Обобщения или generics

class Account<T>{

private T id;

private int sum;

Account(T id, int sum){

this.id = id;

this.sum = sum;

}

public T getId() { return id; }

public int getSum() { return sum; }

public void setSum(int sum) { this.sum = sum; }

}

Обобщенные интерфейсы

public class Program{

public static void main(String[] args) {

Account<String> acc1 = new Account<String>("1235rwr", 5000);

Account<String> acc2 = new Account<String>("2373", 4300);

System.out.println(acc1.getId());

System.out.println(acc2.getId());

}

}

interface Accountable<T>{

T getId();

int getSum();

void setSum(int sum);

}

class Account<T> implements Accountable<T>{

private T id;

private int sum;

Account(T id, int sum){

this.id = id;

this.sum = sum;

}

public T getId() { return id; }

public int getSum() { return sum; }

public void setSum(int sum) { this.sum = sum; }

}

Обобщенные методы

public class Program{

public static void main(String[] args) {

Printer printer = new Printer();

String[] people = {"Tom", "Alice", "Sam", "Kate", "Bob", "Helen"};

Integer[] numbers = {23, 4, 5, 2, 13, 456, 4};

printer.<String>print(people);

printer.<Integer>print(numbers);

}

}

class Printer{

public <T> void print(T[] items){

for(T item: items){

System.out.println(item);

}

}

}

Несколько параметров

public class Program{

public static void main(String[] args) {

Account<String, Double> acc1 = new Account<String, Double>("354", 5000.87);

String id = acc1.getId();

Double sum = acc1.getSum();

System.out.printf("Id: %s Sum: %f \n", id, sum);

}

}

class Account<T, S>{

private T id;

private S sum;

Account(T id, S sum){

this.id = id;

this.sum = sum;

}

public T getId() { return id; }

public S getSum() { return sum; }

public void setSum(S sum) { this.sum = sum; }

}