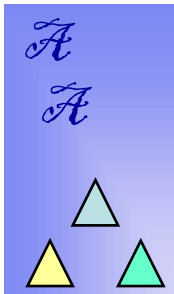


Generics.
Колекции от данни.

Аделина Алексиева

Generics

- Подобрение към синтаксиса на класове, което ни позволяват да специфицираме класа за даден тип или набор от типове
- Синтаксис:

```
public class List< E > {
...
public void add( E element ) { ... }
public E get( int i ) { ... }
}
```

Generics

Без generics:

```
Object obj=new Object();
List ls = new LinkedList();
ls.add(new String("alabala"));
ls.add(new Student("Ivan Ivanov", "fn555555"));
ls.add(obj);
```

```
List<String> ls = new LinkedList<String>();
ls.add(new String("alabala"));
ls.add(new Student("Ivan Ivanov",
  "fn555555")); //ГРЕШКА! (приема само от
  тип String!)
```

Предимства

- Кодът с generics е по сигурен и по ясен:

```
Student s = (Student)ls.get(i); // изключение ClassCastException!
```

- Така се елиминират “несигурни” конвертирания, и се избягват излишни скоби.

```
Student student = ls.get(i);
```

- Най-важното е, че част от спецификацията е преместена от коментара, към самата сигнатура на колекцията:

```
List ls = new LinkedList(); // contain list of students

List<Student> ls = new LinkedList<Student>();
```

Generics – изтриване на типа

- ***изтриване на типа (type erasure)*** – след успешна компилация на кода в клас информацията за generic е изтрита от компилатора.

Колекции от данни

- Колекциите са обекти, които сочат към група от обекти.
- Колекциите съдържат референции към данни от тип обект.
- Всякакъв тип обект може да се съхранява в колекция.

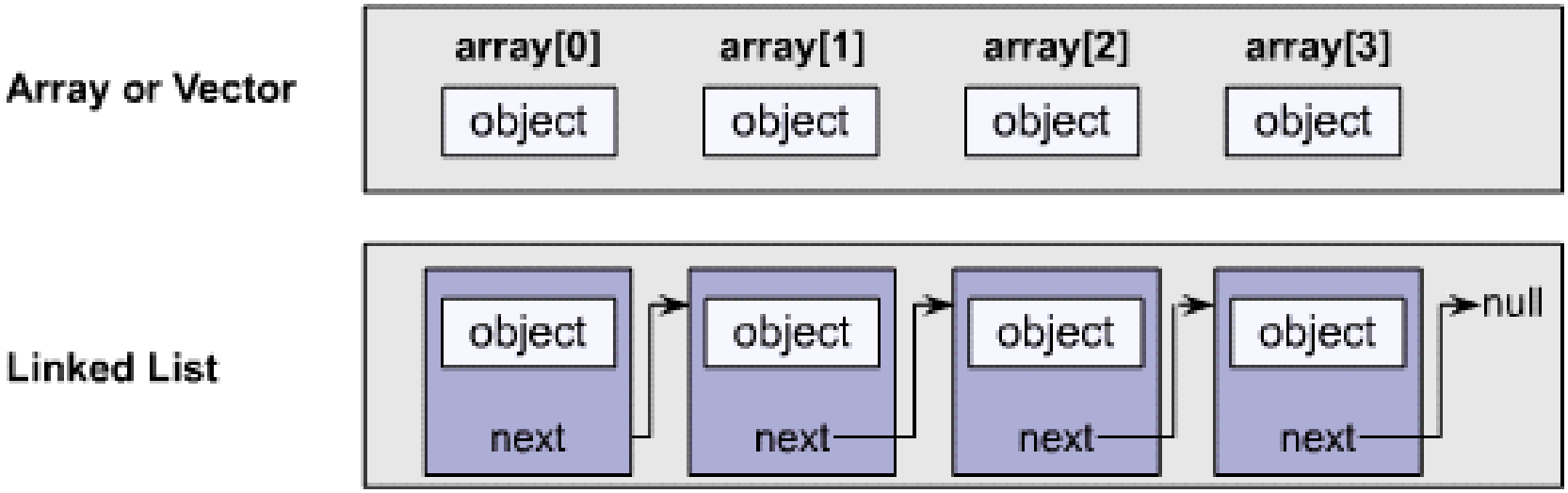
Collections Framework

- **интерфейси**
- **имплементация**
- **алгоритми**

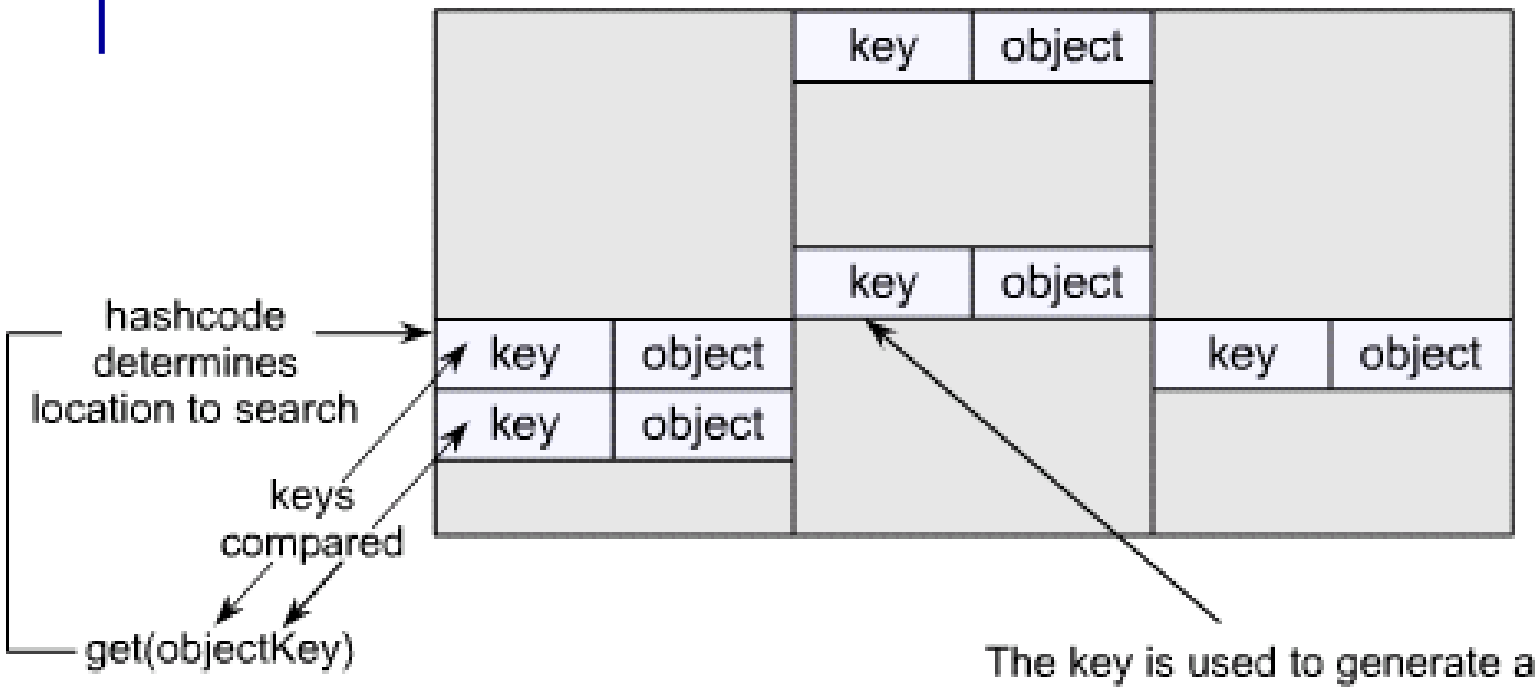
Технология на съхраняване на колекцията

- Масив (Array):
  - Фиксиран размер, бързо и ефикасно за достъп, трудно за модифициране.
- Свързан списък (Linked List):
  - Елементите имат референция за упътване към следващия елемент, лесен за промяна, бавен за претърсване.
- Дърво (Tree):
  - Лесен за промяна, съхранява елементите в ред.
- Хеш таблица (Hashtable):
  - Използва се за индексиране на ключ който идентифицира елементите. Елементите са извличат от хеш таблицата чрез използване на ключ на елемента.

Масиви и свързан списък



Хеш таблица

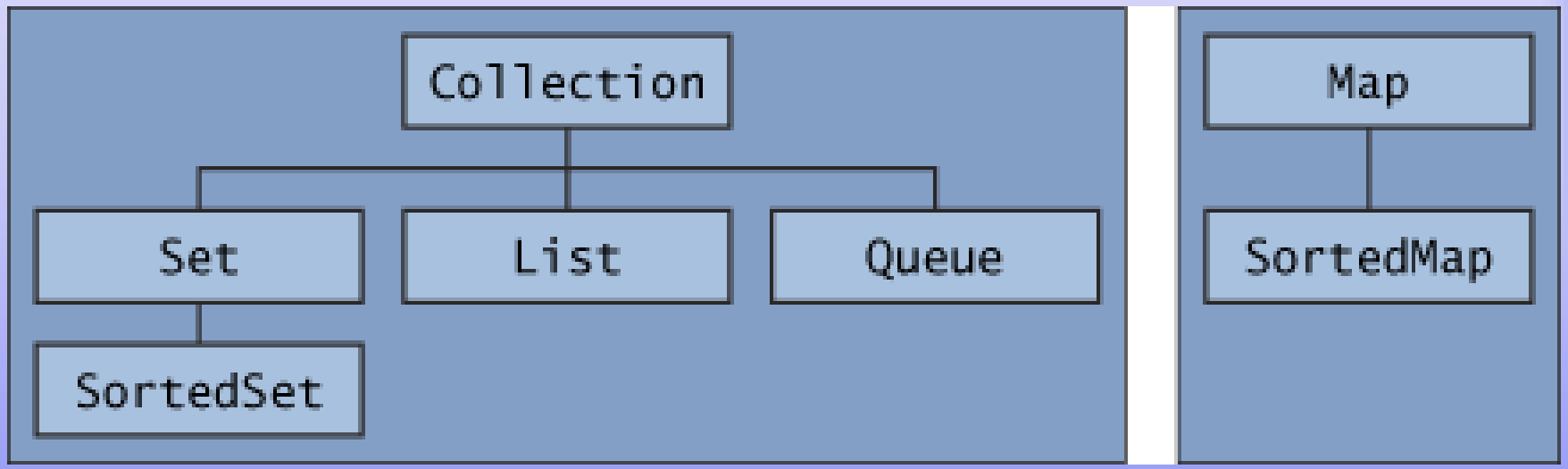


Retrieving an object requires a key to be supplied. A hashcode is generated and the key (or keys) at the location determined by the hashcode is compared with the supplied key.

Видове колекции

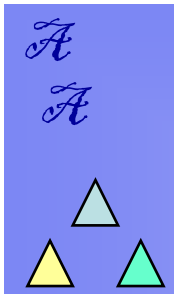
- Collection
  - Прост контейнер от неподредени обекти. Повторенията са позволени.
- List
  - Контейнер от подредени елементи. Повторенията са позволени.
- Set
  - Неподредена колекция от обекти, в която повторенията не са позволени.
- Map
  - Колекция от ключ/стойност по двойки. Ключът се използва за индекс на елемента. Повторение на ключове не се допуска.

Интерфейси за колекции



Интерфейс Collection

```
public interface Collection<E> extends Iterable<E> {
        // Basic operations
        int size();
        boolean isEmpty();
        boolean contains(Object element);
        boolean add(E element);
        //optional boolean remove(Object element);
        //optional Iterator<E> iterator();
```



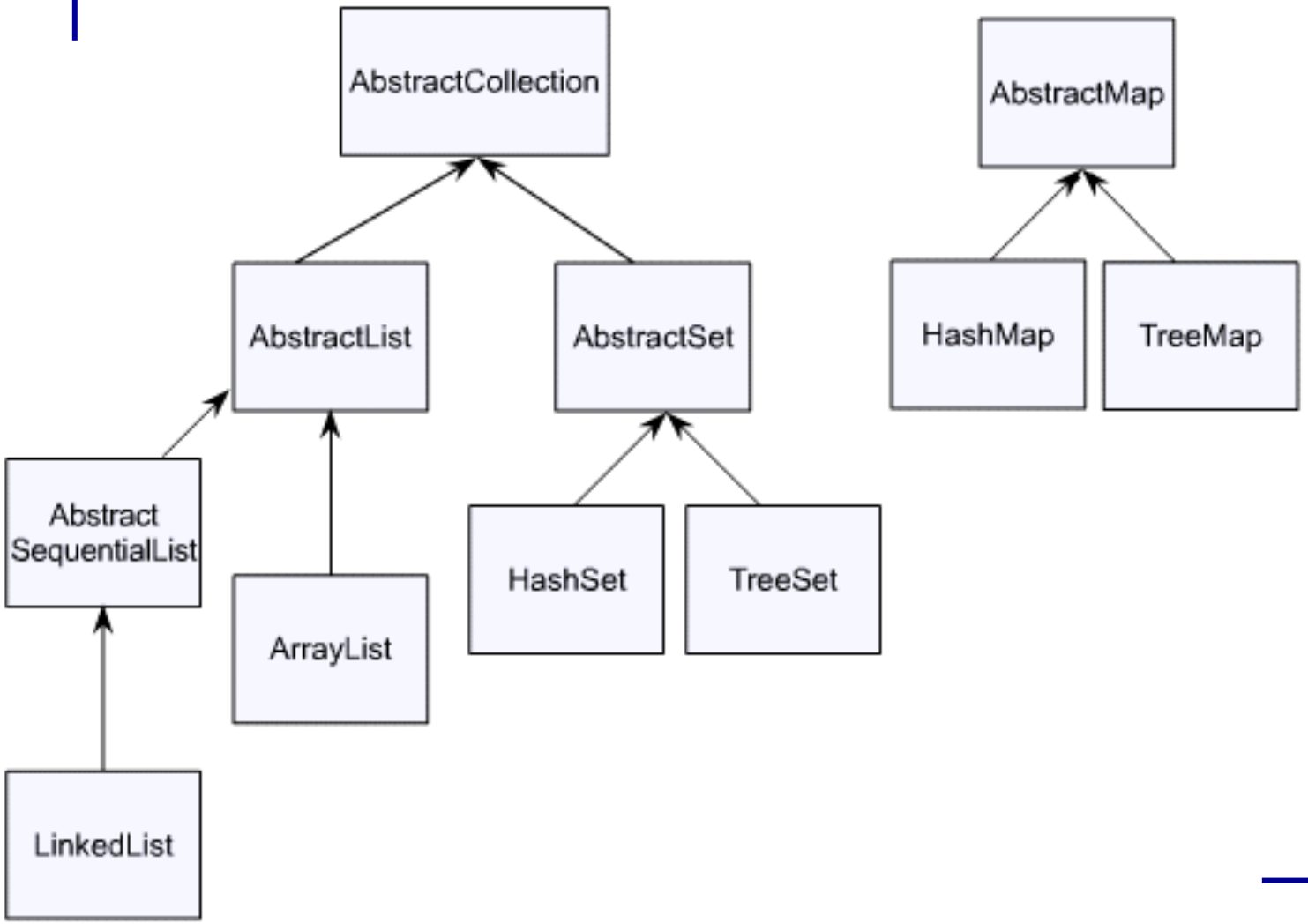
```
// Bulk operations
boolean containsAll(Collection<?> c);
boolean addAll(Collection<? extends E> c);
boolean removeAll(Collection<?> c);
boolean retainAll(Collection<?> c);
void clear();

// Array operations
Object[] toArray();
<T> T[] toArray(T[] a); }
```

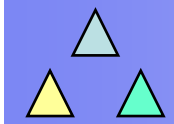
Интерфейс Set

```
public interface Set<E> extends Collection<E> {
        // Basic operations
        int size();
        boolean isEmpty();
        boolean contains(Object element);
        boolean add(E element);
        boolean remove(Object element);
        Iterator<E> iterator();
```

```
// Bulk operations
    boolean containsAll(Collection<?> c);
    boolean addAll(Collection<? extends E> c);
    boolean removeAll(Collection<?> c);
    boolean retainAll(Collection<?> c);
    void clear();
// Array Operations
    Object[] toArray();
     <T> T[] toArray(T[] a);
    }
```

Класове за колекции





			Collection Properties			
Collection Class	**Storage Technologies**	**Collection Interface**	**Sorted**	**Ordered**	**Duplicates**	**Key**
LinkedList	Linked-List	List		x	x	
ArrayList	Array	List		x	x	
Vector	Array	List		x	x	
HashSet	HashTable	Set				
TreeSet	Tree	SortedSet	x			
HashMap	HashTable	Map				x
TreeMap	Tree	SortedMap	x			x
Hashtable/ Properties	HashTable	Map				x

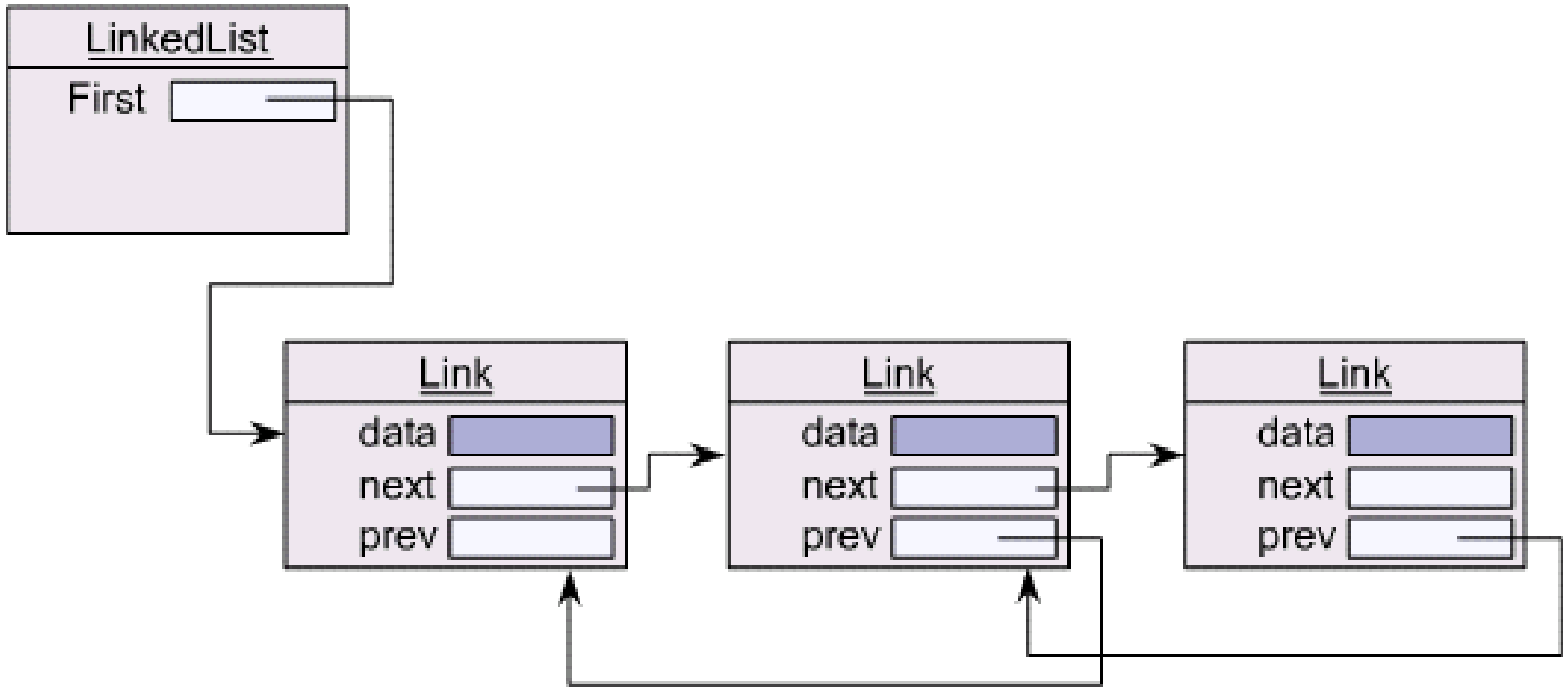
Set (множество) обекти

- **HashSet** имплементира **Set** интерфейс. Повторенията са позволени.
- **TreeSet** имплементира **OrderedSet**. Включва методи, които използват предимството на подреждането:
  - TreeSet.first()
  - TreeSet.last()
  - TreeSet.headSet()
  - TreeSet.subSet()

List (списък) обекти

- Vector
  - Колекция от обекти. Имплементират **List** интерфейс. Съдържат се в ред в който са добавени в колекцията.
- ArrayList
  - Променят си размера и са подредени
- LinkedList
  - Всеки елемент е референция към следващия.

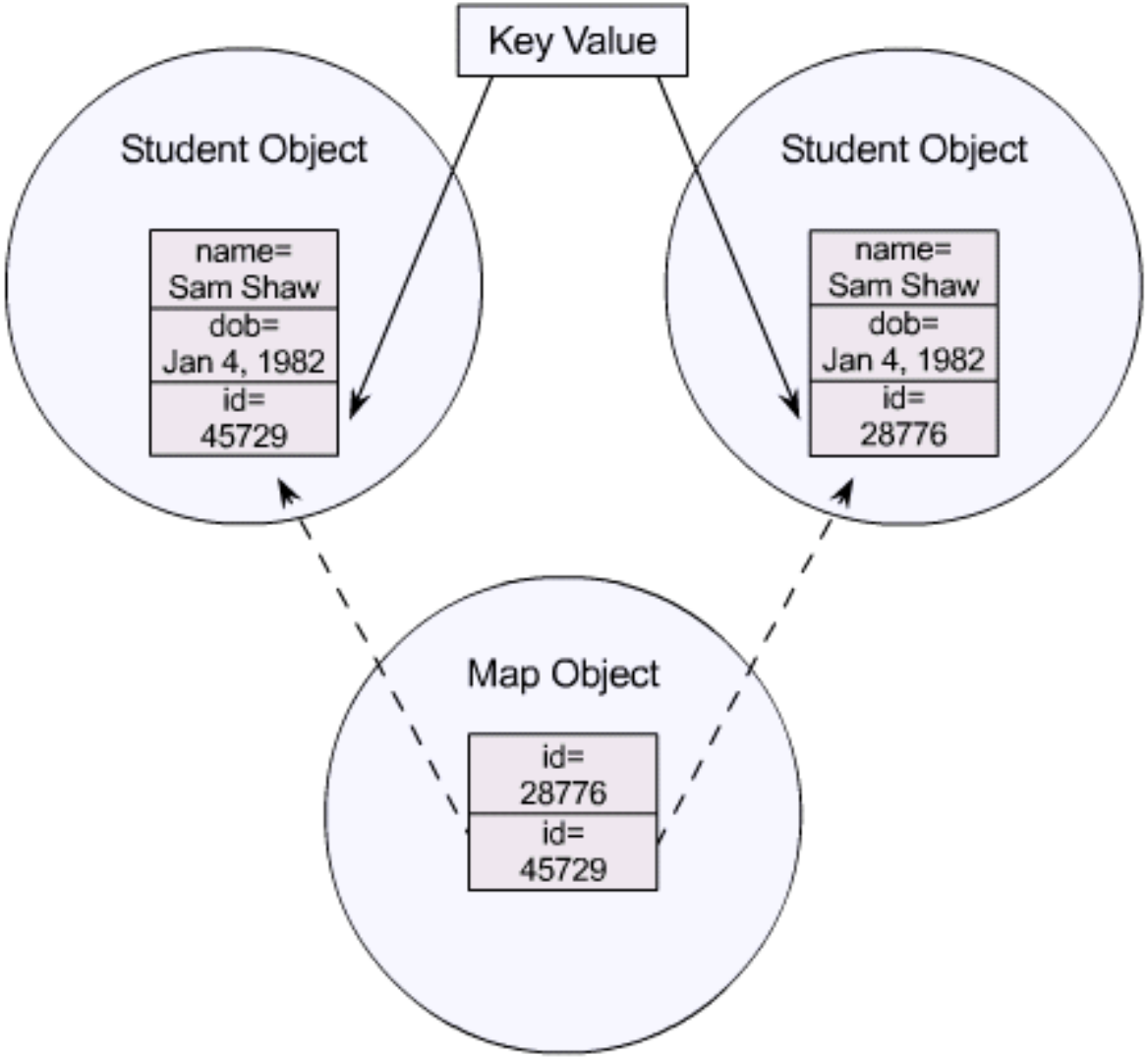
List (списък) обекти

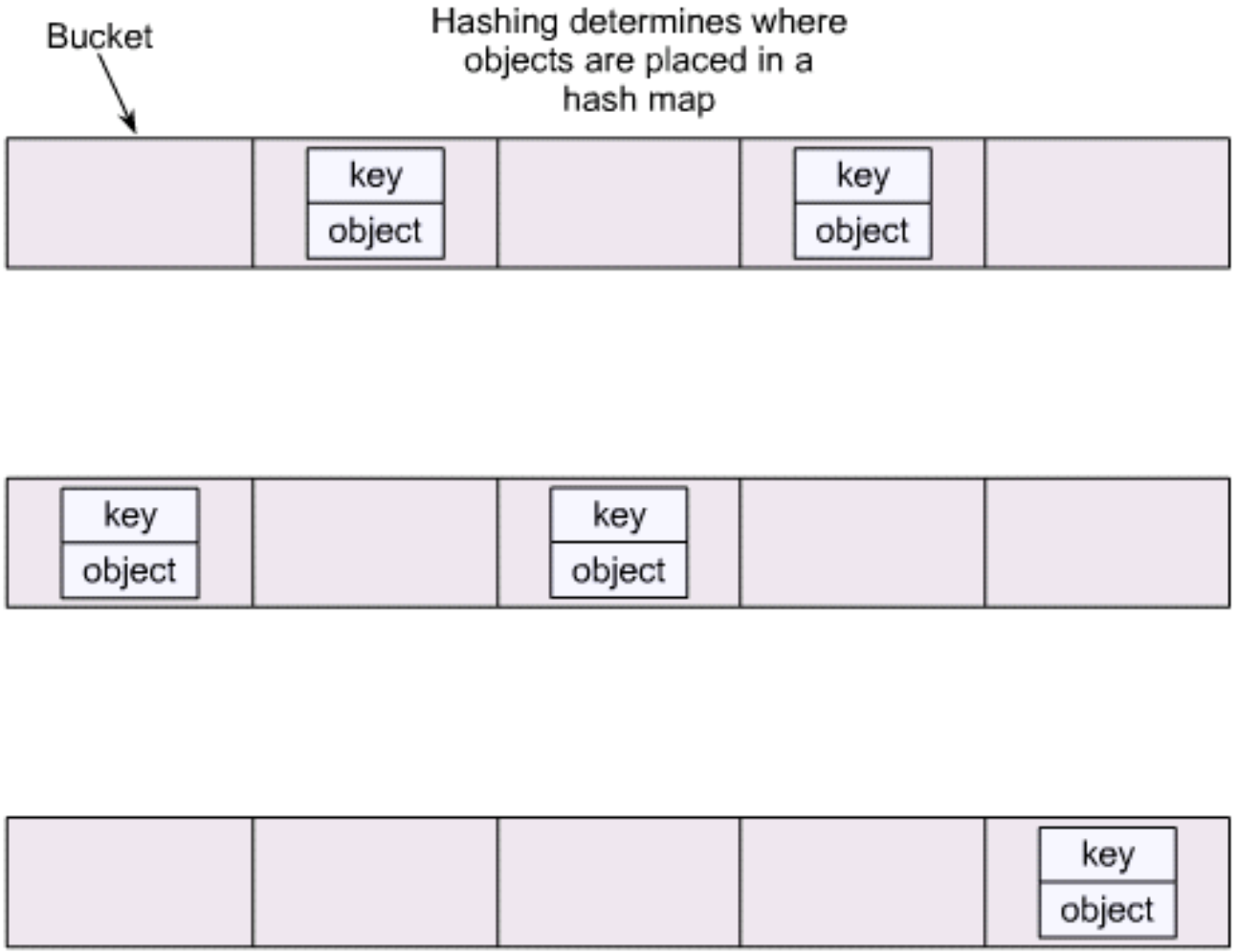


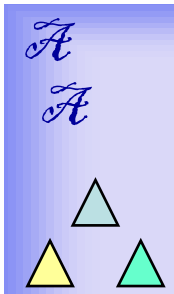
Map обекти

- Използват уникален ключ за достъп до елемент.
- **Ключът** указва къде е съхранен елемента.
- Методът **hashCode()** от класа **Object** може да се пренапише за да се осигури уникален ключ.

Пример


Key Value
Student Object
name=
Sam Shaw
dob=
Jan 4, 1982
id=
45729
Student Object
name=
Sam Shaw
dob=
Jan 4, 1982
id=
28776
Map Object
id=
28776
id=
45729


Bucket
Hashing determines where objects are placed in a hash map
key
object
key
object
key
object
key
object
key
object



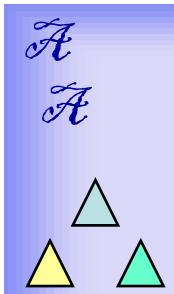
Обхождане на колекции

- конструкцията for-each
- с използване на Iterators

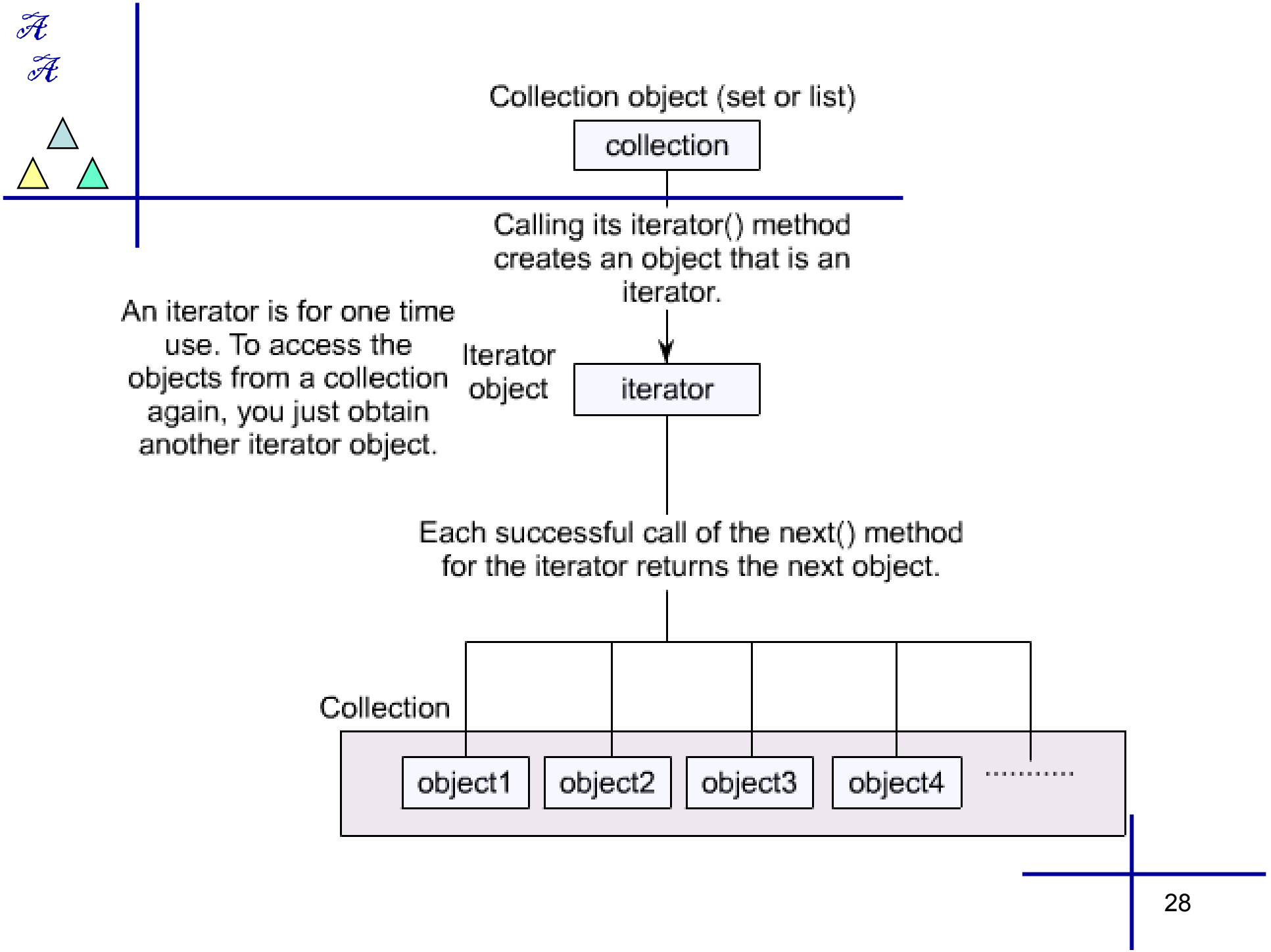
for-each

```
for (Object o : collection)
        System.out.println(o);
```

```
void cancelAll(Collection<TimerTask> c) {
    for (TimerTask t : c) t.cancel();
}
```



- Поддържа метод за обхождане на колекции.
- Обхождането (итерирането) по множество (set) е неопределено.
- Използва се класа **ListIterator** за обхождане на списък.


Collection object (set or list)
collection
Calling its iterator() method creates an object that is an iterator.
An iterator is for one time use. To access the objects from a collection again, you just obtain another iterator object.
Iterator object
iterator
Each successful call of the next() method for the iterator returns the next object.
Collection
object1
object2
object3
object4

Методи

- public boolean hasNext()
- public Object next()
- public void remove()

Сортиране и премесване на обекти от списък

- Колекциите имат методи за сортиране:
  - Collection.sort() – сортира целия списък
  - Collection.reverse() – обръща списъка
- Преместване на елементи от списък
  - Collection.shuffle() – разбъркване на колекцията
  - Collection.shuffle(Random r)

Ресурси

Колекции от данни.

Ресурси към темата:

http://java.sun.com/docs/books/tutorial/collections/index.html

Generics

Ресурси към темата:

http://java.sun.com/docs/books/tutorial/java/generics/index.html