

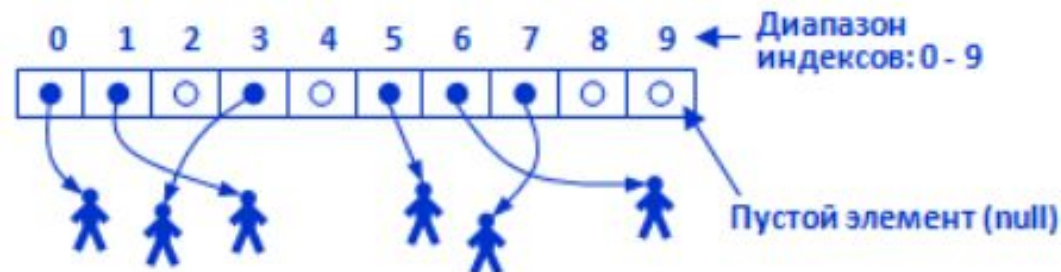
Saraksti - СПИСКИ

Dinamiskā datu
struktūra

Массив из 5 целых чисел



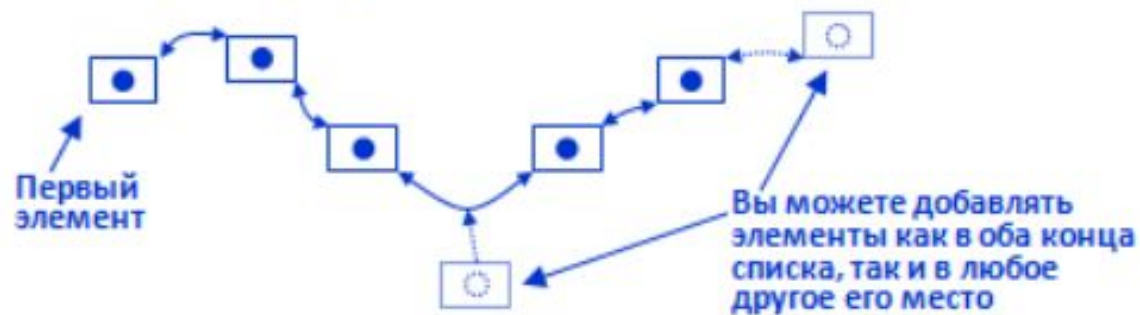
Массив из 10 агентов



ArrayList (коллекция) строк, сейчас содержит 6 элементов



LinkedList (коллекция)



Uzdevums. *Izveidot vārdu sarakstu.*

```
import java.util.ArrayList;

public class saraksts1 {
    public static void main(String[] args) {

        ArrayList <String> name = new ArrayList<>();

        name.add("Emma");
        name.add("Alex");
        name.add("Anton");
        name.add("Anna");
        name.add("Marija");
        name.add("Robet");

        System.out.println(name);

    }
}
```

0

1

2

3

4

5

[Emma, Alex, Anton, Anna, Marija, Robet]

```
import java.util.ArrayList;

public class saraksts1 {
    public static void main(String[] args) {

        ArrayList <String> name = new ArrayList<>();

        name.add("Emma");
        name.add("Alex");
        name.add("Anton");
        name.add("Anna");
        name.add("Marija");
        name.add("Robet");

        for (int i = 0; i < name.size(); i++){
            System.out.println(name.get(i));
        }

    }
}
```

Emma
Alex
Anton
Anna
Marija
Robet

```
import java.util.ArrayList;

public class saraksts1 {
    public static void main(String[] args) {

        ArrayList <String> name = new ArrayList<>();

        name.add("Emma");
        name.add("Alex");
        name.add("Anton");
        name.add("Anna");
        name.add("Marija");
        name.add("Robet");

        name.forEach(str -> System.out.println(str));
    }
}
```

Emma
Alex
Anton
Anna
Marija
Robet

6-09 Uzdevums.

Izveidot sarakstu ar ogu nosaukumiem.

```
import java.util.ArrayList;
```

```
public class Uzd1 {
```

```
    public static void main(String[] args) {
```

```
        ArrayList<String> ogas = new ArrayList<>();
```

```
        ogas.add("avene");
```

```
        ogas.add("mellene");
```

```
        ogas.add("zemene");
```

```
        ogas.add("jāņoga");
```

```
        ogas.add("upene");
```

```
        ogas.add("dzērvene");
```

```
        ogas.add("ērķšķoga");
```

```
        System.out.println(ogas);
```

```
        for(int i=0; i<ogas.size(); i++) {
```

```
            System.out.println(ogas.get(i));
```

```
        }
```

```
        ogas.forEach(str ->System.out.println(str));
```

```
    }
```

```
}
```

[avene, mellene, zemene, jāņoga, upene, dzērvene, ērkšķoga]

avene
mellene
zemene
jāņoga
upene
dzērvene
ērķšķoga

avene
mellene
zemene
jāņoga
upene
dzērvene
ērķšķoga

```
import java.util.ArrayList;

public class saraksts1 {
    public static void main(String[] args) {

        ArrayList <String> name = new ArrayList<>();

        name.add("Emma");
        name.add("Alex");
        name.add("Anton");
        name.add("Anna");
        name.add("Marija");
        name.add("Robet");

        System.out.println(name);

        name.remove(3);

        System.out.println(name);
    }
}
```

```
[Emma, Alex, Anton, Anna, Marija, Robet]
[Emma, Alex, Anton, Marija, Robet]
```

```
name.add(5, "Pol");  
System.out.println(name);
```

0	1	2	3	4	5
[Emma, Alex, Anton, Marija, Robet, Pol]					

6-09 Uzdevums.

Izveidot sarakstu un ogu nosaukumus ievadīt no tastatūras.

```
import java.util.ArrayList;
import java.util.Scanner;

public class Uzd2 {

    public static void main(String[] args) {
        ArrayList<String> ogas = new ArrayList<>();
        Scanner sc = new Scanner(System.in);
        System.out.println("Ievadi ogu nosaukumus");

        while(!sc.hasNext("-")) {
            ogas.add(sc.next());
        }
        System.out.println(ogas);

        ogas.remove("zemene");
        ogas.remove(2);

        System.out.println(ogas.size());

        ogas.add("ķirbis");
        ogas.add(0, "ķirbis");
        System.out.println(ogas);

    }
}
```

Ievadam elementus, kamēr nē «-»

Izdēšam elementu «zemene»
Izdēšam 2 elementu

Izvadām saraksta garumu

Pievienojām elementu «ķirbis»
Pievienojām 0 elementu «ķirbis»

Klases *ArrayList* metodes

.add(n) – pievienot sarakstam elementu n (beigās)

.add(i,n) - .add(0,12) – pirmais elements sarakstā būs 12

.size – saraksta garums

.remove(i) – izdzēst i elementu sarakstā

.contains(n) – vai satur doto elementu n ?

.get(i) – saņemt (получить)

.forEach(...) – katrs saraksta elements

.retainAll(saraksts) – найти совпадения `animalsOne.retainAll(animalsTwo);`

.forEach(mainīgais -> metode) – katram saraksta elementam

.set(index, vērtība) – nomainīt vienu elementu uz otru elementu

Klases *Collections* metodes

.retainAll(saraksts) – **сохранить совпадения**

.replaceAll(saraksta nosaukums, elementa vērtība, elementa vērtība) – **nomainīt vienu elementu uz otru elementu**

```
Collections.replaceAll(animals, "zilonis", "ezis");
```

```
Collections.replaceAll(num, 6, 2);
```

.swap(saraksta nosaukums, elementa index, elementa index) – **samainīt elementus vietām**

```
Collections.swap(num, 0, 1);
```

12-09 Uzdevumi.

Dotajā sarakstā :

- izmainīt 3 elementa vērtību uz kādu citu;
- izmainīt visu elementu vērtības uz kādu citu.
- samainīt elementus vietām;



```
ogas.set(3, "ķirsis");  
System.out.println(ogas);
```



```
Collections.replaceAll(ogas, "ķirsis", "tomāts");  
System.out.println(ogas);
```



```
Collections.swap(ogas, 2, 5);  
System.out.println(ogas);
```

12-09 Uzdevums.

Izveidot veselo skaitļu sarakstu, vērtības ievadot no tastatūras.
Sakārtot tos augošā secībā DIVOS variantos.

```
public class Uzd3 {  
    public static void main(String[] args) {  
  
        ArrayList<Integer> num = new ArrayList<>();  
        Scanner sc = new Scanner(System.in);  
        System.out.println("Ievadi veselus skaitļus:");  
  
        while(sc.hasNextInt()) {  
            num.add(sc.nextInt());  
        }  
        System.out.println(num);  
        sc.close();  
  
        Collections.sort(num);  
        System.out.println(num);  
  
        num.sort((Integer sk1, Integer sk2)->{  
            return sk1-sk2;  
        });  
  
        System.out.println(num);  
  
    }  
}
```

Elementu kārtošana augošā secībā

Elementu kārtošana augošā secībā, izveidojot komparatoru
(это объект, кот. реализует логику сравнения двух элементов списка)

Saraksta elementu kārtošana

Var būt veidota dažādi:

Collections.sort(saraksta nosaukums)

```
Collections.sort(num);
```

saraksta nosaukums(comparator)

Objekts, kurš realizē divu elementu salīdzinājumu

Veselo skaitļu kārtošana augošā secībā

```
num.sort((Integer n1, Integer n2) -> {  
    return n1 - n2;  
});
```

Возвращает одно из значений:
-1 или 0 или 1

Saraksta elementu kārtošana

Var būt veidota dažādi:

Collections.sort(saraksta nosaukums)

```
Collections.sort(animals);
```

saraksta nosaukums(comparator)

Objekts, kurš realizē divu elementu salīdzinājumu

String tipa elementu kārtošana no A līdz Z

```
animals.sort((String str1, String str2) -> {  
    return str1.compareTo(str2);  
});
```

```
animals.sort((String str1, String str2) -> {  
    return str1.compareToIgnoreCase(str2);  
});
```

Возвращает одно из значений:
-1 или 0 или 1

Saraksta elementu kārtošana

Var būt veidota dažādi:

Collections.reverse(saraksta nosaukums)

```
Collections.sort(num, Collections.reverseOrder());
```

saraksta nosaukums(comparator)

Veselo skaitļu saraksts

```
num.sort((Integer n1, Integer n2) -> {  
    return n2 - n1;  
});
```

Saraksta elementi tiek sakārtoti otrādi

Saraksta elementi sakārtoti no Z līdz A

Objekts, kurš realizē divu elementu salīdzinājumu

Возвращает одно из значений:
-1 или 0 или 1

13.09.- Saraksta elementu kārtošana

```
ArrayList<Integer> list = new ArrayList<>();
Scanner sc = new Scanner(System.in);
System.out.println("Ievadi veselus skaitļus:");

while(sc.hasNextInt()) {
    list.add(sc.nextInt());
}
System.out.println(list);
sc.close();

list.sort((Integer sk1, Integer sk2)->{
    return sk1-sk2;
});
System.out.println(list);

list.sort((Integer sk1, Integer sk2)->{
    return sk2-sk1;
});
System.out.println(list);
```

Ievadi veselus skaitļus:
3 4 8 1 2 9 5 0 -
[3, 4, 8, 1, 2, 9, 5, 0]
[0, 1, 2, 3, 4, 5, 8, 9]
[9, 8, 5, 4, 3, 2, 1, 0]

Elementu kārtošana augošā secībā, izveidojot **komparatoru** (это объект, кот. реализует логику сравнения двух элементов списка и возвращает одно из трех значений: -1, 0 или +1)

$sk1 < sk2$	-1
$sk1 > sk2$	+1
$sk1 == sk2$	0

```

list.sort((Integer sk1, Integer sk2) ->{
    if(sk1%2==0 && sk2%2!=0) {
        return -1;
    }else if(sk1%2!=0 && sk2%2==0) {
        return 1;
    }else {
        return 0;
    }
});
System.out.println(list);

list.sort((Integer sk1, Integer sk2) ->{
    if(sk1%2==0 && sk2%2!=0) {
        return 1;
    }else if(sk1%2!=0 && sk2%2==0) {
        return -1;
    }else {
        return 0;
    }
});
System.out.println(list);

```

13.09.- Saraksta elementu kārtošana:

Vispirms izvadīt pāra skaitļus, pēc tam – nepāra.

Un otrādi, vispirms, nepāra un tad – pāra.

3 4 8 1 2 9 5 0

[8, 4, 2, 0, 9, 5, 3, 1]

[9, 5, 3, 1, 8, 4, 2, 0]

sk1 < sk2 -1

sk1 > sk2 +1

sk1 == sk2 0

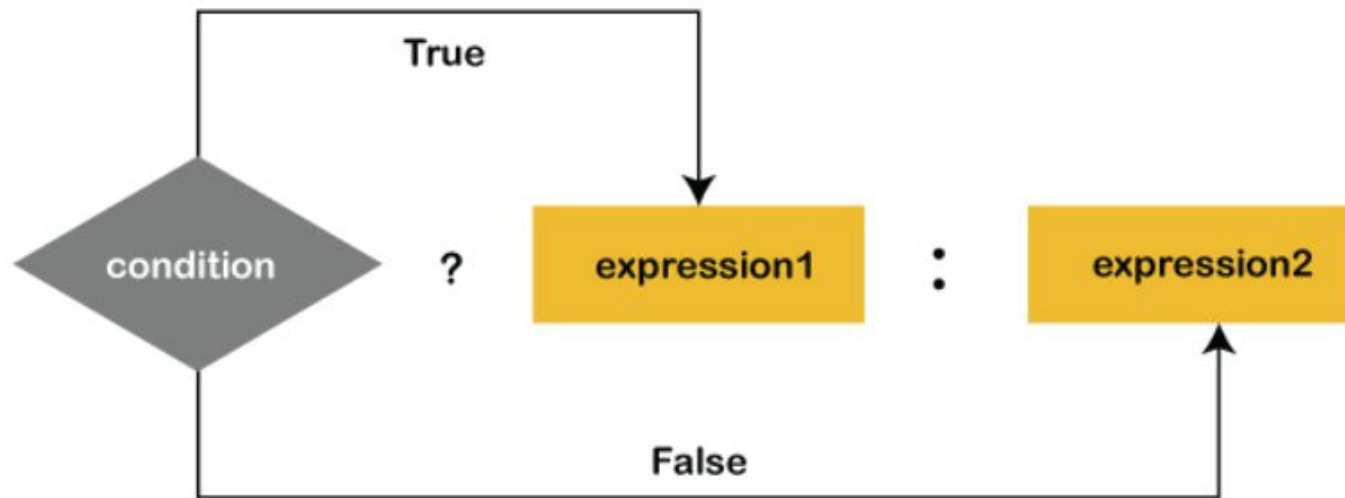
Operator	Example	Same As
=	x = 5	x = 5
+=	x += 3	x = x + 3
-=	x -= 3	x = x - 3
*=	x *= 3	x = x * 3
/=	x /= 3	x = x / 3
%=	x %= 3	x = x % 3
&=	x &= 3	x = x & 3
=	x = 3	x = x 3
^=	x ^= 3	x = x ^ 3
>>=	x >>= 3	x = x >> 3
<<=	x <<= 3	x = x << 3

<https://proglang.su/java/operators>

Тернарный оператор - Trīskāršs operātors

“Тернарный” в переводе означает “**тройной**”. Это альтернатива условному оператору **IF – ELSE**

tips **mainīgais = nosacījums ? komanda1 : komanda2**



Тернарный оператор - Trīskāršs operātors

Uzdevums.

Ja skaitlis dalās ar divi, tad skaitlis ir pāra, citādi – nepara.

```
if(sk%2==0){  
    System.out.println("Pāra skaitlis");  
} else {  
    System.out.println("Nepāra skaitlis");  
}
```

```
String text = sk%2==0?"Pāra skaitlis":"Nepāra skaitlis";  
System.out.println(text);
```

Тернарный оператор - Trīskāršs operātors

Uzdevumi.

Ja atzīme ir <4, tad **ieskaite nav nokārtota**, citādi – **ir nokārtota**.

```
String text = atzime<4 ? "Ieskaite nav nokārtota" : "Ieskaite nokārtota";  
System.out.println(text);
```

No diviem skaitļiem izvadīt **lielāko**.

```
int rez = a>b ? a : b;
```

```
String rez = a>b? "Lielakais: "+a : "Lielakais: "+b;
```

19.09. Visu saraksta elementu pārskats

```
for (int i = 0; i < name.size(); i++){  
    System.out.println(name.get(i));  
}
```

```
int k=0;  
for (int i=0; i<num.size(); i++) {  
    if(num.get(i)<0) {  
        k++;  
    }  
}
```

```
animals.forEach(str-> System.out.println(str));
```

lapsa
ezis
vilks
lācis

```
animals.forEach(str -> {  
    if(str.length()<5) {  
        System.out.println(str);  
    }  
});
```

[ezis, bruņurupucis, lācis, gailis, vilks]
ezis

```
ArrayList<String>temporary = new ArrayList<>();  
animals.forEach(str -> {  
    if(str.length()<4) {  
        temporary.add(str);  
    }  
});
```



19.09. Aritmētiskās darbības ar saraksta elementiem

```
int sum = 0;
for (Integer sk: num) {
    sum+=sk;
}
System.out.println("Elementu summa ir "+sum);
```

```
int sumPara = 0;
for (Integer sk: num) {
    if(sk%2==0) {
        sumPara +=sk;
    }
}
System.out.println("Para elementu summa ir "+sumPara);
```

20.09. Uzdevums

Veikals «Jautrā vāga» strādā dažas nedēļas (sākot no pirmdienas). Katru dienu veikalā piegādā svaigus burkānus un nākošās nedēļas sākuma veic iepriekšējās nedēļas kopsavilkumu.

Ievadi katras dienas piegādātu burkānu svaru.

Analizē «Jautrās vāgas» darbu:

1. Cik nedēļas un dienas strādāja veikals.
2. Kāds ir kopējais piegādāto burkānu svars.
3. Aprēķini mazāko un lielāko burkānu piegādes svaru.
4. Aprēķini dienas vidējo piegādes svaru.
5. Cik dienas piegādāja burkānus vairāk pa 5kg.
6. Cik dienas nepiegādāja burkānus.
7. Sakārto burkānu svarus augošā secībā.
8. Cik dienas burkānus piegādāja vairāk nekā dienas vidējais svars.

