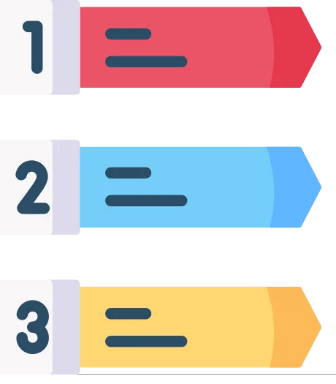
**Lesson 29. ArrayList**

Минулого разу ми познайомились із масивами, а цього разу познайомимось з їх трошки більш прогресивною версією. Клас ArrayList за великим рахунком теж можна назвати масивом, однак він має свої відмінності від звичайних масивів (часом переваги, часом недоліки). Тим не менш ArrayList часто використовують у Java (для створення телеграм-ботів у тому числі). Отож, до роботи!

**ArrayList**

**ArrayList** дуже схожий на звичайний масив, адже в ньому так само міститься сукупність однотипних змінних, які мають індекси, та розташовані по порядку.



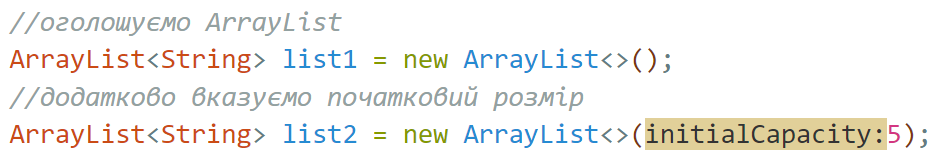
Однак ArrayList має **динамічні властивості**. Вони полягають у тому, що використовуючи ArrayList є можливість легко та зручно **змінювати його довжину** (додавати та видаляти елементи). Такі операції виконувати зі звичайними масивами виходить достатньо не зручно, оскільки довжина масиву не може бути змінена. Натомість доводиться копіювати елементи старого масиву у новий, з потрібною довжиною. ArrayList тим часом вже має усі необхідні алгоритми готові до використання.

Оскільки ArrayList є класом, то для доступу до елементів ArrayList використовуються методи get() та set(), замість квадратних дужок. Крім того ArrayList має ще ряд корисних методів, що дозволяють легко працювати із даними, що збережені в ньому.

Детальніше розглянемо на прикладах.

**Examples**

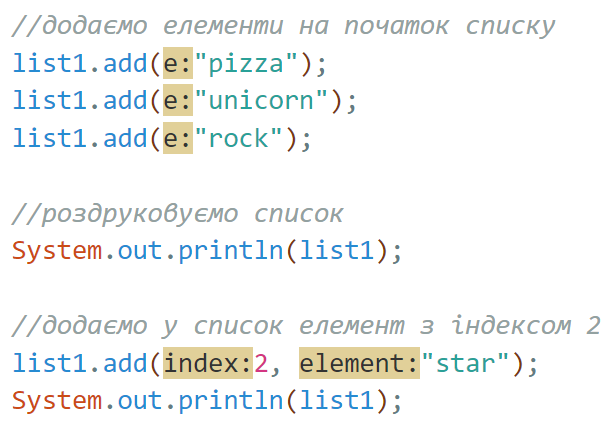
1. **Створення.** Наступним чином можна створити об’єкт ArrayList вказуючи або не вказуючи його початкову довжину. У гострих дужках (<>) записується тип елементів, що будуть зберігатись у ньому:



У другому рядку також показано, що додатково можна вказати початкову довжину списку. У ході роботи із об’єктом, як ми вже зазначали довжина може бути змінена.

1. **Додавання елементів.** Для додавання елементів можна використати метод add(). Кожного разу коли викликається add() у списку створюється ще один елемент.

За замовчуванням елементи додаються на початку списку, тобто новий елемент отримує індекс 0, а всі інші елементи зміщуються на 1. Опційно першим аргументом можна вказати, в яку саме позицію необхідно додати новий елемент.



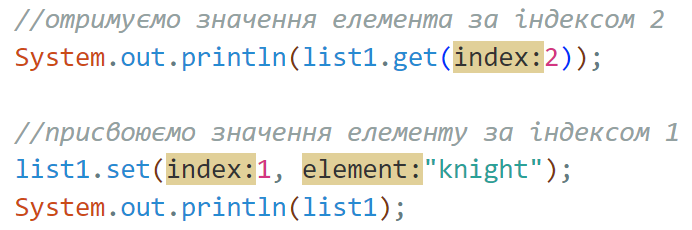
Тоді зміщуються всі елементи, які знаходяться після вказаного.

Достатньо легко роздрукувати ArrayList (просто передати його аргументом у *println*). Результат виведення буде наступний:

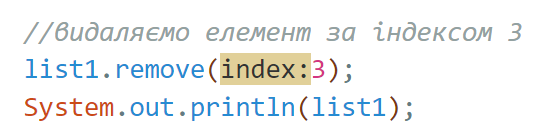
*[pizza, unicorn, rock]*

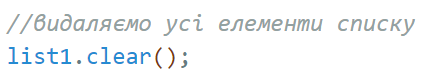
*[pizza, unicorn, star, rock]*

1. **Отримання та присвоєння значення.** У метод get() передаємо один аргумент – індекс елемента, а для set() також вказуємо значення, яке хочемо записати у вказаний елемент.

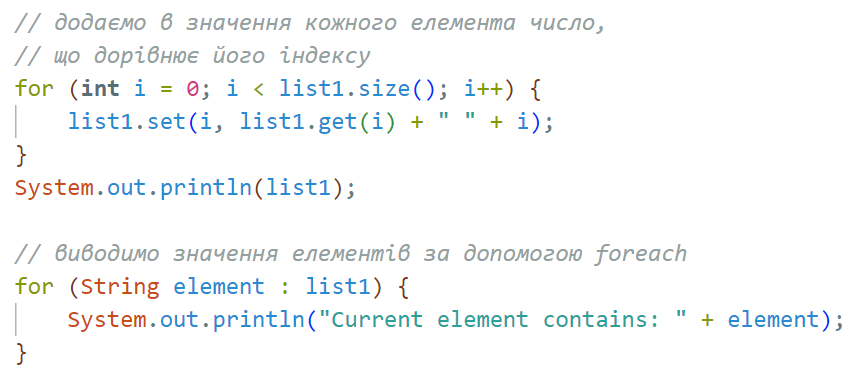


1. **Видалення елементів.** Для того, аби видалити елемент можемо використовувати remove(). У якості аргумента передаємо також індекс елемента.



Для того, аби видалити всі елементи списку одразу можемо використати метод clear().

1. **«Пробіжка» по всіх елементах.** Для того, аби пробігтись по елементах ArrayList можемо використовувати як цикл *for* так і *foreach*. Для визначення кількості елементів у списку маємо метод size().



**Practice**

**Завдання 1**. Пройдіться по прикладах із попереднього розділу і виконайте створення списку ArrayList, виведення значення одного із елементів, виведення значень усіх елементів трьома способами у консоль, а також видалення одного чи всіх елементів у списку.

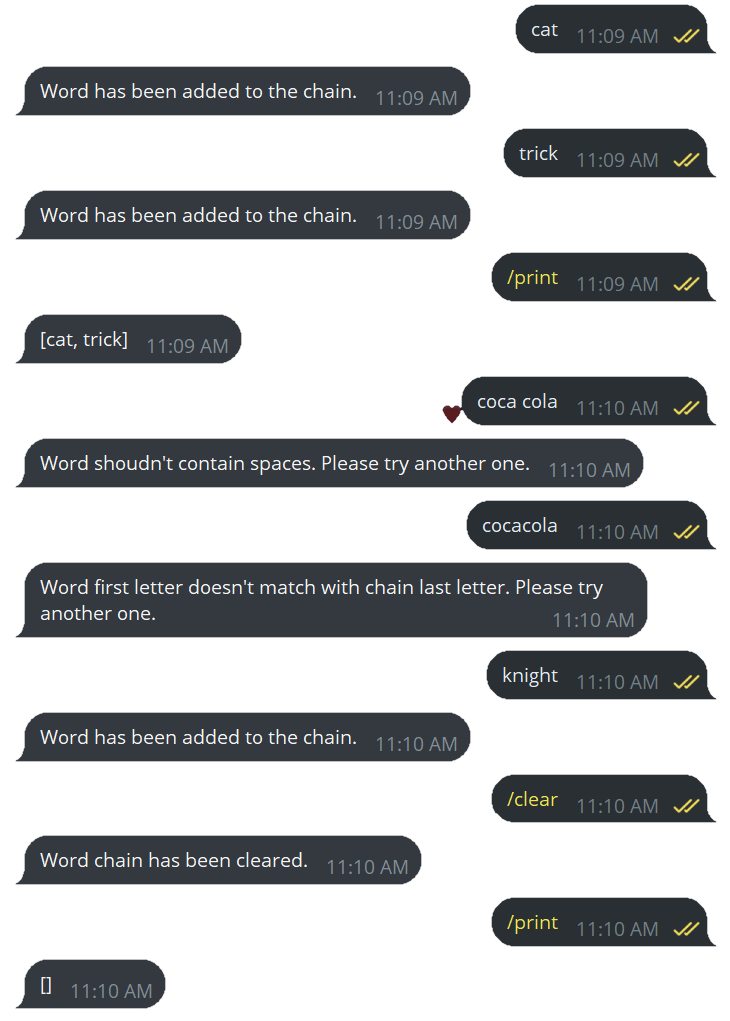
**Завдання 2**. Створіть бота, який буде імітувати гру в слова, де їх потрібно називати одне за одним, так щоб кожне наступне слово починалось на останню літеру попереднього.

Отже бот повинен приймати повідомлення, де міститься тільки одне слово і якщо виконуються умови гри, додавати його у список. За командою **/print** бот надсилає у чат всі слова, що додані у список. По команді **/clear** бот повинен очистити список (гра починається спочатку).

Пропонуємо наступні кроки у створенні даного бота:

1. Зареєструвати команди через функціонал BotFather;
2. Запрограмувати бота на отримання будь-яких текстових повідомлень і додавання їх у список.
3. Додати у алгоритм бота виконання команди **/print**.
4. Додати у алгоритм бота виконання команди **/clear**.
5. Змінити алгоритм так, щоб бот додавав у список повідомлення тільки із одним словом (повідомлення, що не містять пробілів).
6. Додати перевірку слова на головну умову гри (слово починається на останню літеру попереднього).
7. Протестувати роботу бота.

Приклад роботи бота:



**Homework**

**Завдання.** Створіть телеграм-бота, який буде зберігати у список надіслані йому повідомлення. Коли у списку опиниться 5 повідомлень, бот виводить їх у чат у наступному вигляді:

Після виведення список очищується.