Faculty Civil Engineering
Chair of Intelligent Technical Design
Prof. Dr.-Ing. Christian Koch

Bauhaus-Universität Weimar

Object-oriented Modeling and Programming in Engineering (OOMPE)

Winter semester 2018-19

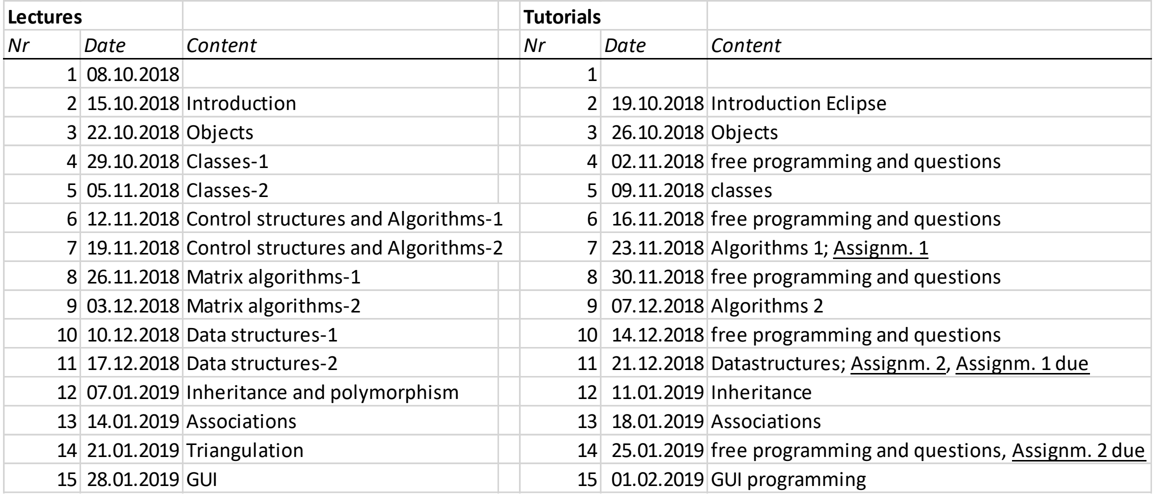
06 – Data structures

Organisation

- Convenor/ contributors
  - Christian Koch, c.koch@uni-weimar.de, M13A room 208
  - Mathias Artus, mathias.artus@uni-werimar.de, M13A room 204
- Time
  - Lectures: Mondays, 15:15-16:45, Coudraystr. 13B, room 210
  - Tutorials: Fridays, 9:15-10:45, Marienstr. 7, Luna blue+grey (computer pool)
- Moodle
  - Learning material related to lectures and tutorials, messages
- Examination
  - Passed 2 coursework assignments
  - Written exam (100%)

Organisation

- Schedule

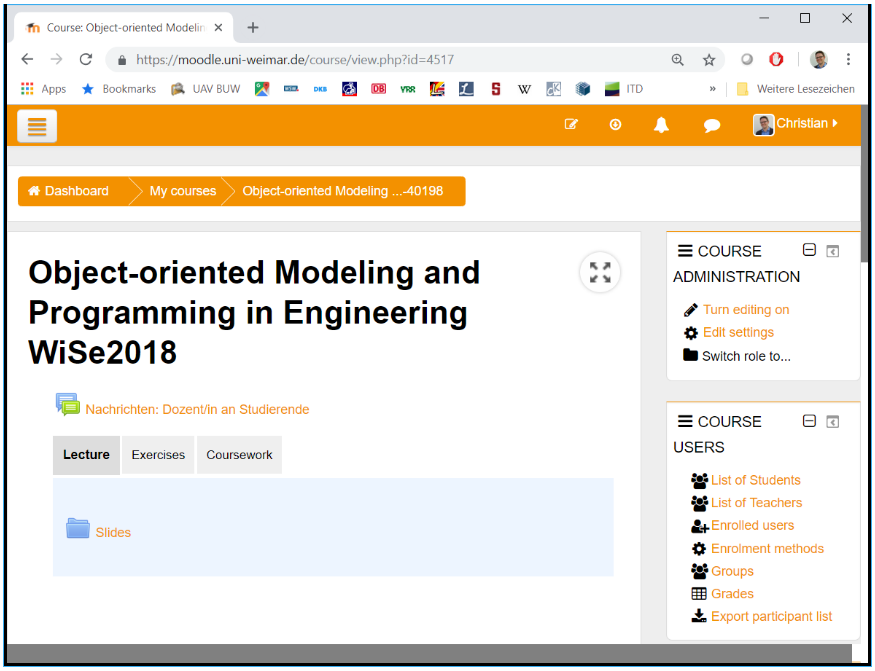


Lectures				**Tutorials**		
Nr	*Date*	*Content*		*Nr*	*Date*	*Content*
1	08.10.2018			1		
2	15.10.2018	Introduction		2	19.10.2018	Introduction Eclipse
3	22.10.2018	Objects		3	26.10.2018	Objects
4	29.10.2018	Classes-1		4	02.11.2018	free programming and questions
5	05.11.2018	Classes-2		5	09.11.2018	classes
6	12.11.2018	Control structures and Algorithms-1		6	16.11.2018	free programming and questions
7	19.11.2018	Control structures and Algorithms-2		7	23.11.2018	Algorithms 1; Assignm. 1
8	26.11.2018	Matrix algorithms-1		8	30.11.2018	free programming and questions
9	03.12.2018	Matrix algorithms-2		9	07.12.2018	Algorithms 2
10	10.12.2018	Data structures-1		10	14.12.2018	free programming and questions
11	17.12.2018	Data structures-2		11	21.12.2018	Datastructures; Assignm. 2, Assignm. 1 due
12	07.01.2019	Inheritance and polymorphism		12	11.01.2019	Inheritance
13	14.01.2019	Associations		13	18.01.2019	Associations
14	21.01.2019	Triangulation		14	25.01.2019	free programming and questions, Assignm. 2 due
15	28.01.2019	GUI		15	01.02.2019	GUI programming

Organisation

Bauhaus-Universität Weimar

- Moodle



Course: Object-oriented Modelin

https://moodle.uni-weimar.de/course/view.php?id=4517

Dashboard > My courses > Object-oriented Modeling ...-40198

Object-oriented Modeling and Programming in Engineering WiSe2018

Nachrichten: Dozent/in an Studierende

Slides

COURSE ADMINISTRATION

- Turn editing on
- Edit settings
- Switch role to...

COURSE USERS

- List of Students
- List of Teachers
- Enrolled users
- Enrolment methods
- Groups
- Grades
- Export participant list

Note

- *The lecture content and slides are based on the course*
  - *„Modern Programming Concepts in Engineering “*
  - *taught by Prof. Dr.-Ing. Matthias Baitsch, Bochum University of Applied Sciences*

Hochschule Bochum
Bochum University
of Applied Sciences

Faculty Civil Engineering
Chair of Intelligent Technical Design
Prof. Dr.-Ing. Christian Koch

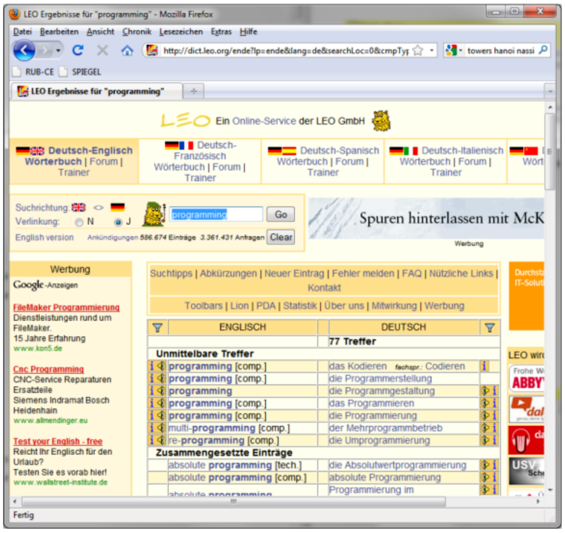
Bauhaus-Universität Weimar

Topic 6: Data structures

Introduction

Bauhaus-Universität Weimar

- A data structure is a particular way of storing and organizing data in a computer so that it can be used efficiently



LEO Ergebnisse für "programming" - Mozilla Firefox

Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe
http://dict.leo.org/ende?lp=ende&lang=de&searchLoc=0&cmpTyp
towers hanoi nassi
RUB-CE
SPIEGEL
LEO Ergebnisse für "programming"
LEO Ein Online-Service der LEO GmbH
Deutsch-Englisch Wörterbuch | Forum | Trainer
Deutsch-Französisch Wörterbuch | Forum | Trainer
Deutsch-Spanisch Wörterbuch | Forum | Trainer
Deutsch-Italienisch Wörterbuch | Forum | Trainer
Suchrichtung: <>
Verlinkung: ○ N ● J
English version
Ankündigungen 586.674 Einträge 3.361.431 Anfragen
Go
Clear
Spuren hinterlassen mit McK
Werbung
Suchtipps | Abkürzungen | Neuer Eintrag | Fehler melden | FAQ | Nützliche Links | Kontakt
Toolbars | Lion | PDA | Statistik | Über uns | Mitwirkung | Werbung
Werbung
Google-Anzeigen
Fertig

Suchbegriff: programming

FileMaker Programmierung
Dienstleistungen rund um FileMaker.
15 Jahre Erfahrung
www.kon5.de

Cnc Programming
CNC-Service Reparaturen Ersatzteile Siemens Indramat Bosch Heidenhain
www.allmendinger.eu

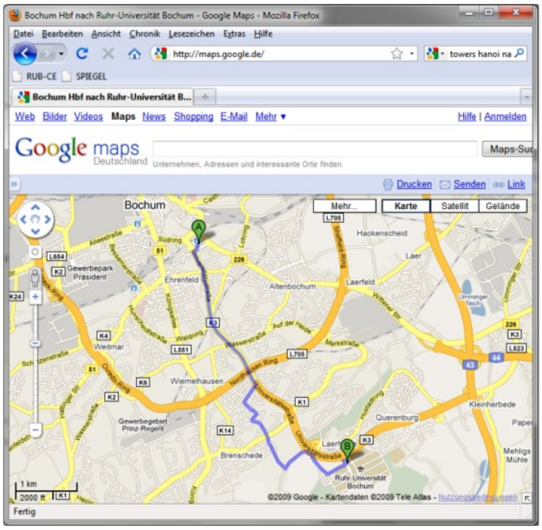
Test your English - free
Reicht Ihr Englisch für den Urlaub?
Testen Sie es vorab hier!
www.wallstreet-institute.de

ENGLISCH	DEUTSCH
	77 Treffer
Unmittelbare Treffer	
programming [comp.]	das Kodieren fachspr.: Codieren
programming [comp.]	die Programmerstellung
programming	die Programmgestaltung
programming [comp.]	das Programmieren
programming [comp.]	die Programmierung
multi-programming [comp.]	der Mehrprogrammbetrieb
re-programming [comp.]	die Umprogrammierung
Zusammengesetzte Einträge	
absolute programming [tech.]	die Absolutwertprogrammierung
absolute programming [comp.]	absolute Programmierung
absolute programming	Programmierung im

  – Dictionary: How to store millions of words for fast access?

Introduction

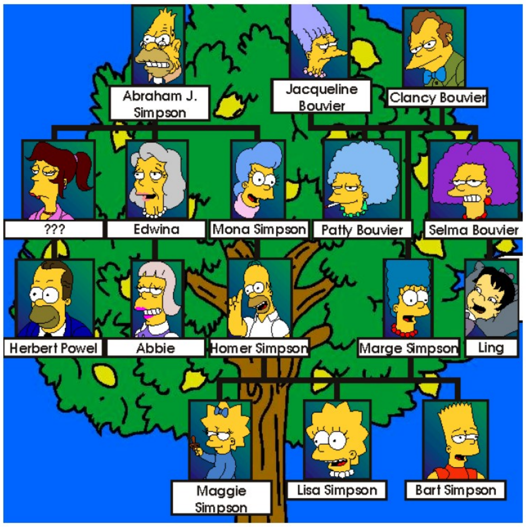
- A data structure is a particular way of storing and organizing data in a computer so that it can be used efficiently


Bochum Hbf nach Ruhr-Universität Bochum - Google Maps - Mozilla Firefox
Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe
http://maps.google.de/
towers hanoi na
RUB-CE SPIEGEL
Bochum Hbf nach Ruhr-Universität B...
Web Bilder Videos Maps News Shopping E-Mail Mehr ▾
Hilfe | Anmelden
Google maps Deutschland
Maps-Suc
Unternehmen, Adressen und interessante Orte finden.
Drucken Senden Link
Mehr... Karte Satellit Gelände
Bochum
A
B
Hordel
Ehrenfeld
Altenbochum
Laerfeld
Weitmar
Wiemelhausen
Querenburg
Brenschede
Kleinherbede
Ruhr-Universität Bochum
1 km
2000 ft
©2009 Google - Kartendaten ©2009 Tele Atlas - Nutzungsbedingungen
Fertig

  - Routing: How to store towns and streets to find the shortest connection?

Introduction

- A data structure is a particular way of storing and organizing data in a computer so that it can be used efficiently

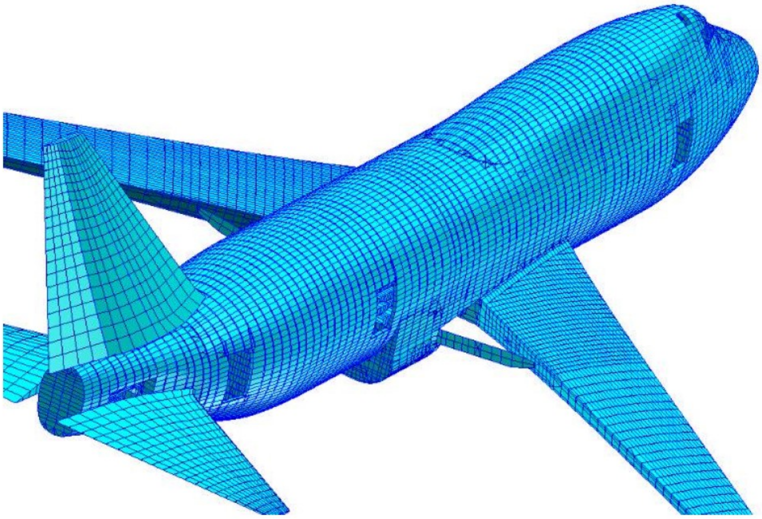


  - Family tree: Who are the descendants of a certain person?

Introduction

Bauhaus-Universität Weimar

- A data structure is a particular way of storing and organizing data in a computer so that it can be used efficiently



  - Finite element program: How to access and modify thousands of nodes and elements.

Introduction

- A data structure is a particular way of storing and organizing data in a computer so that it can be used efficiently
  - Different kinds of data structures are suited to different kinds of applications
  - Specific data structures are essential ingredients of many efficient algorithms, and make possible the management of huge amounts of data

Introduction

- A data structure is a particular way of storing and organizing data in a computer so that it can be used efficiently
  - Different kinds of data structures are suited to different kinds of applications
  - Specific data structures are essential ingredients of many efficient algorithms, and make possible the management of huge amounts of data
  - Two different general types of data structures are distinguished
    - Array structures
    - Linked structures
  - Array data structures are based on computing the addresses of data items with arithmetic operations
  - Linked data structures are based on storing addresses of data items within the structure itself

Introduction

- A data structure is a particular way of storing and organizing data in a computer so that it can be used efficiently
  - In this topic we are dealing with two types data structures:
    - lists and
    - trees
  - Practically, data structures are implemented as classes
  - An instance of a data structure is an object of the data structure type
  - A data structure object stores objects of the same type

Lists

- A list (or sequence) is an ordered collection of elements with a linear structure

$$\mathbf{L} := < e_0, e_1, \ldots, e_{n-1} >$$

- Lists have an absolute order
  - each element has a dedicated index
- as well as a relative order
  - Beside the first and the last element, each element has a predecessor and successor element

Lists

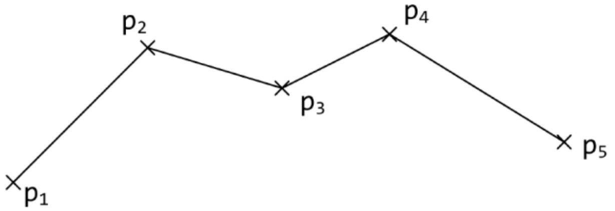
- The elements of a list are objects
  - List of String objects

$$\mathbf{L} := < "Modern", "Programming", "Concepts" >$$

  - List of Integer objects (Fibonacci numbers)

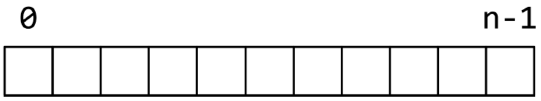
$$\mathbf{L} := < 0, 1, 1, 2, 3, 5, 8, 13 >$$

  - List of Point objects (polyline) $\mathbf{L} := < p_1, p_2, p_3, p_4, p_5 >$

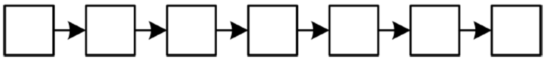


Lists

- Two implementations for lists exist
  - Array list (absolute order)



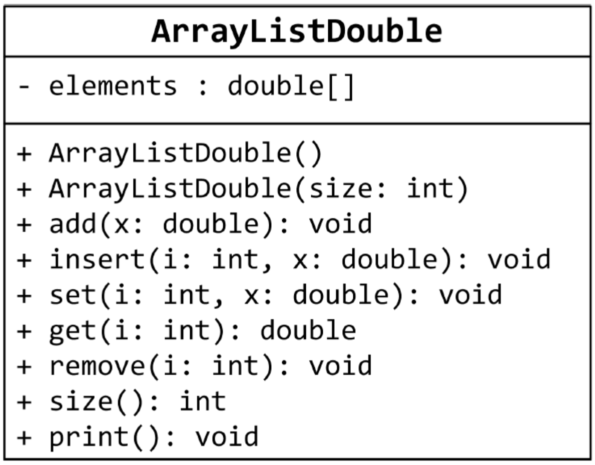
  - Linked list (relative order)



- Typical operations are
  - add an element to the end
  - insert an element at a specified position
  - set an element at a specified position
  - return the element at a specified position
  - remove the element at a specified position
  - return the size
  - print the elements

Array lists…

Array lists



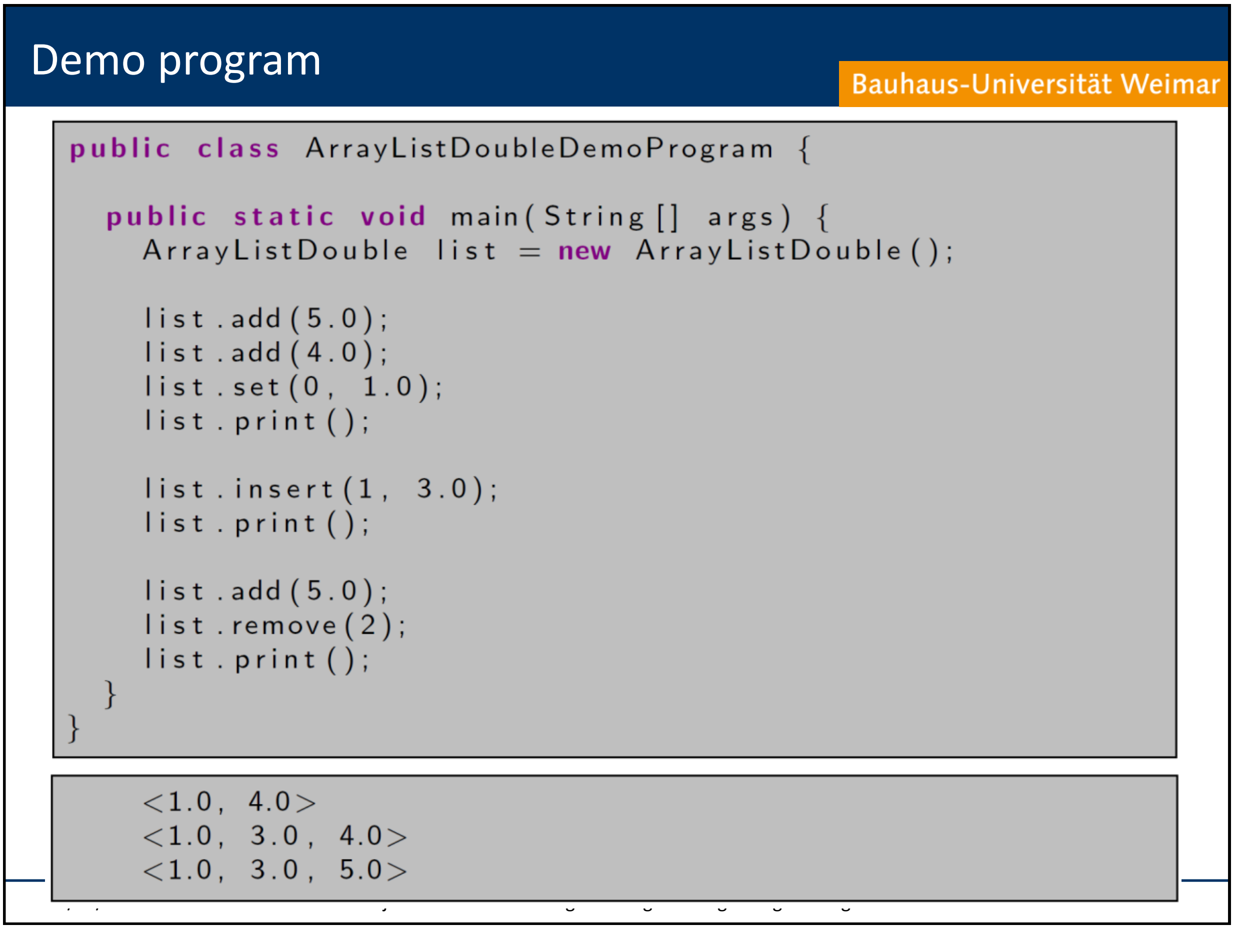
```
ArrayListDouble
- elements : double[]
+ ArrayListDouble()
+ ArrayListDouble(size: int)
+ add(x: double): void
+ insert(i: int, x: double): void
+ set(i: int, x: double): void
+ get(i: int): double
+ remove(i: int): void
+ size(): int
+ print(): void
```

- Array lists store data elements in an array
- A new array has to be created when elements are added or removed and existing elements have to be copied into the new array

Array lists

Bauhaus-Universität Weimar

```java
public class ArrayListDouble {

  private double[] elements = new double[0];

  public void insert(int i, double x) {
    if (i < 0 || i > this.size()) {
      throw new IndexOutOfBoundsException("Index: " + i);
    }
    double[] tmp = new double[this.size() + 1];

    for (int j = 0; j < i; j++) {
      tmp[j] = this.elements[j];
    }
    tmp[i] = x;
    for (int j = i; j < this.size(); j++) {
      tmp[j + 1] = this.elements[j];
    }
    this.elements = tmp;
  }

  public void add(double x) {
    this.insert(this.size(), x);
  }

  // other methods come here ...
}
```



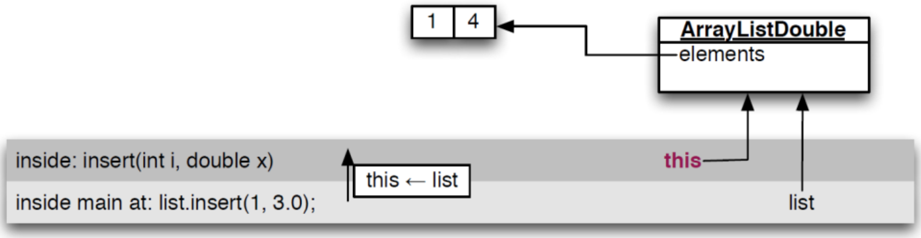
Demo program

Bauhaus-Universität Weimar

```
public class ArrayListDoubleDemoProgram {

  public static void main(String[] args) {
    ArrayListDouble list = new ArrayListDouble();

    list.add(5.0);
    list.add(4.0);
    list.set(0, 1.0);
    list.print();

    list.insert(1, 3.0);
    list.print();

    list.add(5.0);
    list.remove(2);
    list.print();
  }
}
```

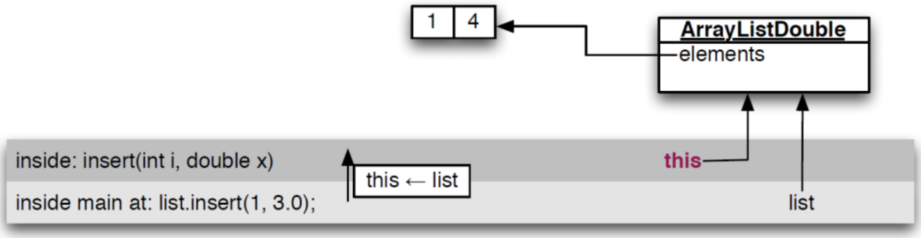
```
    <1.0, 4.0>
    <1.0, 3.0, 4.0>
    <1.0, 3.0, 5.0>
```

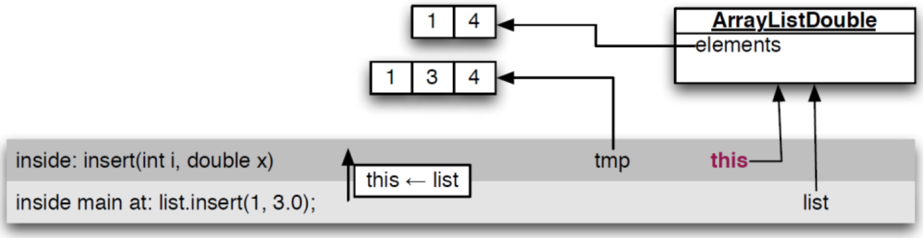
Insert method


1
4
ArrayListDouble
elements
inside: insert(int i, double x)
this ← list
this
inside main at: list.insert(1, 3.0);
list

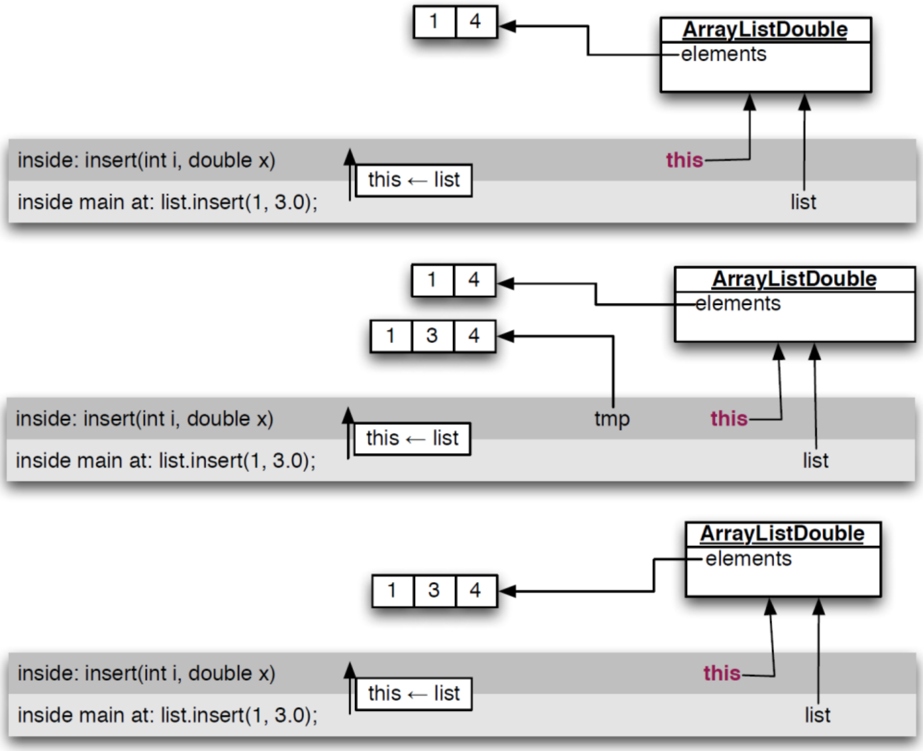
Insert method

Bauhaus-Universität Weimar


1
4
ArrayListDouble
elements
inside: insert(int i, double x)
this ← list
this
inside main at: list.insert(1, 3.0);
list


1
4
ArrayListDouble
elements
1
3
4
inside: insert(int i, double x)
this ← list
tmp
this
inside main at: list.insert(1, 3.0);
list

Insert method


1
4
ArrayListDouble
elements
inside: insert(int i, double x)
this ← list
this
inside main at: list.insert(1, 3.0);
list
1
4
1
3
4
ArrayListDouble
elements
inside: insert(int i, double x)
this ← list
tmp
this
inside main at: list.insert(1, 3.0);
list
ArrayListDouble
elements
1
3
4
inside: insert(int i, double x)
this ← list
this
inside main at: list.insert(1, 3.0);
list

Linked lists…