**Lesson 28. Arrays**

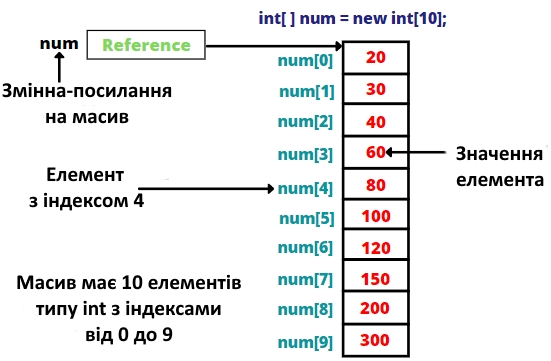
Дане заняття присвячене ще одній дуже важливі темі у Java і не тільки – масиви. Масиви дозволяють зробити код більш ефективним, коли маємо зберігати велику кількість значень. Крім того, певні частини функціоналу телеграм-ботів потребують від розробників знайомства із масивами, саме тому ми і маємо зануритись у них.

Отож, гарного плавання!

**Arrays**

Якщо дуже коротко, то **масив** у Java – це група однотипних змінних, що об’єднані під однією назвою. на які посилається загальна назва масиву.

Змінні в масиві впорядковані, і кожна має індекс (індекси змінних у масиві, так само як і символів у рядках, починаються з 0).

Оскільки масиви є об’єктами в Java, ми можемо отримувати їх довжину (або ж розмір), тобто кількість елементів, за допомогою властивості об’єкта ***length***, де вона і зберігається.

*Розмір масиву не можна змінити* (після ініціалізації). Однак змінну-посилання на масив можна змінити так, щоб вона вказувала на інший масив.

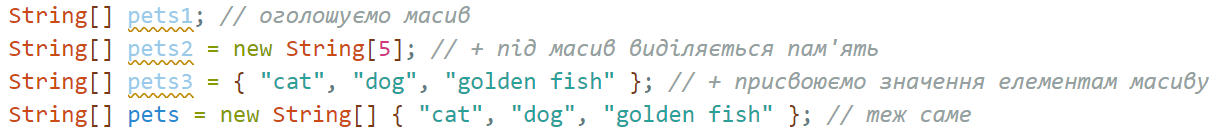
Типи змінних масиву можуть бути як примітивними (*int*, *char* тощо) так об’єктними (референсними), наприклад *String*.

Масиви корисні коли необхідно зберігати ряд значень, що мають одне й теж призначення і не потребують використання різних типів, та пов’язані між собою якимось принципом.

Як працювати з масивами, та що взагалі з ними можна робити з’ясуємо у прикладах з наступного розділу.

**Приклад 1**. Для створення масиву підійде один і наступних варіантів:

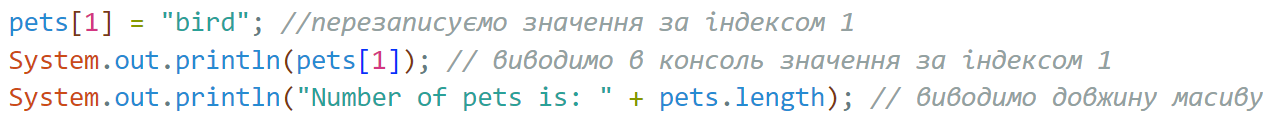
**Examples**



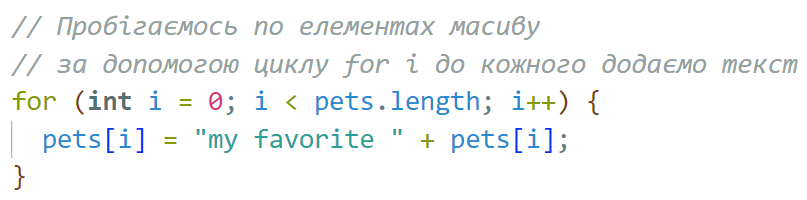
**Приклад 2**. Для того, щоб познайомитись із типовими операціями із масивами, створимо той, у якому зберігаються домашні улюбленці:



Для того, щоб звернутись (**записати або прочитати**) значення масиву використовуються квадратні дужки та індекс. Для отримання довжини масиву використовується властивість *length*:



**Приклад 3**. Для того, щоб **пробігтись по елементах** масиву (тобто виконати якусь дію для кожного із них) можемо скористатись циклом **for**. Змінна лічильник в даному випадку виконує роль ключа, за яким ми отримуємо доступ до поточного елементу в ході циклу:



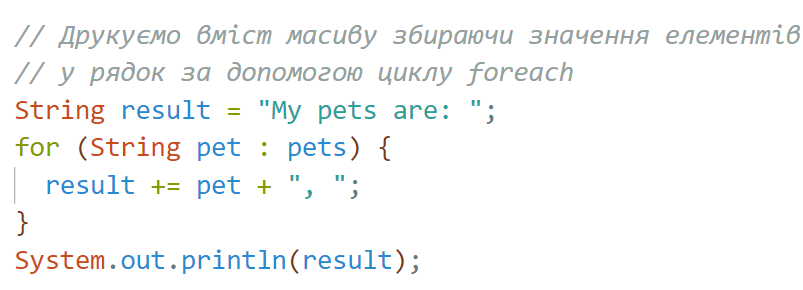
**Приклад 4**. **Роздрукувати масив** можна також за допомогою циклів, однак маємо також коротший/простіший варіант за допомогою методу **toString()** із класу **Arrays**:



**Приклад 5**. Ще один варіант **пробіжки по масиву** – цикл **foreach**. Він також починається словом for, однак працює трохи інакше.

Замість того, щоб ініціалізувати змінну-лічильник циклу, оголошується змінна того самого типу, що й масив, далі двокрапка, за якою ім’я масиву.

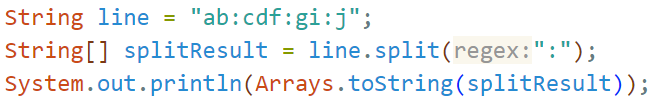
У тілі циклу можна використовувати створену змінну циклу, а не елемент індексованого масиву.



Якщо спробувати перекласти роботу циклу на скріншоті із мови Java на українську, то це звучатиме приблизно так: «Для кожного улюбленця поміж ряду улюбленців виконати … (далі що саме)».

**New Methods**

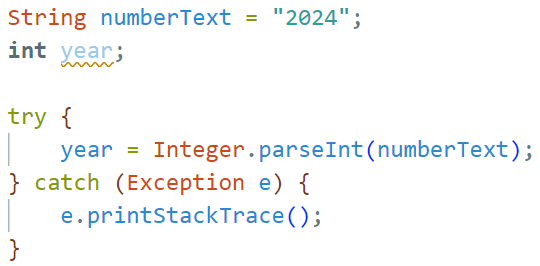
1. Метод **split**() класу String розділяє текстовий рядок за розділювачем, який вказується у параметрі. Метод повертає масив рядків, на які було розділено вхідний рядок. Наведемо приклад:



У результаті виконання даного фрагменту коду в консоль буде виведено:

*[asb, cf, ghi, j]*

1. Метод **parseInt**() класу Integer дозволяє перетворити рядок у числове значення типу int. Якщо отриманий у параметрі рядок не містить валідного значення цілого числа, то метод викидає помилку. Помилку можна і варто обробляти за допомогою конструкції try..catch.



У прикладі вище, помилка не виникне, оскільки в рядку міститься коректне значення цілого числа. Однак, якщо додати в текст яку-небудь літеру, то помилка буде оброблена блоком catch.

**Tasks**

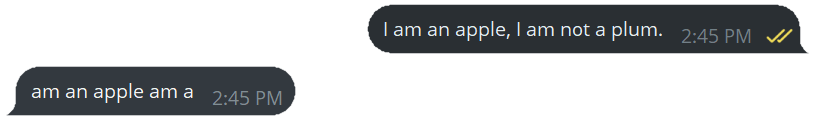
**Завдання 1**. Пройдіться по прикладах із попереднього розділу і виконайте створення масиву, виведення значення одного із елементів масиву, виведення значень усіх елементів трьома способами у консоль.

**Завдання 2**. Cтворіть Telegram-бота, що буде виконувати розрахунок суми чисел отриманих у повідомленні і повертати у відповідь результат. Числа у вхідному повідомленні повинні бути записані кожне в окремому рядку. Якщо в рядку записано значення, яке не є цілим числом, то бот повинен надіслати інформацію про помилку формату числа.

****Для спрощення виконання працювати будемо тільки із цілими числами. Приклад роботи бота наведено на рисунку нижче:

Для виконання розбиття тексту на слова можемо скористатись методом split() класу String. Для перетворення рядків у числа – методом parseInt() класу Integer.

**Завдання 2**. Cтворіть Telegram-бота, що буде вибирати із отриманого повідомлення слова, що починаються на літеру “a”, і повертатиме одним повідомленням.



**Homework**

**Завдання 1.** Створіть три масива типу intдовжиною 6 елементів. Масиви **arr1** та **arr2** заповніть значеннями при створенні. Для **arr3** тільки визначте довжину.

Створіть цикл, який дозволить у третій масив записати значення, які будуть являтись сумою відповідних значень двох перших масивів. Після циклу виведіть усі три масиви у консоль:

Наприклад:

*arr1 = [17, 20, 11, 20, 21, 22]*

*arr2 = [-7, 2, 121, 0, 2, -44]*

*arr3 = [10, 22, 132, 20, 23, -22]*