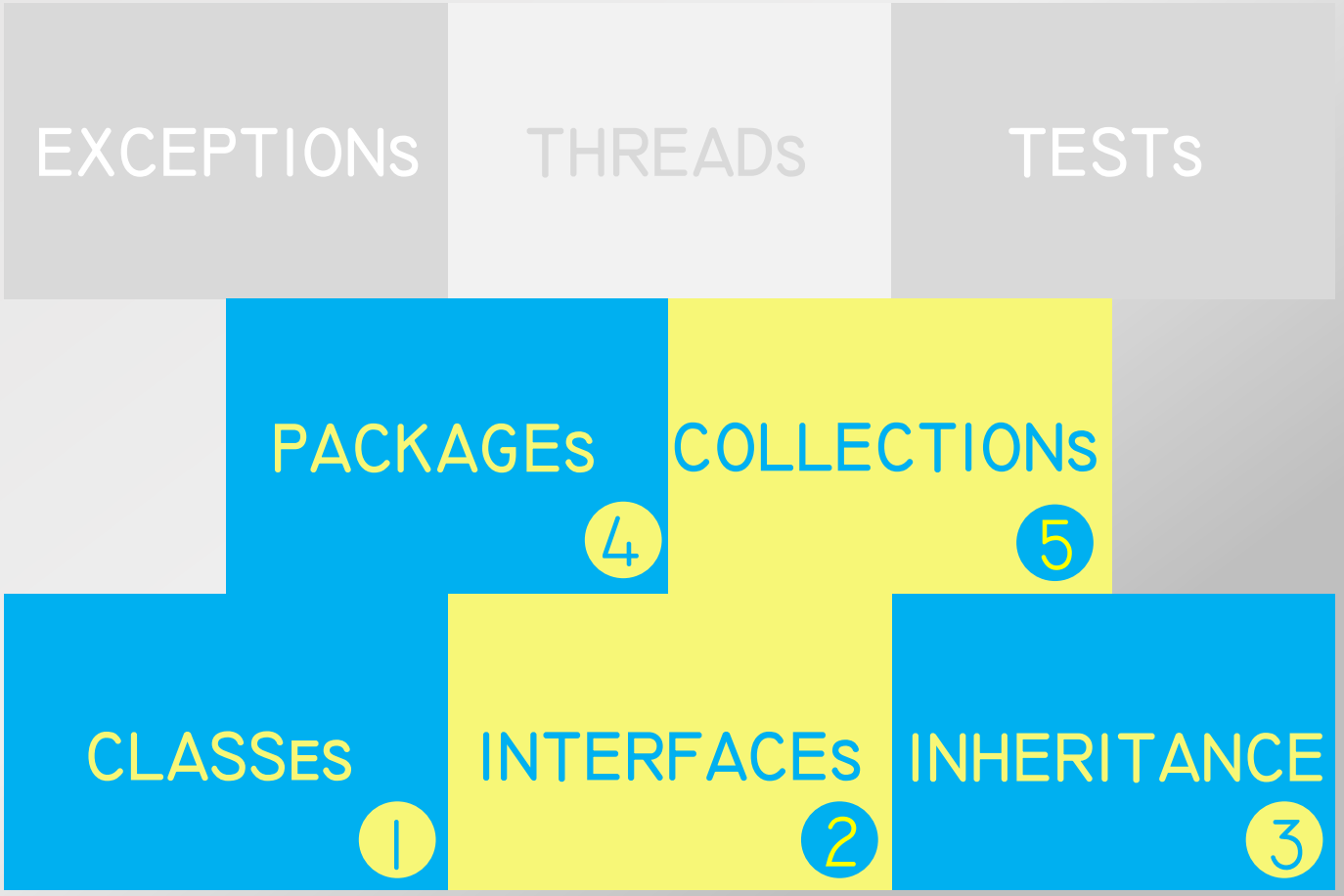
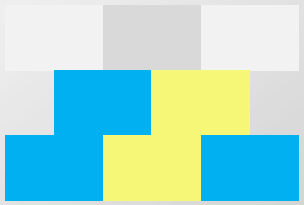
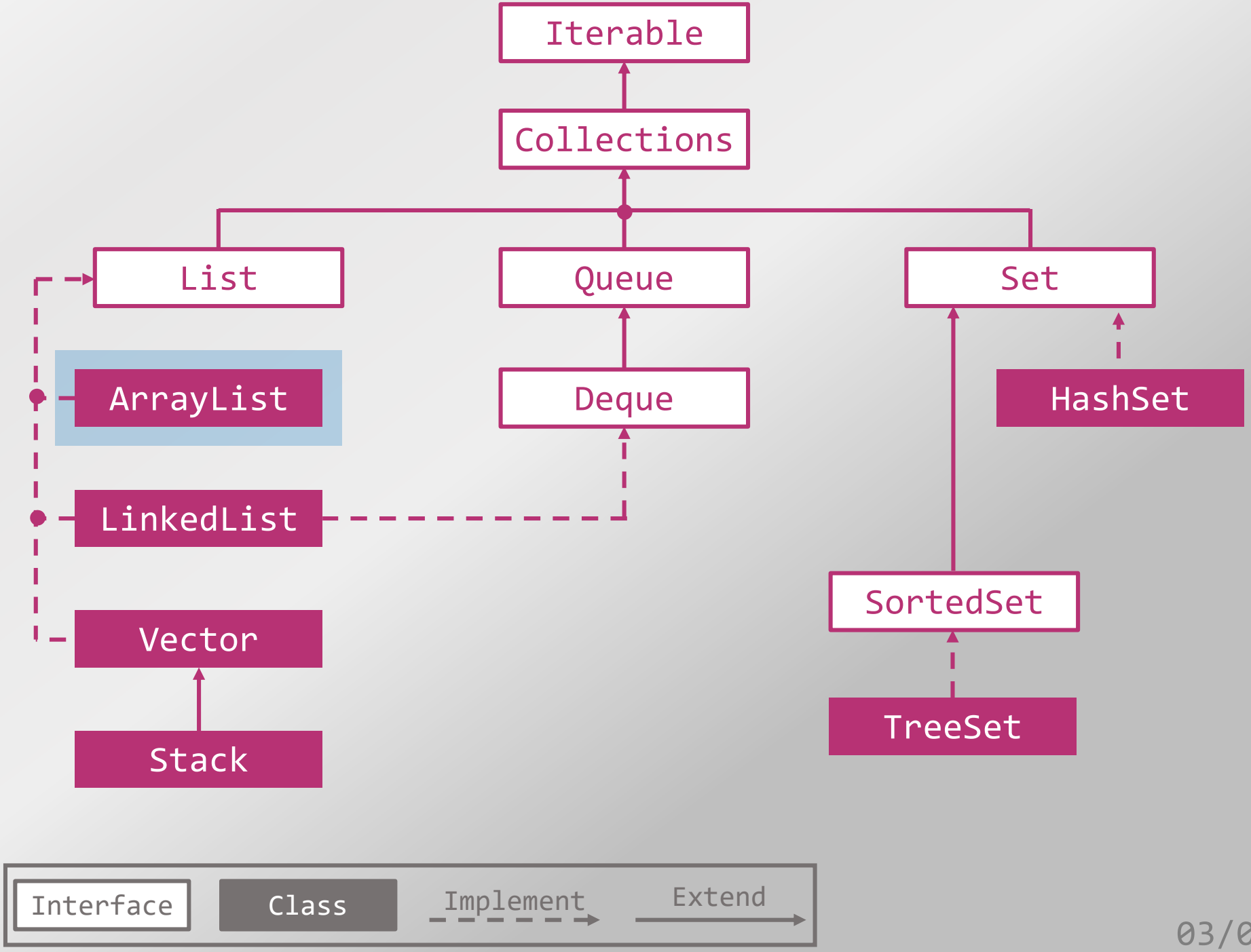
BASICS of Java



COLLECTIONS

THE <ArrayList> CLASS




Iterable
Collections
List
Queue
Set
ArrayList
Deque
HashSet
LinkedList
SortedSet
Vector
TreeSet
Stack
Interface
Class
Implement
Extend

Array VS ArrayList

(FIXED SIZED ARRAY)

(DYNAMIC SIZED ARRAY)

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

DATA TYPE — NAME — DATA TYPE

```
int arr[] = new int[5];
```

DATA TYPE — KEYWORD — SIZE

```
ArrayList<Integer> arrL =
        new ArrayList<Integer>();
```

KEYWORD — DATA TYPE

```
String arr[] = new String[]{"zero",
                            "jeden",
                            "dwa"};
```

```
ArrayList<String> arrL =
        new ArrayList<String>();
arrL.add("zero");
arrL.add("jeden");
arrL.add("dwa");
```

Example:

```java
public class Book {
    //fields
    private String title;
    private String author;
    //constructor
    Book(String title, String author){
        this.title = title;
        this.author = author;
    }
    //methods
    public String getTitle() {
        return title;
    }
    public void setTitle(String title) {
        this.title = title;
    }
    public String getAuthor() {
        return author;
    }
    public void setAuthor(String author) {
        this.author = author;
    }
}
```

```java
import java.util.ArrayList;

public class Library {
    //fields
    static ArrayList<Book> books = new ArrayList<>();
    //methods
    public static int numberOfBooks(){
        return books.size();
    }
    public static void addBook(String title, String author){
        Book newBook = new Book(title,author);
        books.add(newBook);
    }
    public static String removeBook(String title){
        String message = "There is no such book";
        for(int i=0; i<books.size();i++){
            if(books.get(i).getTitle().equals(title)){
                books.add(books.remove(i));
                message = "The book <"+title+ "> is removed";
                break;
            }
        }
        return message;
    }
}
```

! Take a look at this video: *LodówkaApp.mp4*

1-4

☐ CREATE A CLASS CALLED <Jedzenie> THAT CONTAINS TWO FIELDS: THE FIRST FIELD OF THE String CLASS CALLED <nazwa> AND THE SECOND FIELD OF THE int DATATYPE CALLED <ilość>.

☐ ADD A CONSTRUCTOR THAT TAKES AS PARAMETERS THE FOOD NAME AND ITS AMOUNT AND THEN ASSIGNS THEM TO <nazwa> AND <ilość>.

☐ CREATE A CLASS CALLED <Lodówka> THAT CONTAINS AN ARRAY OF OBJECTS OF THE <Jedzenie> CLASS. USE THE <ArrayList> CLASS TO KEEP THESE OBJECTS.

☐ ADD THREE METHODS TO THE <Lodówka> CLASS:

- dodaj() → TO ADD A PRODUCT TO THE FRIDGE,
- weź() → TO TAKE A PRODUCT FROM THE FRIDGE,
- wyświetl() → TO PRINT WHAT IS IN THE FRIDGE.

☐ CREATE A CLASS CALLED <LodówkaApp> THAT CONTAINS THE <main> METHOD.

5-7

☐ IN THE MAIN PROGRAM ASK A USER TO CHOOSE AN OPERATION "dodaj", "weź","wyświetl", OR "zamknij". DEPENDING ON THE USER'S INPUT CALL dodaj(), weź() OR wyświetl() METHOD. WHEN THE USER ENTERS "zamknij", END THE PROGRAM.

☐ USING THE **while** LOOP KEEP ASKING THE USER TO CHOOSE AN OPERATION UNTIL S/HE ENTERS "zamknij".

```
int i = 5;
while(i > 0){
    i--;
    // your code
}
```

```
while(true){
    // your code
}
```

INFINITE LOOP