хранилището не бива да бъде бъркано, преправено, импровизирано или подложено на творческа интерпретация. Ако това се случи, за ваше съжаление задачата ви няма да бъде приета и проверена, което ще донесе само и единствено тъга в сърцата на всички панди в средната земя.

Линк към хранилището трябва да бъде предоставен в Google Classroom преди крайния срок на заданието. За целта, трябва да го прикачите към заданието и след това да го предадете като натиснете бутон "Предай". Хранилище, което е предоставено като коментар няма да бъде отворено и оценено!

Ако изпитвате затруднение комуникирайте с преподавател за правилния начин, по който да предадете решението. Като е добре това да се случва в срокове различни от последната минута на последния ден.

СРОКОВЕ ЗА ПРЕДАВАНЕ НА ЗАДАНИЕТО

Краен срок за предаване на заданието е **до час 23:59 часа на дата 11/05/2021** - ден **Вторник**.