План занятия

1. Массивы
2. Строки: String, StringBuilder, StringBuffer

Теория

Структура данных – способ расположения данных в памяти. Структура данных подразумевает особый алгоритм работы с собой. Для каждой структуры эти алгоритмы отличаются

Массив – это структура данных, которая располагает элемент последовательно друг за другом. Для этого в массиве необходимо при его создании сразу указать его длину, т.е. количество элементов, которое там будет храниться

Доступ к элементу массива можно получить по его индексу.

У каждой структуры данных есть ключ – это то, что дает возможность получить доступ к самим данным(элементу структуры)

В массиве ключом является индекс элемента.

При обращении к массиву мы обращаемся к началу выделенной области памяти, затем размер элемента умножается на его индекс и таким образом мы получаем адрес ячейки, в которой находится элемент

Например:

У нас есть массив типа int. Нам нужен его 5 индекс.

int [] arr = new int[10];

int[5] = 15; Под капотом произойдет умножение размера int, т.е. 4 байт на 5

* 4\*5 = 20 байт. Сдвиг относитель начала массива будет на 20 байт

Индексация(счет) элементов в массиве идет с 0

Строки – это массивы символов

Класс String – обертка над массивом символов. Он решает основные задачи, которые нужны программисту. Для него в Java есть некоторые исключения.

1. Объект класс String можно создать без new -> String str = “Привет” ->

Создание объекта str

1. String позволяет складывать строки: str + “ Мир” – > будет создан новый объект с названием str, который будет содержать фразу “Привет Мир”

Класс String относится к immutable классам – неизменяемым. Объекты таких классов нельзя менять. Если мы меняем значение объекта типа String под капотом создается новый объект, который записывается в эту переменную. Все переменные в классе string имеют модификатор final – константы, т.е. их нельзя менять. Так сделано по определенным причинам, которые станут ясны позже.

Если часто изменять String, то часто будут создавать новые объкты, а старые будут уничтожаться сборщиком мусора(Garbage collector). Такие запуски сборщика мусора будут замедлять систему.

Для борьбы с частыми запусками сборщика мусора в Java существуют два класса для работы со строками: StringBuilder и StringBuffer. Эти классы имеют методы с одинаковыми названиями и эти методы выполняют одну и ту же задачу. Разница между ними в том, что один потокобезопасный, а другой нет

Логика выбора класса для работы со строками:

1. Должен ли объект класса часто меняться?

Нет–> String(значения объектов этого класса не могут измениться)

Да-> переходим ко второму пункту

1. Работа в однопоточной среде?

Да -> StringBuilder

Нет -> StringBuffer

Для класса String есть еще одно исключение. Все значения этого типа в куча создают специальный пул и если строка уже есть в этом пуле, то все переменные с таким значением получают ссылку на эту строку

Пул строк относится только к классу String. Все его значения записываются в этот пул и если появляется еще одна переменная с таким же значением, то ссылка в этой переменной запишется не на новую позицию в пуле строк, а на уже сущестующее значение:

String str1 = “Привет”;

String str2 = “Привет”;

str1 добавляет в пул строк “Привет”, а str2 просто получит ссылку на “Привет”. Т.е. они оба ссылаются на одну запись. Алгоритм добавление строки в пул выглядит так:

1. Проверить есть ли такое, значение в пуле,

Да -> Записываем в переменную ссылку на это значение

Нет-> Добавляем в пул значение и записываем в переменную ссылку на него

Если в нашем примере мы решим изменить значение str1, то может возникнуть ложное впечатление, что изменится и значение str2, но это не так, давайте вспомним алгоритм выбора класса для строк, который указан выше. Значения типа String не могут меняться, поэтому при изменении значения для str1 на “Пока" в пул строк запишется новое слово “Пока” и в str1 будет записана ссылка на “Пока”

str1 = “Пока”

Рассмотреть примыры для мутабельных и имутабельных в проекте Summary\_18.08.23

Больгинство классов являются мутабельными и для низ справдлива та же логика что и для StringBuilder. Покажем это в проекте на примере массива

Заметки

В Java объекты создаются операторм new – он выделяет для них память

Разница между примитивными типами и ссылочными: примитивная переменна содержитв себе свое значение, а ссылочная содержит ссылку на область кучи, в которой хранится значение и по ссылке получает к нему доступ