

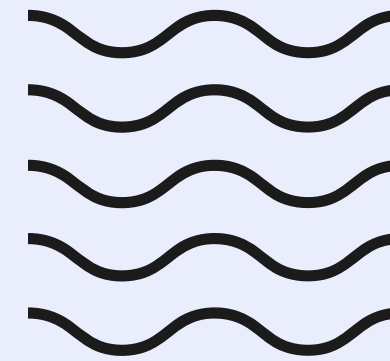
Comparable Comparator sorting



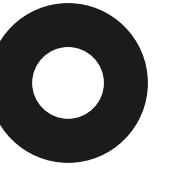
ПОДГОТОВИЛ: АЛИШЕР ХАМИДОВ

План лекции

Основные моменты:



1. Сортировка
2. Comparable
3. Comparator



Зачем нам нужна сортировка?



- **Java Comparable interface** используется, чтобы упорядочить объекты какого-либо класса.
- Интерфейс находится в пакете `java.lang` и содержит только один метод **`compareTo(Object o)`**.
- С помощью этого интерфейса можно реализовать только какой-то один способ сортировки





public int compareTo(Object obj): используется для того, чтобы сравнивать текущий объект (**this**) с указанным (**obj**) объектом.

Метод возвращает

- **positive integer**, если текущий объект больше указанного
- **negative integer**, если текущий объект меньше указанного
- **0**, если объекты равны по сравниваемому полю



Пример реализации метода compareTo

```
public int compareTo(Student st){  
    if(age==st.age)  
        return 0;  
    else if(age>st.age)  
        return 1;  
    else  
        return -1;  
}  
}
```



Метод класса Collections, использующийся для сортировки элементов.

public void sort(List list): используется для сортировки элементов листа.

Элементы листа должны быть Comparable типа.



Пример сортировки

```
ArrayList<Student> stundents=new ArrayList<Student>();  
stundents.add(new Student(101, "Mark", 23));  
stundents.add(new Student(106, "John",27));  
stundents.add(new Student(105, "Bob", 21));  
Collections.sort(stundents);  
// студенты отсортированы по возрасту
```




Если так получилось, что:

- класс не имплементирует Comparable
- или нужно сортировать по разным полям или с разной логикой (например, по-убыванию), - используется класс Comparator.

Пример компаратора

```
import java.util.*;
class AgeComparator implements Comparator <Student> {
    public int compare(Student student){
        if(this.age==student.age)
            return 0;
        else if(this.age>student.age)
            return 1;
        else
            return -1;
    }
}
```