# Пример 1: Программа "Alphabet"

## Цель примера

Продемонстрировать процесс создания простейшей программы с использованием списка.

## Введение

В рамках курса была рассмотрена структура данных – массив, который позволяет хранить ограниченное количество элементов одного типа под одной переменной. Несмотря на высокую скорость работы, массив зачастую неудобен в использовании: его размер фиксирован, сложно изменять содержимое, если нужно выполнять операции в середине, например вставить или удалить элемент.

Java предлагает набор интерфейсов – коллекций, которые представляют собой различные структруы данных разной сложности и эффективности. Один из таких интерфейсов – это `List`, являющийся абстракцией над списком данных и, по сути, более удобной в использовании формой массива.

У интерфейса `List` есть несколько конкретных реализаций и одна из самых популярных и универсальных – это `ArrayList`. Несмотря на альтернативные реализации, например `LinkedList`, в рамках данного примера мы будем работать с `ArrayList`, поскольку использование методов будет исходить от интерфейса `List`.

## Практическое руководство

Рассмотрим программу, которая демонстрирует базовые методы интерфейса `List`.

---

### Шаг 1.

Создадим класс `Main`, где создадим список `alphabet`, выведем его содержимое до и после добавления элементов. Добавлять элементы в список можно используя метод `add()`:

```
public class Main {

    public static void main(String[] args) {

        List<String> alphabet = new ArrayList<>();

        System.out.println("alphabet = " + alphabet);

        alphabet.add("A");
        alphabet.add("B");
        alphabet.add("C");
        alphabet.add("D");

        System.out.println("alphabet = " + alphabet);
```

```
    }

}
```

Результат работы программы:

```
alphabet = [] alphabet = [A, B, C, D]
```

ℹ️ **Информация**: Обратите внимание, что в отличие от синтаксиса создания массива, тип данных, который нужно хранить в списке указывается в `<>` (т.н. `Diamon Operator` или `Generic`), например `List<String>`, `List<Dog>` или `List<Integer>`.

‼️ **Примечание**: В `<>` можно указывать только ссылочные типы данных. Примитивные типы нужно заменять их ссылочной версией, например `List<Boolean>` или `List<Character>`.

---

## Шаг 2.

Добавим еще несколько элементов в список и выведем размер списка в консоль используя метод `size()`:

```
public class Main {

    public static void main(String[] args) {

        List<String> alphabet = new ArrayList<>();

        System.out.println("alphabet = " + alphabet);

        alphabet.add("A");
        alphabet.add("B");
        alphabet.add("C");
        alphabet.add("D");

        System.out.println("alphabet = " + alphabet);

        alphabet.add("F");
        alphabet.add("G");

        System.out.println("alphabet = " + alphabet);
        System.out.println("alphabet.size() = " + alphabet.size());

        System.out.println("alphabet.get(2) = " + alphabet.get(2));

    }

}
```

Результат работы программы:

```
alphabet = [] alphabet = [A, B, C, D] alphabet = [A, B, C, D, F, G] alphabet.size() = 6 alphabet.get(2) = C
```

---

## Шаг 3.

Добавим элемент в список между `D` и `F`. Для этого нужно указать индекс, по которому мы хотим поместить элемент:

```
public class Main {

    public static void main(String[] args) {

        List<String> alphabet = new ArrayList<>();

        System.out.println("alphabet = " + alphabet);

        alphabet.add("A");
        alphabet.add("B");
        alphabet.add("C");
        alphabet.add("D");

        System.out.println("alphabet = " + alphabet);

        alphabet.add("F");
        alphabet.add("G");

        System.out.println("alphabet = " + alphabet);
        System.out.println("alphabet.size() = " + alphabet.size());

        System.out.println("alphabet.get(2) = " + alphabet.get(2));

        alphabet.add(4, "E");

        System.out.println("alphabet = " + alphabet);

    }

}
```

Результат работы программы:

```
alphabet = [] alphabet = [A, B, C, D] alphabet = [A, B, C, D, F, G] alphabet.size() = 6 alphabet.get(2) = C alphabet = [A, B, C, D, E, F, G]
```

---

## Шаг 4.

Проверим, существуют ли в списке объекты `"A"`, `"B"` и `"X"`, а затем удалим их двумя способами:

```
public class Main {

    public static void main(String[] args) {

        List<String> alphabet = new ArrayList<>();

        System.out.println("alphabet = " + alphabet);

        alphabet.add("A");
        alphabet.add("B");
        alphabet.add("C");
        alphabet.add("D");

        System.out.println("alphabet = " + alphabet);

        alphabet.add("F");
        alphabet.add("G");
```

```
        System.out.println("alphabet = " + alphabet);
        System.out.println("alphabet.size() = " + alphabet.size());

        System.out.println("alphabet.get(2) = " + alphabet.get(2));

        alphabet.add(4, "E");

        System.out.println("alphabet = " + alphabet);

        System.out.println();
        System.out.println("-- Contains & Remove --");

        System.out.println("alphabet.contains(\"A\") = " + alphabet.contains("A"));
        System.out.println("alphabet.contains(\"B\") = " + alphabet.contains("B"));
        System.out.println("alphabet.contains(\"X\") = " + alphabet.contains("X"));

        alphabet.remove(0);
        alphabet.remove("B");

        System.out.println("alphabet.contains(\"A\") = " + alphabet.contains("A"));
        System.out.println("alphabet.contains(\"B\") = " + alphabet.contains("B"));

        System.out.println("alphabet = " + alphabet);
        System.out.println("alphabet.size() = " + alphabet.size());

    }

}
```

Результат работы программы:

> alphabet = [] alphabet = [A, B, C, D] alphabet = [A, B, C, D, F, G] alphabet.size() = 6 alphabet.get(2) = C alphabet = [A, B, C, D, E, F, G]
>
> -- Contains & Remove -- alphabet.contains("A") = true alphabet.contains("B") = true alphabet.contains("X") = false alphabet.contains("A") = false alphabet.contains("B") = false alphabet = [C, D, E, F, G] alphabet.size() = 5

Обратие внимание, как изменяется размер списка при добавлении или удалении элементов из списка.

---

## Шаг 5.

Весь список можно очистить при помощи метода `clear()`:

```
public class Main {

    public static void main(String[] args) {

        List<String> alphabet = new ArrayList<>();

        System.out.println("alphabet = " + alphabet);

        alphabet.add("A");
        alphabet.add("B");
        alphabet.add("C");
        alphabet.add("D");

        System.out.println("alphabet = " + alphabet);
```

```
        alphabet.add("F");
        alphabet.add("G");

        System.out.println("alphabet = " + alphabet);
        System.out.println("alphabet.size() = " + alphabet.size());

        System.out.println("alphabet.get(2) = " + alphabet.get(2));

        alphabet.add(4, "E");

        System.out.println("alphabet = " + alphabet);

        System.out.println();
        System.out.println("-- Contains & Remove --");

        System.out.println("alphabet.contains(\"A\") = " + alphabet.contains("A"));
        System.out.println("alphabet.contains(\"B\") = " + alphabet.contains("B"));
        System.out.println("alphabet.contains(\"X\") = " + alphabet.contains("X"));

        alphabet.remove(0);
        alphabet.remove("B");

        System.out.println("alphabet.contains(\"A\") = " + alphabet.contains("A"));
        System.out.println("alphabet.contains(\"B\") = " + alphabet.contains("B"));

        System.out.println("alphabet = " + alphabet);
        System.out.println("alphabet.size() = " + alphabet.size());

        System.out.println();
        System.out.println("-- Clear --");

        alphabet.clear();

        System.out.println("alphabet = " + alphabet);
        System.out.println("alphabet.size() = " + alphabet.size());


    }

}
```

Результат работы программы:

> alphabet = [] alphabet = [A, B, C, D] alphabet = [A, B, C, D, F, G] alphabet.size() = 6 alphabet.get(2) = C alphabet = [A, B, C, D, E, F, G]
>
> -- Contains & Remove -- alphabet.contains("A") = true alphabet.contains("B") = true alphabet.contains("X") = false alphabet.contains("A") = false alphabet.contains("B") = false alphabet = [C, D, E, F, G] alphabet.size() = 5
>
> -- Clear -- alphabet = [] alphabet.size() = 0

## Рекомендации:

- Запустить программу и сравнить результаты;
- Попробовать заменить тип данных, который содержится в списке;