

2. Семинар: данные и функции

2.1. Инструментарий

- Презентация для преподавателя, ведущего семинар;
- Фон GeekBrains для проведения семинара в Zoom;
- JDK любая 11 версии и выше;
- IntelliJ IDEA Community Edition для практики и примеров используется IDEA.

2.2. Цели семинара

- Закрепить полученные на лекции знания, хранения примитивных и ссылочных типов данных;
- Получить практический навык создания функций по описанию;
- Попрактиковаться в написании простых функций, манипулирующих ссылочными данными.

2.3. План-содержание

Что происходит	Время	Слайды	Описание
Организационный момент	5	1-4	Преподаватель ожидает студентов, поддерживает активность и коммуникацию в чате, озвучивает цели и планы на семинар. Важно упомянуть, что выполнение домашних заданий с лекции является, фактически, подготовкой к семинару
Quiz	5	3-14	Преподаватель задаёт вопросы викторины, через 30 секунд демонстрирует слайд-подсказку и ожидает ответов (4 вопроса, по минуте на ответ)
Рассмотрение ДЗ лекции	10	15-18	Преподаватель демонстрирует свой вариант решения домашнего задания с лекции, возможно, по предварительному опросу, демонстрирует и разбирает вариант решения одного из студентов
Вопросы и ответы	10	19	Преподаватель ожидает вопросов по теме прошедшей лекции, викторины и продемонстрированной работы

Что происходит	Время	Слайды	Описание
Задание 1	10	20-22	Сравнить насколько разные могут быть прочтения одного и того же технического задания - одна функция для двух значений, возврат значений, возврат индекса, объявление исходного массива внутри функции поиска и др);
Задание 2	10	23-25	Корректная манипуляция индексами, как следствие, сокращение числа возможных проходов по массиву и ускорение работы приложения
Перерыв (если нужен)	5	26	Преподаватель предлагает студентам перерыв на 5 минут (студенты голосуют)
Задание 3	20	27-29	Формирование алгоритмического мышления при решении задач с описанием верхнего уровня
Задание 4	15	27-29	Понимание внутренней механики работы фреймворка коллекций, повышение уровня абстракции написанного кода
Задание 5 (необязат)	20	27-29	Описание базовых алгоритмов манипуляции данными с применением вспомогательных массивов
Домашнее задание	5	39	Объясните домашнее задание, подведите итоги урока
Рефлексия	10	40-42	Преподаватель запрашивает обратную связь
Длительность	125		

2.4. Подробности

2.4.1. Организационный момент

- **Цель этапа:** Позитивно начать урок, создать комфортную среду для обучения.
- **Тайминг:** 3-5 минут.
- **Действия преподавателя:**
 - Презентует название курса (первый раз) и семинара;
 - Рассказывает немного о себе;
 - Запрашивает активность от аудитории в чате;
 - Презентует цели курса и семинара;
 - Презентует краткий план семинара и что студент научится делать.

2.4.2. Quiz

- **Цель этапа:** Вовлечение аудитории в обратную связь.
- **Тайминг:** 5-7 минут (4 вопроса, по минуте на ответ).

— **Действия преподавателя:**

- Преподаватель задаёт вопросы викторины, представленные на слайдах презентации;
- через 30 секунд демонстрирует слайд-подсказку и ожидает ответов.

— **Вопросы и ответы:**

1. Магическое число – это: (1)
 - (a) числовая константа без пояснений;
 - (b) число, помогающее в вычислениях;
 - (c) числовая константа, присваиваемая при объявлении переменной.
2. Какое значение будет содержаться в переменной a после выполнения строки `int a = 10.0f/3.0f;` (3)
3. Сколько будет создано одномерных массивов при инициализации массива `3x3x3`? (13)
4. `2 + 2 * 2 == 2 << 2 >> 1`? (false? `6 != 4`)

2.4.3. Рассмотрение ДЗ

- **Цель этапа:** Пояснить неочевидные моменты в формулировке ДЗ с лекции, синхронизировать прочитанный на лекции материал к началу семинара.

- **Тайминг:** 15-20 минут.

— **Действия преподавателя:**

- Преподаватель демонстрирует свой вариант решения домашнего задания из лекции;
- возможно, по предварительному опросу, демонстрирует и разбирает вариант решения одного из студентов.

— **Домашнее задание из лекции:**

- Написать метод «Шифр Цезаря», с булевым параметром зашифрования/расшифрования, и числовым ключом;

Вариант решения

Листинг 1: Шифр Цезаря

```
1 private static String caesar(String in, int key, boolean encrypt) {
2     if (in == null || in.isEmpty())
3         return null;
4
5     final int len = in.length();
6     char[] out = new char[len];
7     for (int i = 0; i < len; ++i) {
8         out[i] = (char) (in.charAt(i) + ((encrypt) ? key : -key));
9     }
10    return new String(out);
11 }
```

- Написать метод, принимающий на вход массив чисел и параметр n. Метод должен осуществить циклический (последний элемент при сдвиге становится первым) сдвиг всех элементов массива на n позиций;

Вариант решения

Листинг 2: Сдвиговый метод

```

1 private static void shifter(int[] a, int n) {
2     n %= a.length;
3     int shift = a.length + n;
4     shift %= a.length;
5
6     for (int i = 0; i < shift; i++) {
7         int temp = a[a.length - 1];
8         System.arraycopy(a, 0, a, 1, a.length - 1);
9         a[0] = temp;
10    }
11 }

```

- Написать метод, которому можно передать в качестве аргумента массив, состоящий строго из единиц и нулей (целые числа типа `int`). Метод должен заменить единицы в массиве на нули, а нули на единицы и не содержать ветвлений. Написать как можно больше вариантов метода

Вариант решения

Листинг 3: Инверсия

```

1 private static void change(int[] a) {
2     for (int i = 0; i < a.length; i++) {
3         a[i] = 1 - a[i];
4         // a[i] = (a[i] - 1) * -1;
5         // a[i] = (a[i] + 1) % 2;
6     }
7 }

```

2.4.4. Вопросы и ответы

- **Ценность этапа** Задать задание для самостоятельного выполнения между занятиями.
- **Тайминг** 5-15 минут
- **Действия преподавателя**
 - Преподаватель ожидает вопросов по теме прошедшей лекции, викторины и продемонстрированной работы;
 - Если преподаватель затрудняется с ответом, необходимо мягко предложить студенту ответить на его вопрос на следующем семинаре (и не забыть найти ответ на вопрос студента!);
 - Предложить и показать пути самостоятельного поиска студентом ответа на заданный вопрос;
 - Посоветовать литературу на тему заданного вопроса;
 - Дополнительно указать на то, что все сведения для выполнения домашнего задания, прохождения викторины и работы на семинаре были рассмотрены в методическом материале к этому или предыдущим урокам.

2.4.5. Задание 1

- **Ценность этапа** Базовая манипуляция данными внутри массива.
- **Тайминг** 10-15 минут.
- **Действия преподавателя**

- Первые пять минут уклоняться от ответов на уточняющие вопросы
- Пояснить студентам ценность опыта (сравнить насколько разные могут быть прочтения одного и того же технического задания - одна функция для двух значений, возврат значений, возврат индекса, объявление исходного массива внутри функции поиска и др);
- Выдать задание группам студентов, показать где именно следует искать терминал ОС;
- Если группа студентов справилась с заданием, а времени осталось более 5 минут, выдавать группе задания «со звёздочкой».

– **Задания:**

- Задать одномерный массив. Написать методы поиска в нём минимального и максимального элемента;

Вариант решения

Листинг 4: Поиск минимального значения

```

1 private static int findMin(int[] a) { // returns the minimum value
2     int min = a[0];
3     for (int i = 1; i < a.length; i++) {
4         if (a[i] < min) {
5             min = a[i];
6         }
7     }
8     return min;
9 }

```

- *1 Привести функции к корректному виду и дополнительно написать ещё две функции так, чтобы получились (четыре) функции поиска минимального и максимального как значения, так и индекса.

Вариант решения

Листинг 5: Поиск индекса максимального значения

```

1 private static int findMax(int[] a) { // returns the maximum index
2     int max = 0;
3     for (int i = 1; i < a.length; i++) {
4         if (a[i] > a[max])
5             max = i;
6     }
7     return max;
8 }

```

2.4.6. Задание 2

- **Ценность этапа** Оптимизация сложности алгоритмов при работе с многомерными массивами.
- **Тайминг** 10-15 минут.
- **Действия преподавателя**
 - Пояснить студентам ценность этого опыта (корректная манипуляция индексами, как следствие, сокращение числа возможных проходов по массиву и ускорение работы приложения);
 - Пояснить студентам в каком виде выполнять и сдавать задания;

- Выдать задание группам студентов, показать где и как скачивать необходимый инструментарий, если он ещё не установлен;
- Если группа студентов справилась с заданием, а времени осталось более 5 минут, выдавать группе задания «со звёздочкой».

– Задания

- Создать квадратный целочисленный массив (количество строк и столбцов одинаковое), заполнить его диагональные элементы единицами, используя цикл(ы)

Вариант решения

Листинг 6: Заполнение диагональных элементов

```

1 private static void fillDiagonal(int[][] a) {
2     for (int i = 0; i < a.length; i++) {
3         a[i][i] = 1;
4         a[i][a.length - 1 - i] = 1;
5     }
6 }

```

- *1 дописать функцию вывода двумерного массива в консоль

Вариант решения

Листинг 7: Вывод массива в терминал

```

1 private static void printTwoDimArray(int[][] a) {
2     for (int i = 0; i < a.length; i++) {
3         System.out.println(Arrays.toString(a[i]));
4     }
5 }

```

2.4.7. Задание 3

- **Ценность этапа** Формирование алгоритмического мышления при решении задач с описанием верхнего уровня.
- **Тайминг** 15-20 минут
- **Действия преподавателя**
 - Пояснить студентам ценность этого опыта (ТЗ довольно редко бывают чёткими и никогда не говорят программисту, что именно нужно написать);
 - Выдать задание группам студентов;
 - Если группа студентов справилась с заданием, а времени осталось более 5 минут, выдать группе задание «со звёздочкой».

– Задания

- Написать метод, в который передается не пустой одномерный целочисленный массив, метод должен вернуть true если в массиве есть место, в котором сумма левой и правой части массива равны. Примеры:

checkBalance([1, 1, 1, || 2, 1]) → true,

checkBalance([2, 1, 1, 2, 1]) → false,

checkBalance([10, || 1, 2, 3, 4]) → true.

Абстрактная граница показана символами ||, эти символы в массив не входят.

Вариант решения

Листинг 8: Вариант со сложностью $O(n^2)$

```

1 private static boolean checkBalance(int[] a) {
2     int left = 0;
3     for (int i = 0; i < a.length - 1; i++) {
4         left += a[i];
5         int right = 0;
6         for (int j = i + 1; j < a.length; j++) {
7             right += a[j];
8         }
9         if (left == right) return true;
10    }
11    return false;
12 }

```

Листинг 9: Вариант со сложностью $O(2n)$

```

1 private static boolean checkBalance2(int[] a) {
2     int sum = 0;
3     for (int i = 0; i < a.length; i++) {
4         sum += a[i];
5     }
6     if (sum % 2 != 0) return false;
7     int left = 0;
8     for (int i = 0; i < a.length; i++) {
9         left += a[i];
10        sum -= a[i];
11        if (left == sum) return true;
12    }
13    return false;
14 }

```

*₁ написать этот же метод таким образом, чтобы в нём использовался только один цикл.

Вариант решения

Листинг 10: * Вариант со сложностью $O(n)$

```

1 private static boolean checkBalance3(int[] a) {
2     int lbound = 0;
3     int rbound = a.length - 1;
4     int left = 0;
5     int right = 0;
6     while (lbound <= rbound) {
7         if (left > right)
8             right += a[rbound--];
9         else
10            left += a[lbound++];
11    }
12    return left == right;
13 }

```

2.4.8. Задание 4

- **Ценность этапа** Понимание внутренней механики работы фреймворка коллекций, повышение уровня абстракции написанного кода.
- **Тайминг** 15-20 минут
- **Действия преподавателя**
 - Пояснить студентам ценность этого опыта (написание собственных функций, реализующих алгоритмы часто помогает в ситуациях, когда задача не решается

стандартными средствами)

- Выдать задание группам студентов
- Если группа студентов справилась с заданием, а времени осталось более 5 минут, выдать группе задание «со звёздочкой».
- Если нужно, через 7 минут после старта, дать подсказку для первой части задания (сигнатура функции должна содержать не только передаваемый массив, но и его текущее заполнение, которое нужно отслеживать самостоятельно)

– Задания

- Написать функцию добавления элемента в конец массива таким образом, чтобы она расширяла массив при необходимости.

Здесь нет смысла показывать не лучшее, но самое популярное решение, поэтому можно продемонстрировать сразу вариант решения «со звёздочкой».

- *₁ Функция должна возвращать ссылку на вновь созданный внутри себя массив, а не использовать глобальный

Вариант решения

Листинг 11: * Вариант без глобального массива

```
1 int[] add(int[] arr, int current, int value) {
2     if (current == arr.length) {
3         int[] temp = new int[arr.length * 2];
4         System.arraycopy(arr, 0, temp, 0, arr.length);
5         arr = temp;
6     }
7     arr[current++] = value;
8     return arr;
9 }
10
11 // main
12 int[] array = {1,2};
13 int size = 2;
14 System.out.println(size + " = " + Arrays.toString(array));
15 array = add(array, size++, 6);
16 System.out.println(size + " = " + Arrays.toString(array));
17 array = add(array, size++, 6);
```

2.4.9. Задание 5 (необязательное)

- **Ценность этапа** Описание базовых алгоритмов манипуляции данными с применением вспомогательных массивов.

- **Тайминг** 15-20 минут

– Действия преподавателя

- Объяснить студентам, в чём заключается алгоритм сортировки подсчётом. Для сортировки подсчётом алгоритм совершает проход по исходному массиву, подсчитывая количество повторений каждого значения, и записывая эту информацию в промежуточный (частотный) массив. Вторым шагом алгоритма совершается обход вспомогательного массива и запись нужного количества значений по возрастанию в исходный массив. Сложность сортировки $O(2n)$. Например:

$$x[2, 1, 0, 4, 3, 0, 0, 1, 2] \rightarrow t[3(x_0), 2(x_1), 2(x_2), 1(x_3), 1(x_4)] \rightarrow x[0, 0, 0, 1, 1, 2, 2, 3, 4]$$

- Выдать задание группам студентов
- Если группа студентов справилась с заданием, а времени осталось более 5 минут, выдать группе задание «со звёздочкой».

– **Задания**

- Написать метод, осуществляющий сортировку одномерного массива подсчётом. Важное ограничение состоит в том, что для этой сортировки диапазон значений исходного массива должен находиться в разумных пределах, например, не более 1000.

Вариант решения

Листинг 12: Pigeonhole sort

```

1 void pigeon(int[] arr) {
2     final int min = getMin(arr);
3     final int max = getMax(arr);
4     int[] freq = new int[max - min + 1];
5     for (int i = 0; i < size; i++)
6         freq[arr[i] - min]++;
7
8     int arrIndex = 0;
9     for (int i = 0; i < freq.length; i++)
10        for (int elems = freq[i]; elems > 0; elems--)
11            arr[arrIndex++] = i + min;
12 }

```

2.4.10. Домашнее задание

- **Ценность этапа** Задать задание для самостоятельного выполнения между занятиями.
- **Тайминг** 5-10 минут.
- **Действия преподавателя**
 - Пояснить студентам в каком виде выполнять и сдавать задания
 - Уточнить кто будет проверять работы (преподаватель или ревьюер)
 - Объяснить к кому обращаться за помощью и где искать подсказки
 - Объяснить где взять проект заготовки для дз

– **Задания**

5-25 мин Решить все задания (в том числе «со звёздочкой»), если они не были решены на семинаре, без ограничений по времени;

Все варианты решения приведены в тексте семинара выше

5-10 мин Написать метод, возвращающий количество чётных элементов массива.

countEvens([2, 1, 2, 3, 4]) → 3

countEvens([2, 2, 0]) → 3

countEvens([1, 3, 5]) → 0

Листинг 13: CountEvens.java

```

1 int countEvens(int[] arr) {
2     int counter = 0;
3     for (int i = 0; i < arr.length; ++i) {
4         if (arr[i] % 2 == 0) {
5             counter++;
6         }
7     }
8 }

```

```
7     }
8     return counter;
9 }
```

10 мин Написать функцию, возвращающую разницу между самым большим и самым маленьким элементами переданного не пустого массива.

Листинг 14: Spread.java

```
1 int spread(int[] arr) {
2     int min = arr[0];
3     int max = arr[0];
4     for (int i = 1; i < arr.length; ++i) {
5         if (arr[i] < min) min = arr[i];
6         if (arr[i] > max) max = arr[i];
7     }
8     return max - min;
9 }
```

10 мин Написать функцию, возвращающую истину, если в переданном массиве есть два соседних элемента, с нулевым значением.

Листинг 15: Zero2.java

```
1 boolean zero2(int[] arr) {
2     for (int i = 0; i < arr.length - 1; ++i) {
3         if (arr[i] == 0 && arr[i + 1] == 0)
4             return true;
5     }
6     return false;
7 }
```

2.4.11. Рефлексия и завершение семинара

- **Цель этапа:** Привести урок к логическому завершению, посмотреть что студентам удалось, что было сложно и над чем нужно еще поработать
- **Тайминг:** 5-10 минут
- **Действия преподавателя:**
 - Запросить обратную связь от студентов.
 - Подчеркните то, чему студенты научились на занятии.
 - Дайте рекомендации по решению заданий, если в этом есть необходимость
 - Дайте краткую обратную связь студентам.
 - Поделитесь ощущением от семинара.
 - Поблагодарите за проделанную работу.