COMP4038/G54SOD (Spring 2019)

Lecture 02

Introduction to AnyLogic and Java

Peer-Olaf Siebers

pos@cs.nott.ac.uk

The University of Nottingham
UNITED KINGDOM · CHINA · MALAYSIA

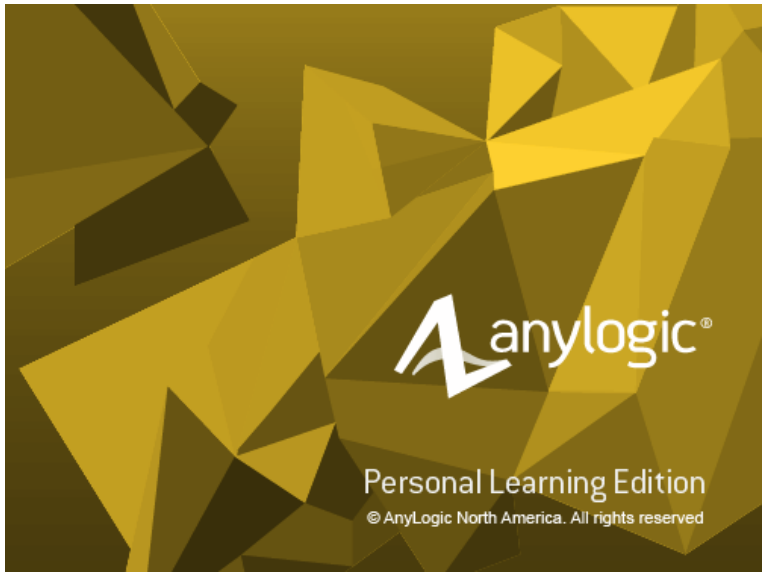
Content

- AnyLogic
- Objects and Java in 20 Minutes
- Java Basics for AnyLogic
- The Laptop Challenge

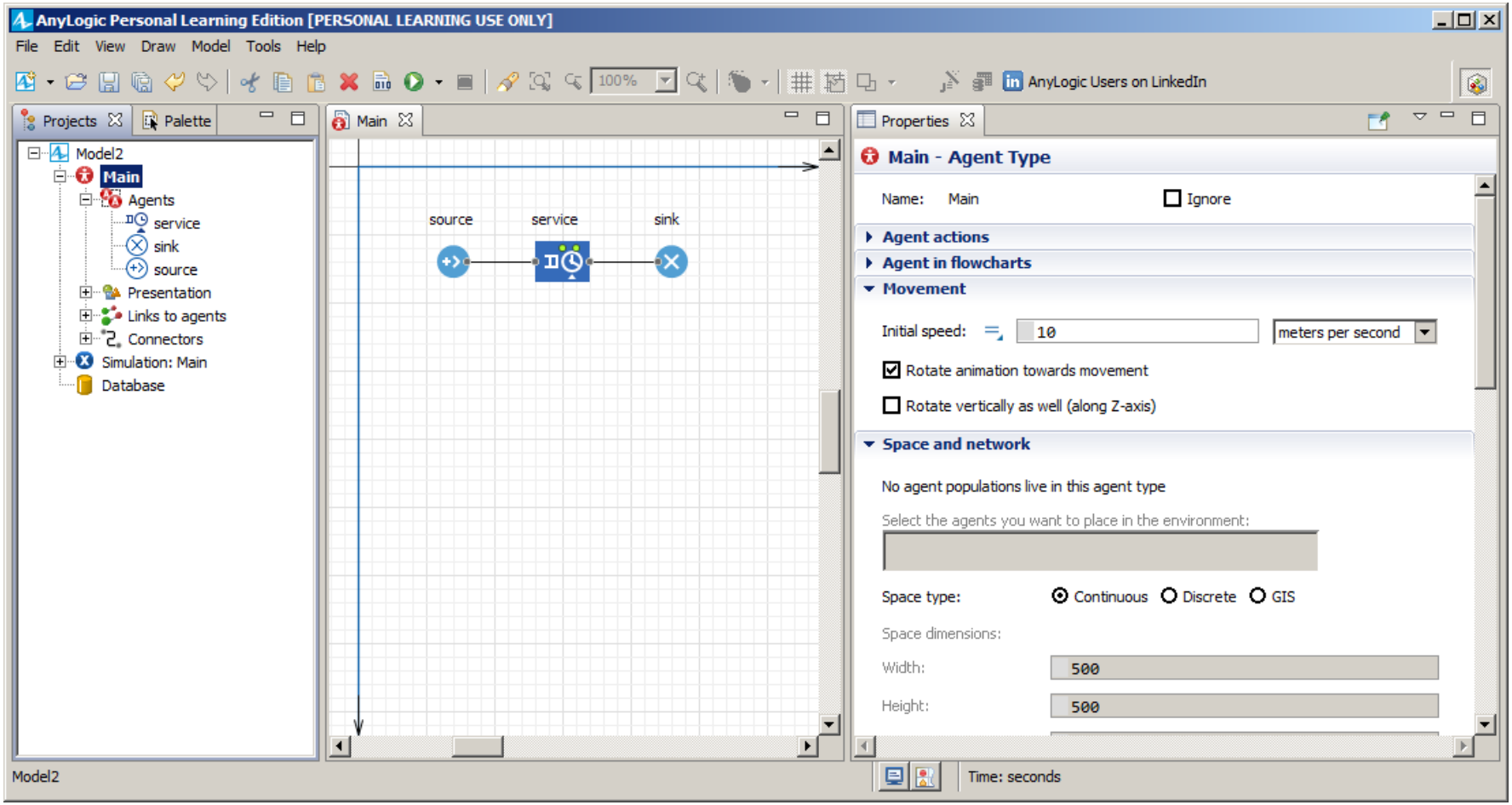
AnyLogic

AnyLogic

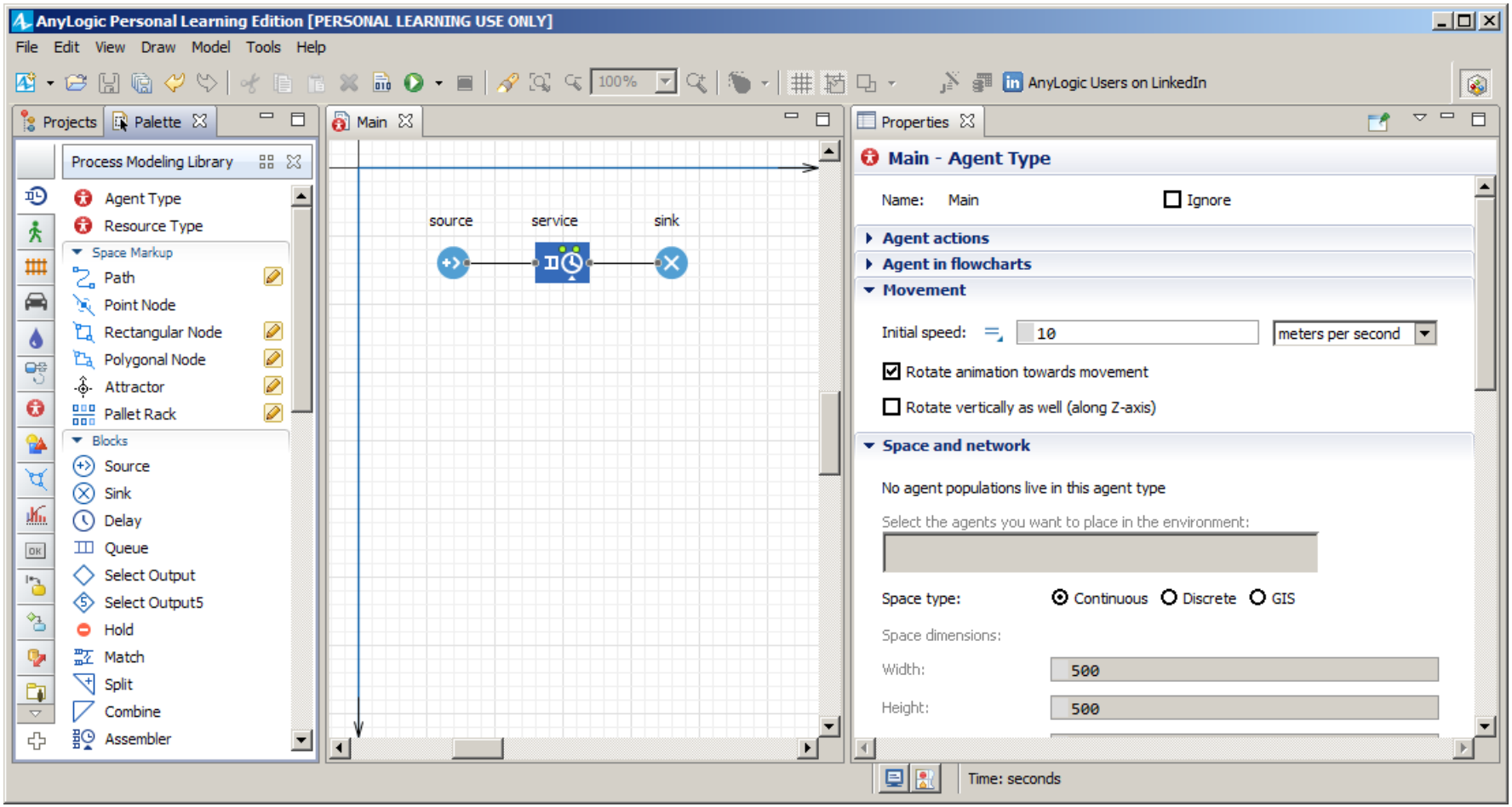
- We use AnyLogic 8.3.2 PLE (should be compatible with 8.3.3)
  - In AnyLogic you are not writing the full code of Java classes from the beginning to the end; instead you are entering pieces of code and expressions in numerous small edit boxes in the properties of various model elements



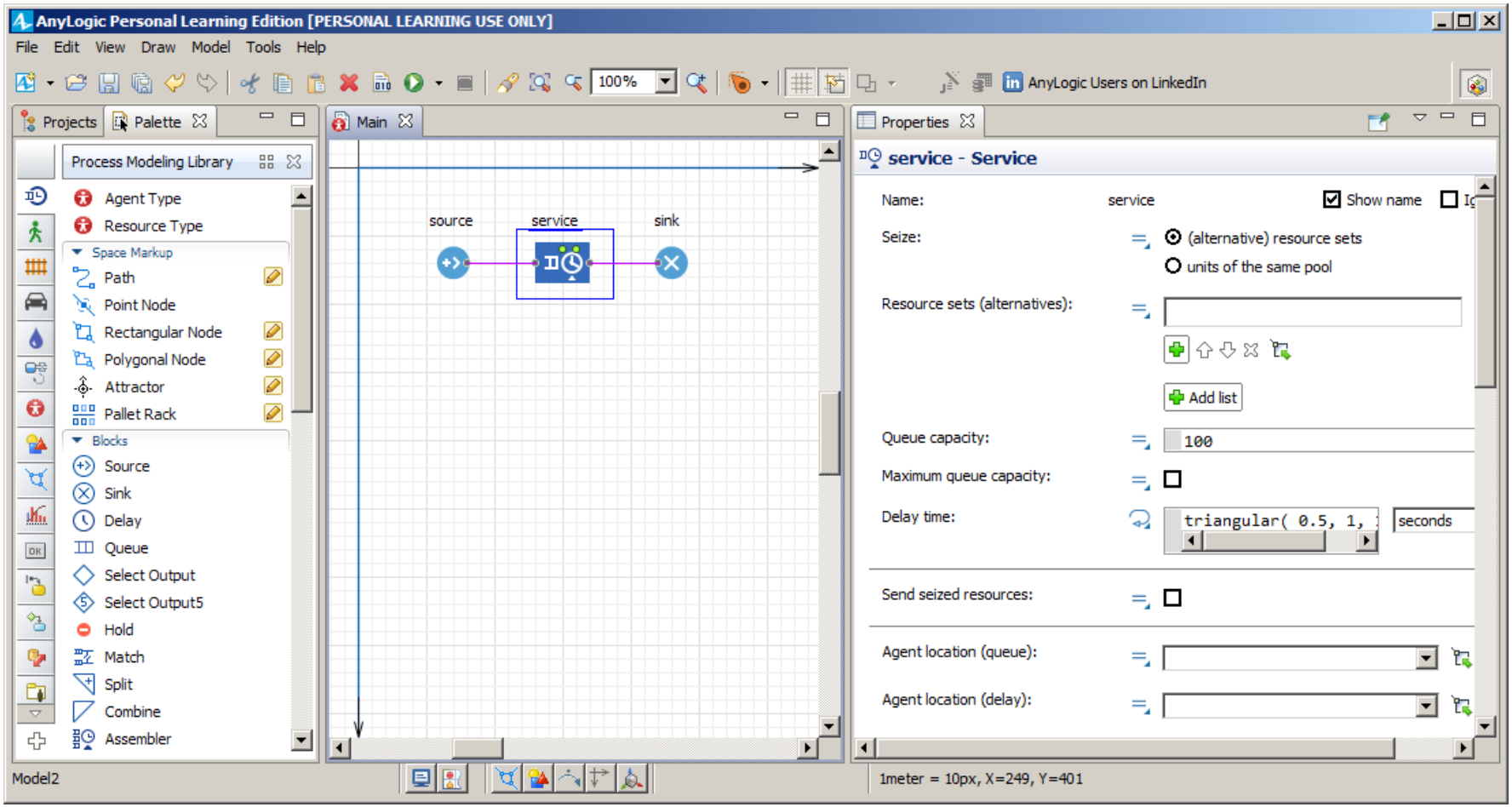
AnyLogic IDE



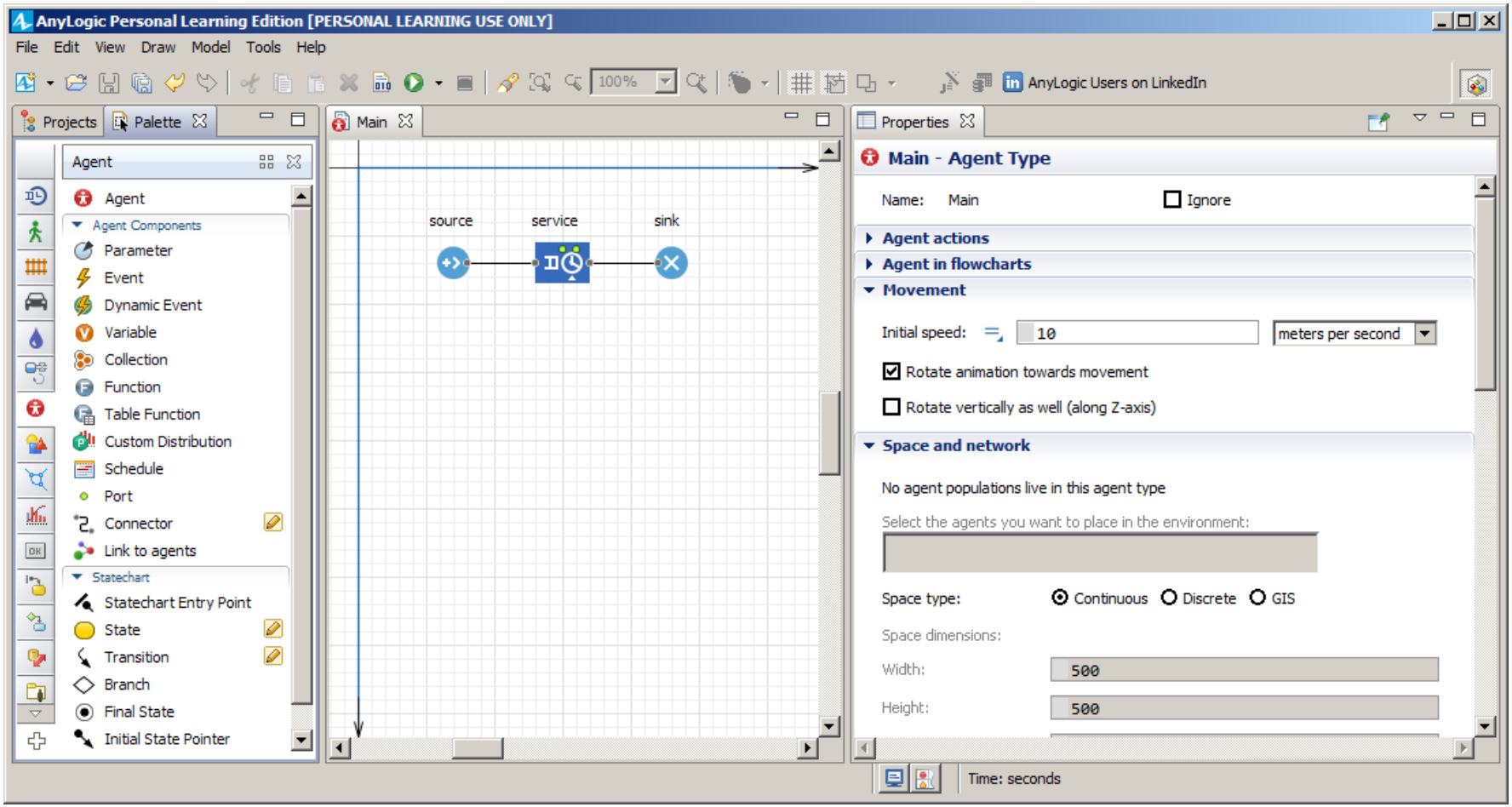
AnyLogic IDE



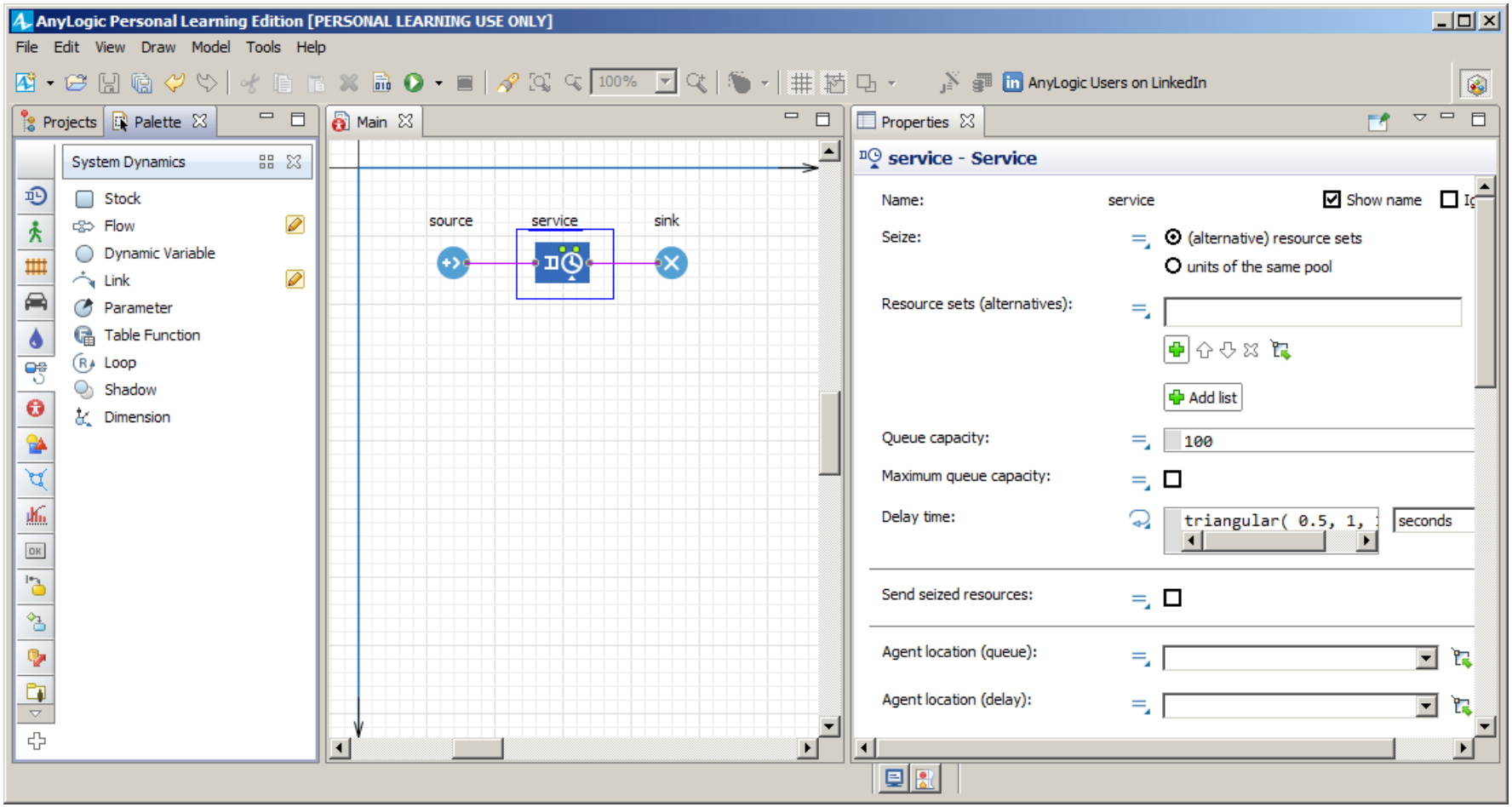
AnyLogic IDE



AnyLogic IDE



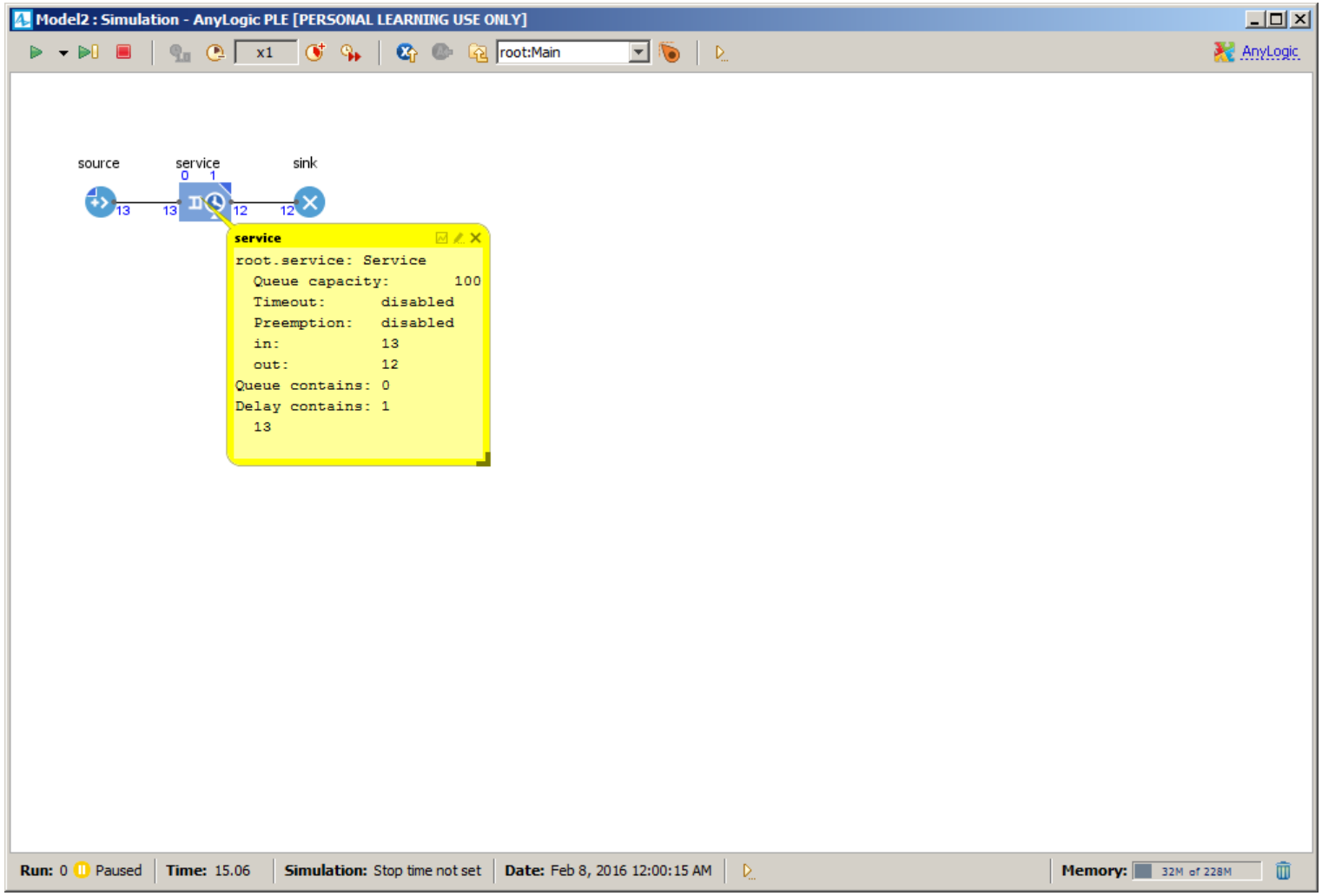
AnyLogic IDE



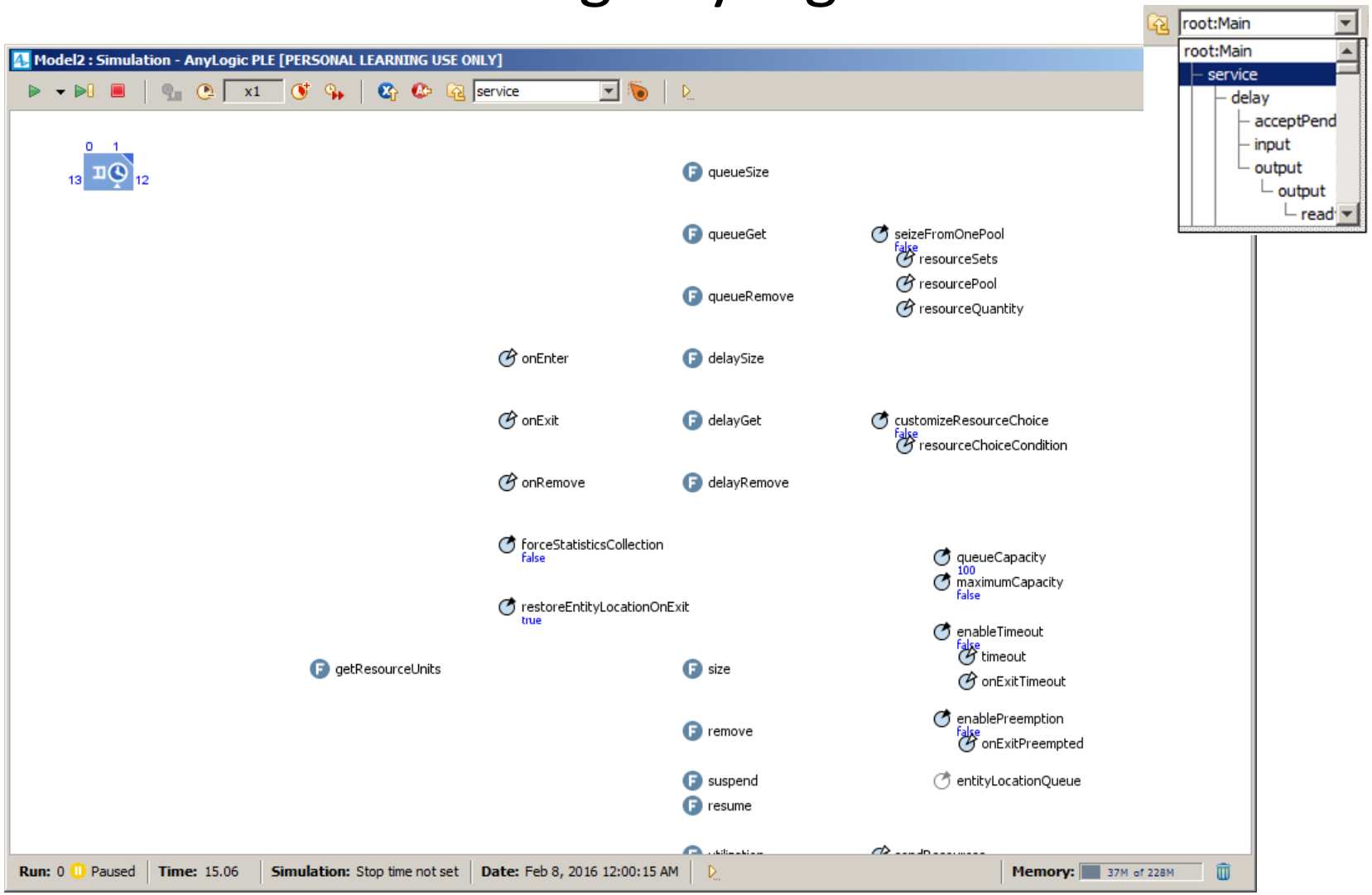
Things to Remember

- Important things
  - F1: Help
  - Ctrl-Space: Code completion support
  - Ctrl-Enter: Perform refactoring (replace name occurrences)
  - Make sure you select the correct model when pressing "Run"
  - Make sure you set up model time units correctly in the "Model"
  - Use the "magic lightbulb" ...
- Since AnyLogic 7 …
  - Everything is called "Agent" (entities, resources, agents, …)
  - PLE version limits number of entities per simulation run to 50,000

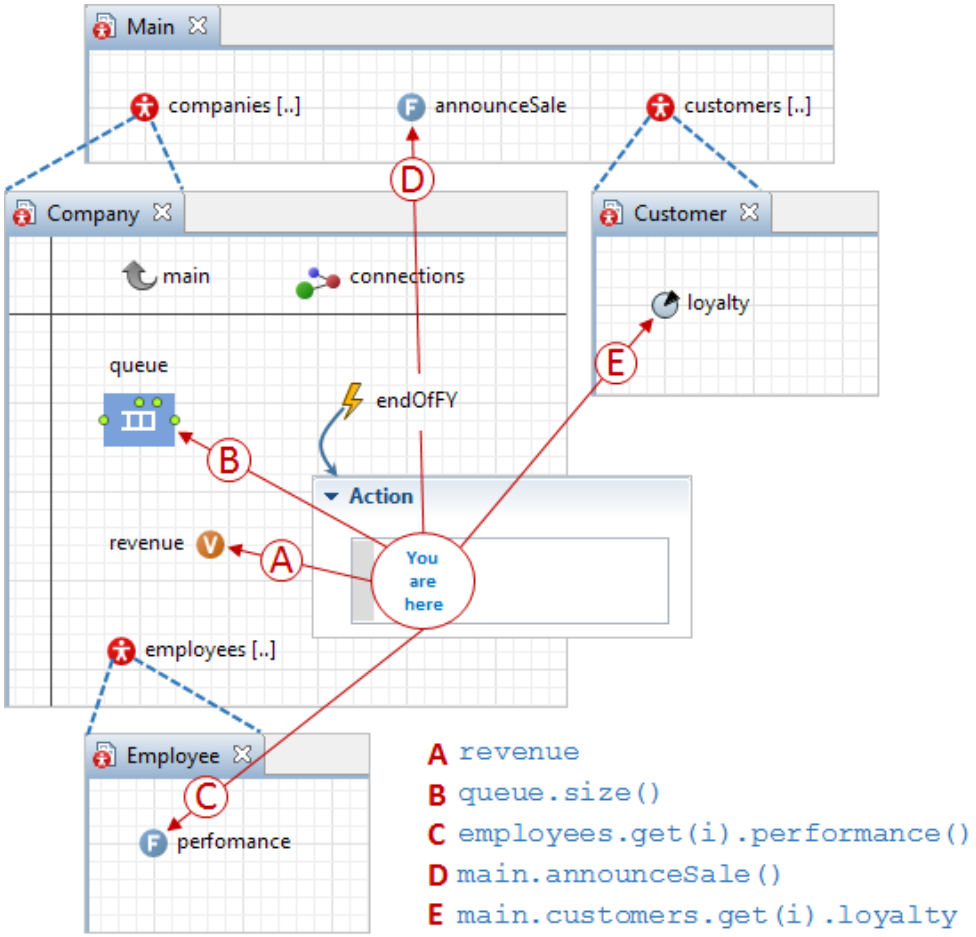
Running AnyLogic



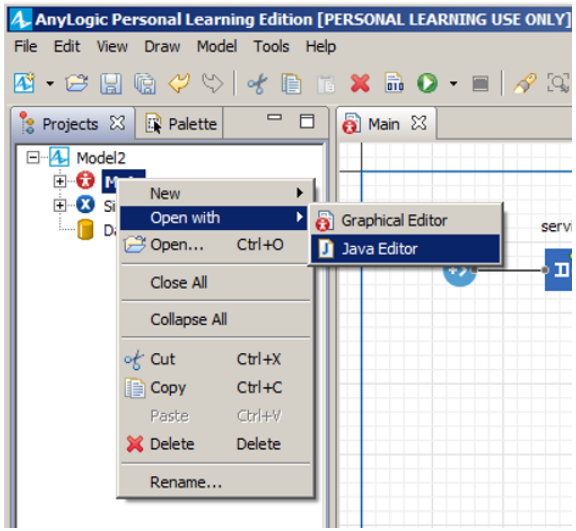
Running AnyLogic



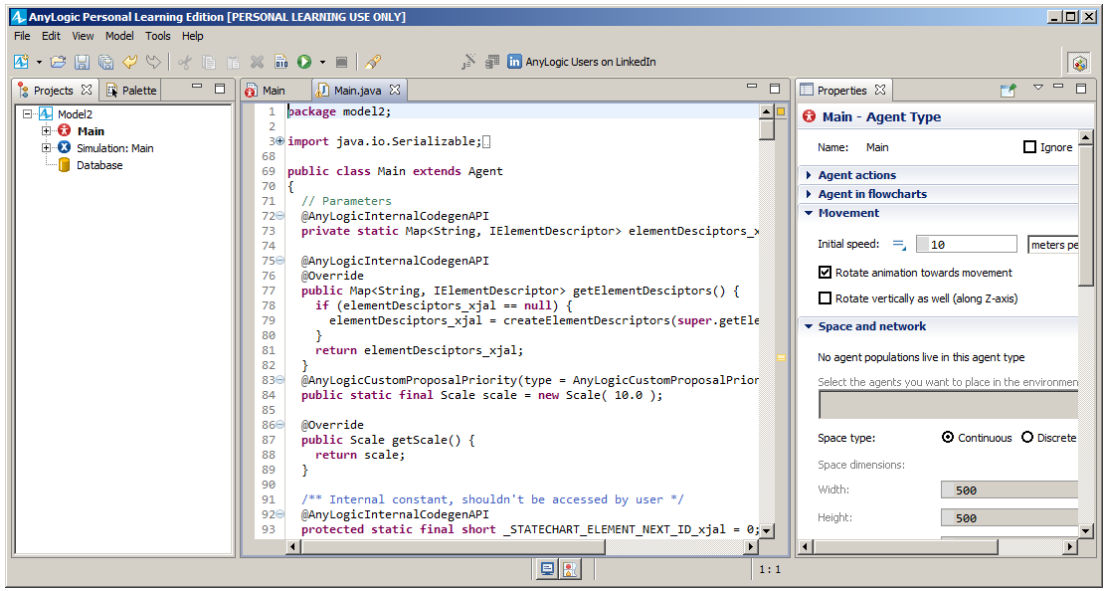
Where am I and how do I get to…?



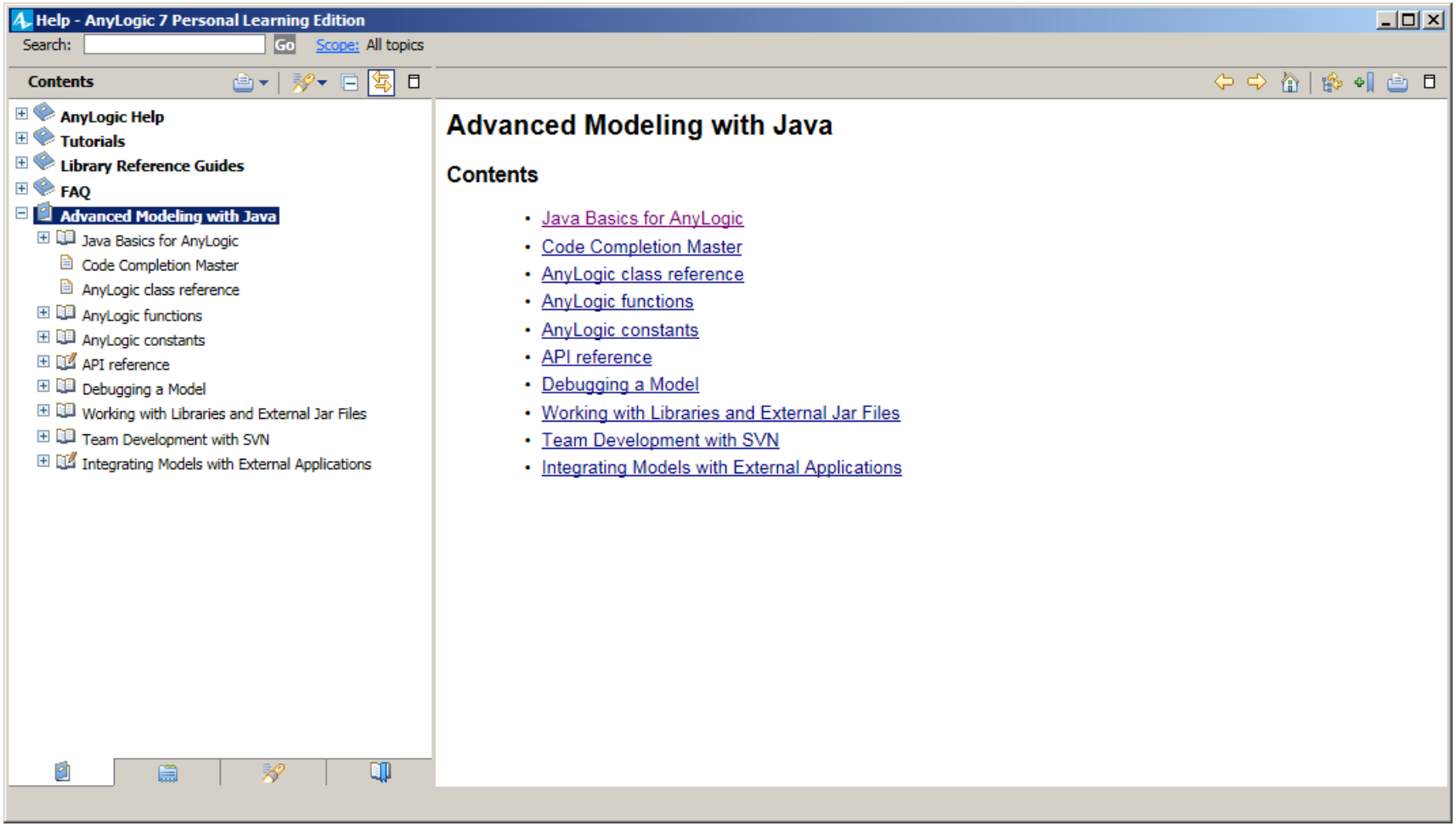
AnyLogic IDE



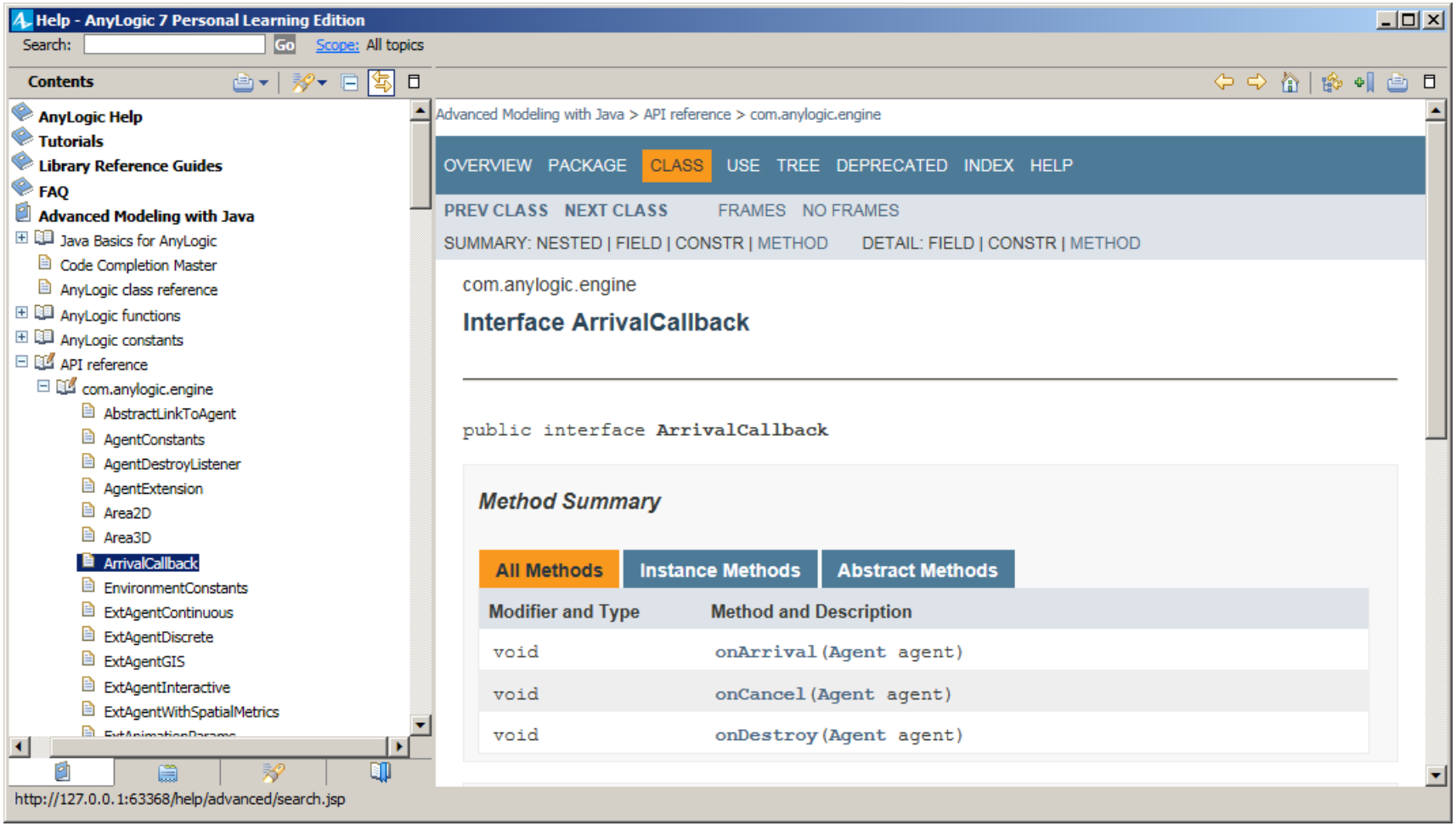
- Or press "Ctrl+J" to go to the point in the Java code that is associated with the current code snippet highlighted in the Properties window



AnyLogic Help



AnyLogic Help



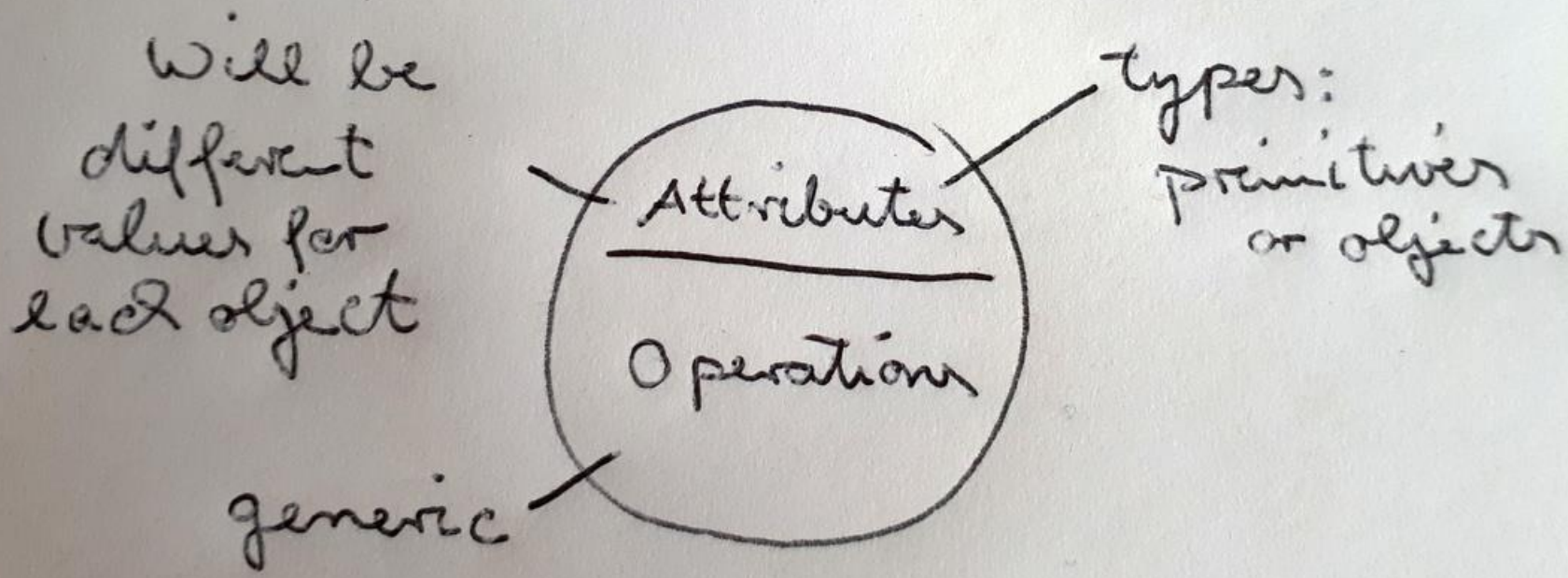
Objects and Java in 20 Minutes

The University of Nottingham
UNITED KINGDOM · CHINA · MALAYSIA

(1)

Classes

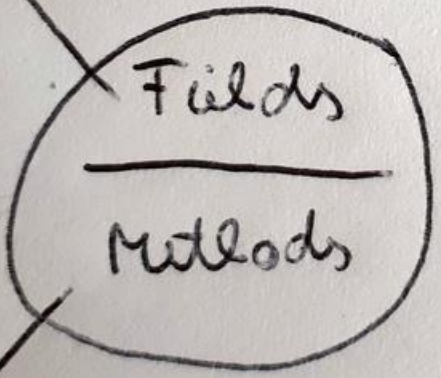
↳ Blueprints for objects



(2)

Objects in Java (instantiations of classes)

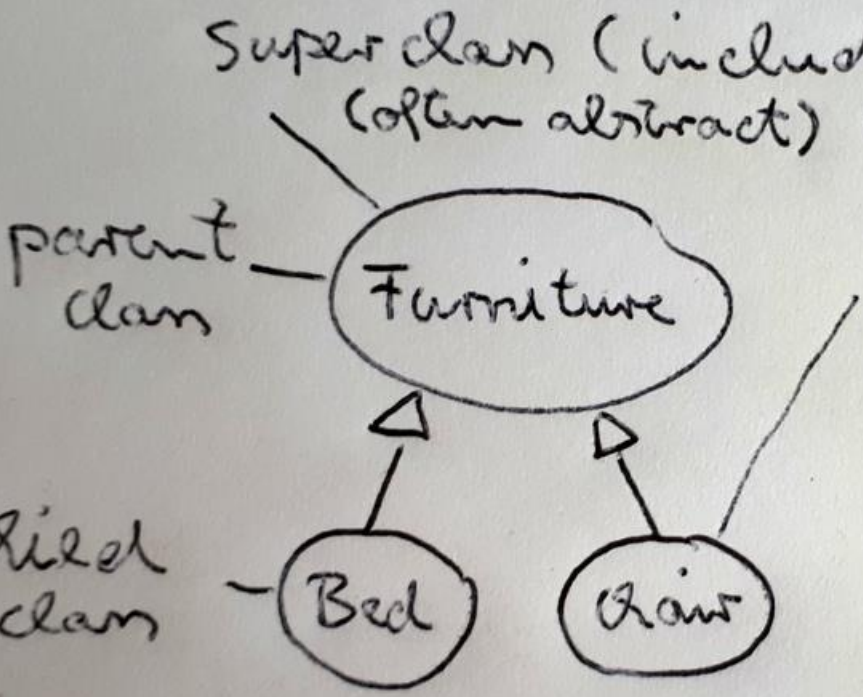
usually private - use getters and setters to access (encapsulation)



often used to separate concerns

usually public, unless helper methods
↳ accessible from other objects

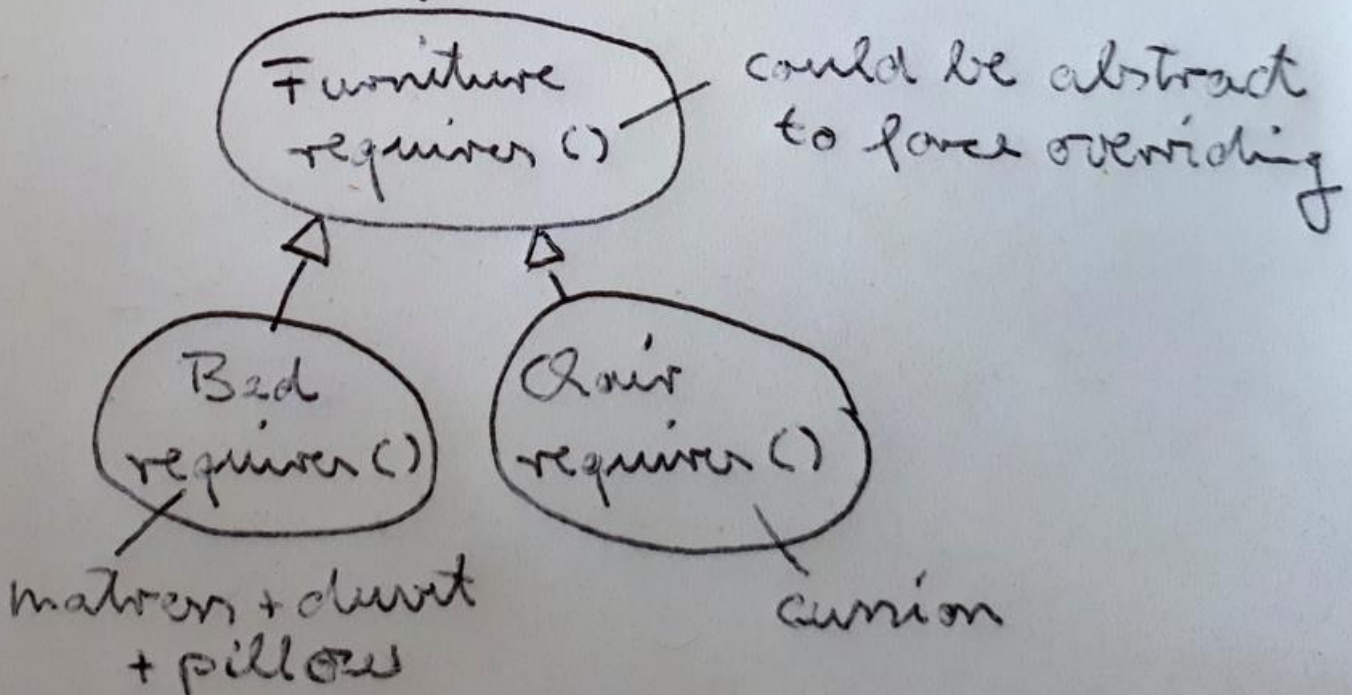
(3) Inheritance (of fields and methods)



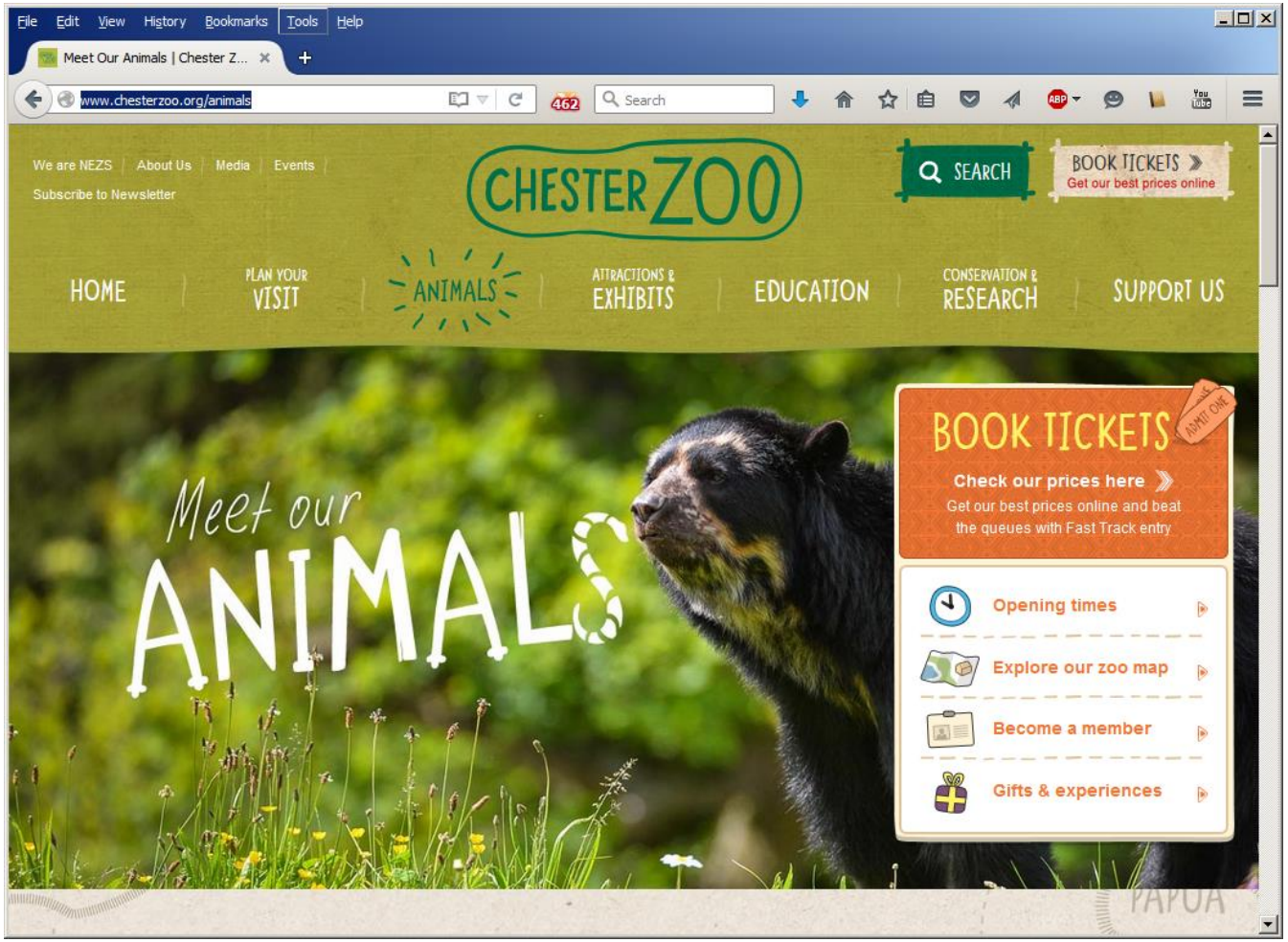
=) In Java only single inheritance

(4) Compile time Polymorphism (static binding)
↳ method overloading
↳ multiple methods with same name but different parameter list
Public boolean order()
public boolean order(int num, Color col)

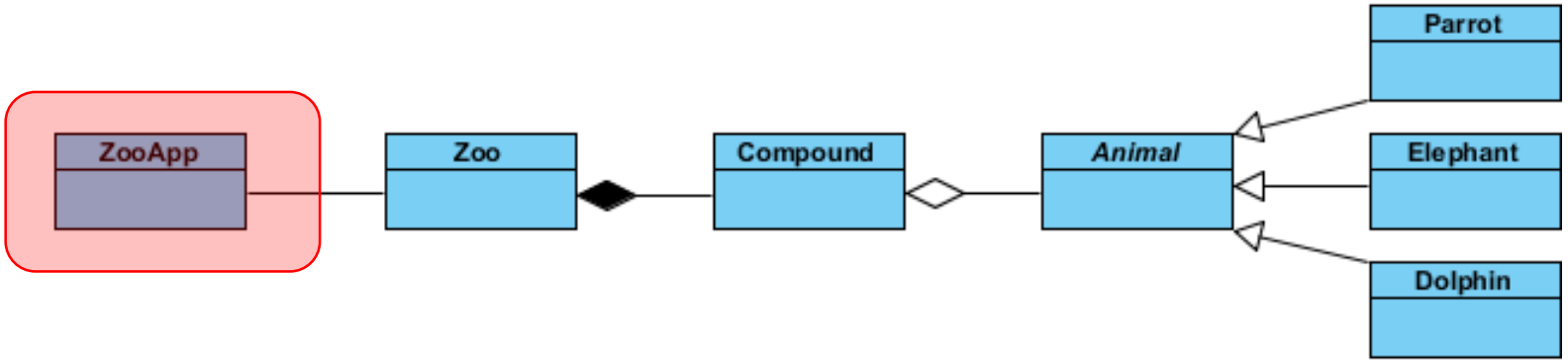
Run time Polymorphism
↳ method overriding



Zoo Management



Zoo Management



Java - Java - Zoo project/src/ZooApp.java - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer JUnit

- Zoo project
  - src
    - (default package)
      - Animal.java
      - Compound.java
      - Dolphin.java
      - Elephant.java
      - Parrot.java
      - Zoo.java
      - ZooApp.java
  - JRE System Library [JavaSE-1.8]

*ZooApp.java *Zoo.java *Animal.java *Compound.java *Dolphin.java

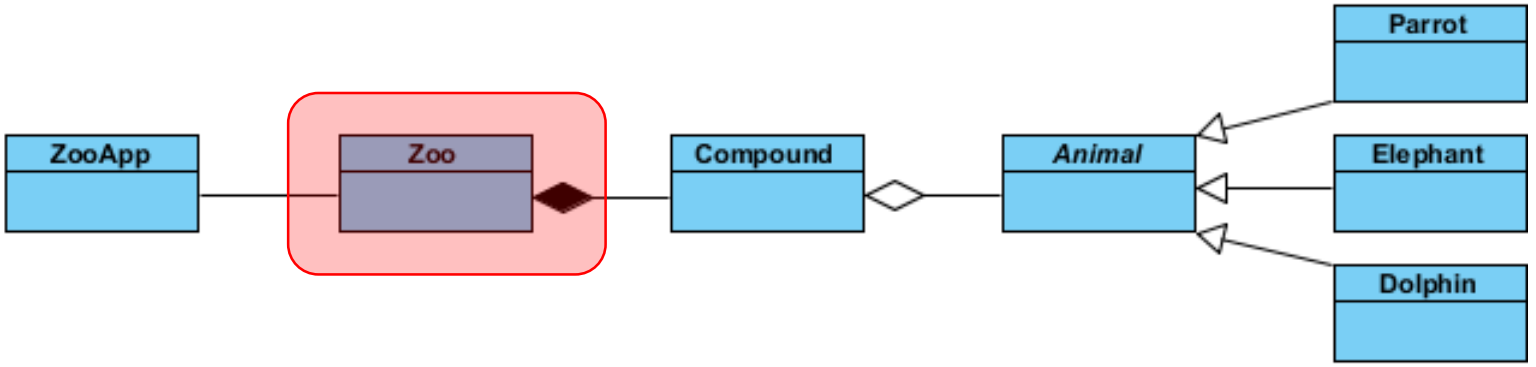
```
1  import java.util.ArrayList;
2
3  public class ZooApp {
4
5      public static void main(String[] args) {
6          ArrayList<Zoo> zoos=new ArrayList<Zoo>();
7          zoos.add(new Zoo("London", 10));
8          zoos.add(new Zoo("Paris", 20));
9          for(int i=0; i<zoos.size(); i++){
10             zoos.get(i).printInfo();
11         }
12         zoos.get(0).getCompounds().get(0).addAnimal(new Dolphin(2));
13         zoos.get(0).getCompounds().get(0).removeAnimal(1);
14         Dolphin d=(Dolphin)(zoos.get(0).getCompounds().get(0).getAnimals().get(0));
15         d.printType();
16         d.swim();
17     }
18 }
19
```

Problems Javadoc Declaration Console Git Staging History

<terminated> ZooApp (1) [Java Application] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe (3 Feb 2019, 16:56:47)

```
Location:London; Compounds:10
Location:Paris; Compounds:20
animal does not exist
Dolphine
... swimming
```

Zoo Management



Java - Java - Zoo project/src/Zoo.java - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer JUnit

- Zoo project
  - src
    - (default package)
      - Animal.java
      - Compound.java
      - Dolphin.java
      - Elephant.java
      - Parrot.java
      - Zoo.java
      - ZooApp.java
  - JRE System Library [JavaSE-1.8]

*ZooApp.java *Zoo.java *Animal.java *Compound.java *Dolphin.java

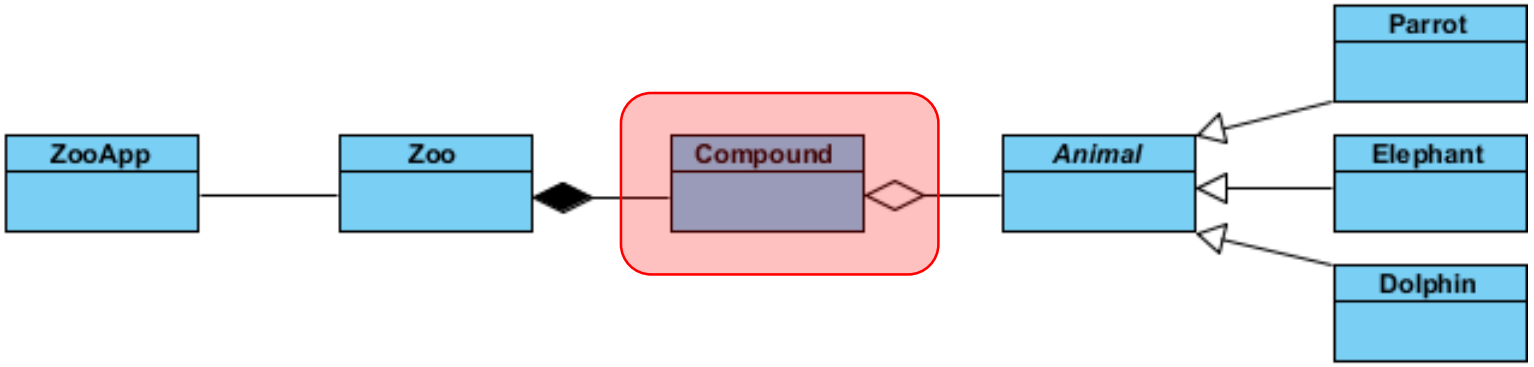
```
1  import java.util.ArrayList;
2
3  public class Zoo {
4
5      private String location;
6      private int numberOfCompounds;
7      private ArrayList<Compound> compounds;
8
9      public Zoo(String location, int numberOfCompounds) {
10         this.location = location;
11         this.numberOfCompounds = numberOfCompounds;
12         setCompounds(new ArrayList<Compound>());
13         for (int i=0; i<numberOfCompounds; i++) {
14             this.getCompounds().add(new Compound());
15         }
16     }
17     public String getLocation() {
18         return location;
19     }
20     public void setLocation(String location) {
21         this.location = location;
22     }
23     public void printInfo() {
24         System.out.println("Location:"+location+"; Compounds:"+numberOfCompounds);
25     }
26     public ArrayList<Compound> getCompounds() {
27         return compounds;
28     }
29     private void setCompounds(ArrayList<Compound> compounds) {
30         this.compounds = compounds;
31     }
32 }
```

Problems Javadoc Declaration Console Git Staging History

<terminated> ZooApp (1) [Java Application] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe (3 Feb 2019, 16:56:47)

```
Location:London; Compounds:10
Location:Paris; Compounds:20
animal does not exist
Dolphine
... swimming
```

Zoo Management



Java - Java - Zoo project/src/Compound.java - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer JUnit

- Zoo project
  - src
    - (default package)
      - Animal.java
      - Compound.java
      - Dolphin.java
      - Elephant.java
      - Parrot.java
      - Zoo.java
      - ZooApp.java
  - JRE System Library [JavaSE-1.8]

*ZooApp.java *Zoo.java *Animal.java *Compound.java *Dolphin.java

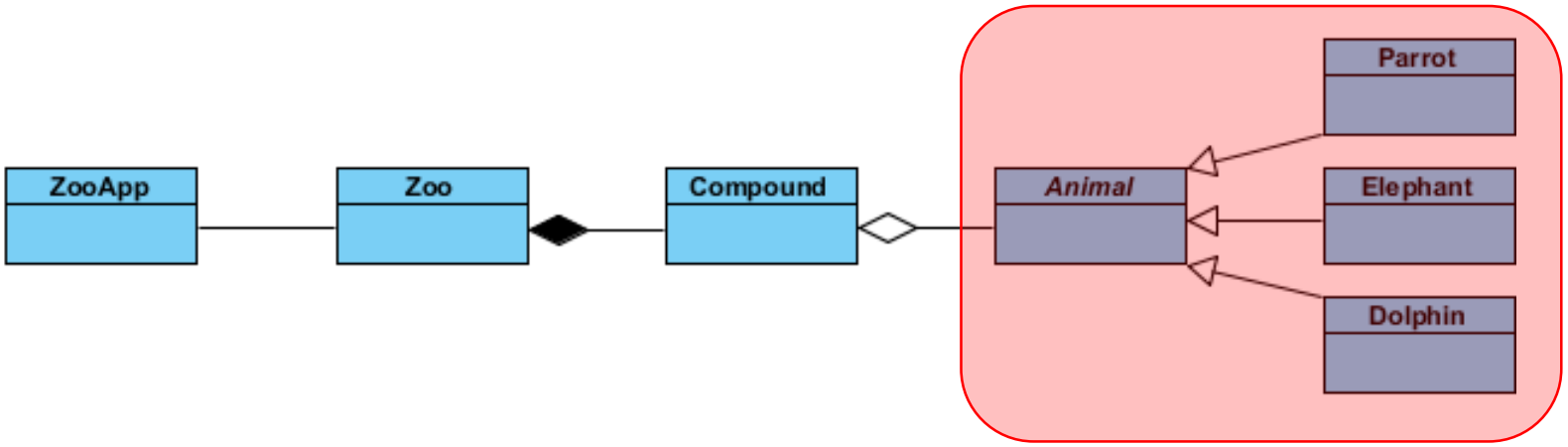
```
1  import java.util.ArrayList;
2
3  public class Compound {
4
5      private ArrayList<Animal> animals;
6
7      public Compound(){
8          setAnimals(new ArrayList<Animal>());
9      }
10     public void addAnimal(Animal animal){
11         getAnimals().add(animal);
12     }
13     public void removeAnimal(int index){
14         try{
15             getAnimals().remove(index);
16         }catch(IndexOutOfBoundsException e){
17             System.out.println("animal does not exist");
18         }
19     }
20     public ArrayList<Animal> getAnimals() {
21         return animals;
22     }
23     private void setAnimals(ArrayList<Animal> animals) {
24         this.animals = animals;
25     }
26 }
27
```

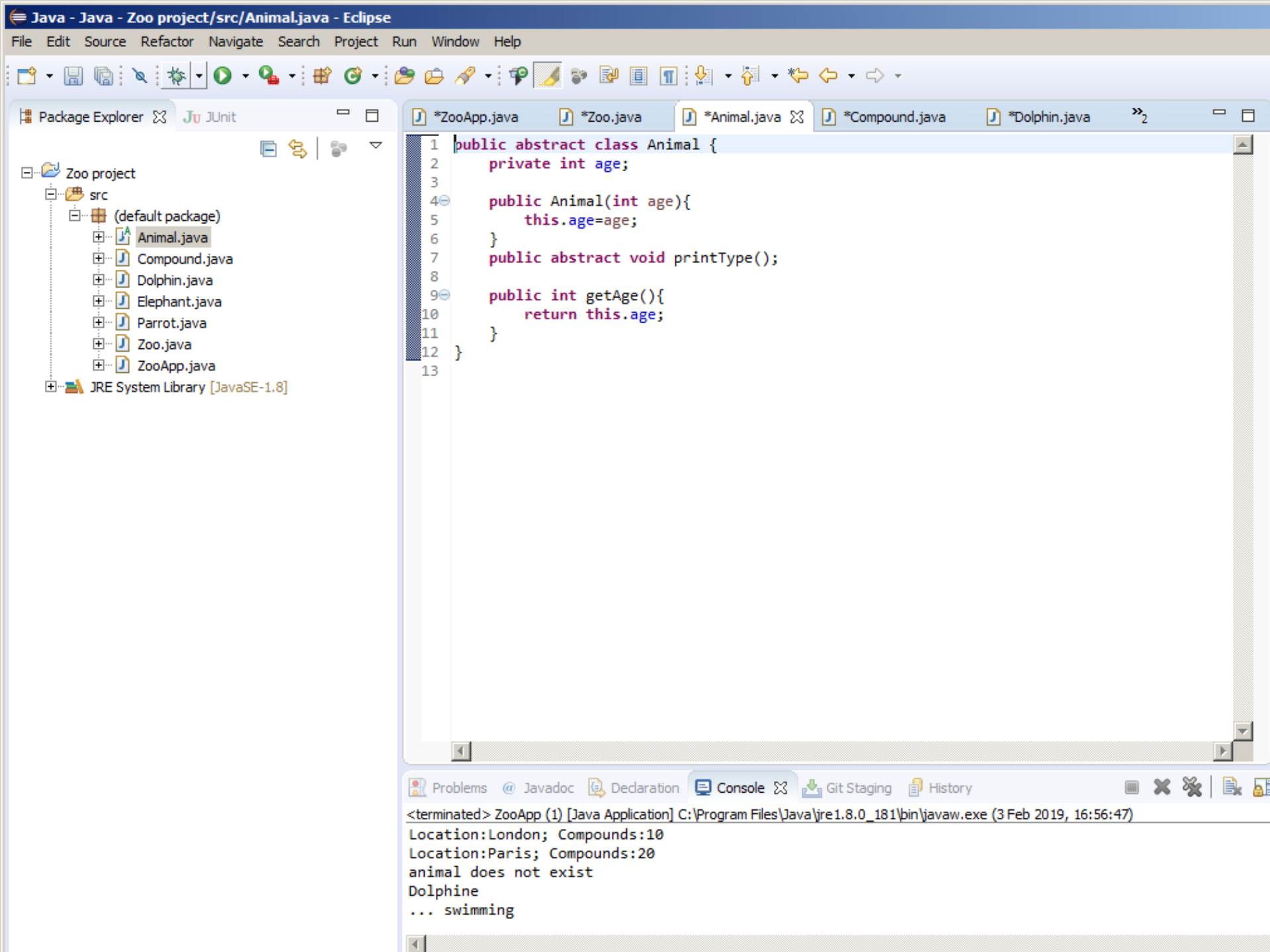
Problems Javadoc Declaration Console Git Staging History

<terminated> ZooApp (1) [Java Application] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe (3 Feb 2019, 16:56:47)

```
Location:London; Compounds:10
Location:Paris; Compounds:20
animal does not exist
Dolphine
... swimming
```

Zoo Management




Java - Java - Zoo project/src/Animal.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
JUnit
Zoo project
src
(default package)
Animal.java
Compound.java
Dolphin.java
Elephant.java
Parrot.java
Zoo.java
ZooApp.java
JRE System Library [JavaSE-1.8]
*ZooApp.java
*Zoo.java
*Animal.java
*Compound.java
*Dolphin.java
1 public abstract class Animal {
2 private int age;
3
4 public Animal(int age){
5 this.age=age;
6 }
7 public abstract void printType();
8
9 public int getAge(){
10 return this.age;
11 }
12 }
13
Problems
Javadoc
Declaration
Console
Git Staging
History
<terminated> ZooApp (1) [Java Application] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe (3 Feb 2019, 16:56:47)
Location:London; Compounds:10
Location:Paris; Compounds:20
animal does not exist
Dolphine
... swimming

Java - Java - Zoo project/src/Dolphin.java - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer JUnit

- Zoo project
  - src
    - (default package)
      - Animal.java
      - Compound.java
      - Dolphin.java
      - Elephant.java
      - Parrot.java
      - Zoo.java
      - ZooApp.java
  - JRE System Library [JavaSE-1.8]

*ZooApp.java *Zoo.java *Animal.java *Compound.java *Dolphin.java

```
1  public class Dolphin extends Animal {
2
3      public Dolphin(int age){
4          super(age);
5      }
6      @Override
7      public void printType(){
8          System.out.println("Dolphine");
9      }
10     public void swim(){
11         System.out.println("... swimming");
12     }
13 }
14
```

Problems Javadoc Declaration Console Git Staging History

<terminated> ZooApp (1) [Java Application] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe (3 Feb 2019, 16:56:47)

```
Location:London; Compounds:10
Location:Paris; Compounds:20
animal does not exist
Dolphine
... swimming
```

Java - Java - Zoo project/src/Elephant.java - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer JUnit

- Zoo project
  - src
    - (default package)
      - Animal.java
      - Compound.java
      - Dolphin.java
      - Elephant.java
      - Parrot.java
      - Zoo.java
      - ZooApp.java
  - JRE System Library [JavaSE-1.8]

*Animal.java *Compound.java *Dolphin.java *Elephant.java *Parrot.java

```
1  public class Elephant extends Animal {
2
3      public Elephant(int age){
4          super(age);
5      }
6      @Override
7      public void printType(){
8          System.out.println("Elephant");
9      }
10     public void run(){
11         System.out.println("... running");
12     }
13 }
14
```

Problems Javadoc Declaration Console Git Staging History

<terminated> ZooApp (1) [Java Application] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe (3 Feb 2019, 16:56:47)

```
Location:London; Compounds:10
Location:Paris; Compounds:20
animal does not exist
Dolphine
... swimming
```

Java - Java - Zoo project/src/Parrot.java - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer JUnit

- Zoo project
  - src
    - (default package)
      - Animal.java
      - Compound.java
      - Dolphin.java
      - Elephant.java
      - Parrot.java
      - Zoo.java
      - ZooApp.java
  - JRE System Library [JavaSE-1.8]

*Animal.java *Compound.java *Dolphin.java *Elephant.java *Parrot.java

```
1  public class Parrot extends Animal {
2
3      public Parrot(int age) {
4          super(age);
5      }
6      @Override
7      public void printType() {
8          System.out.println("Parrot");
9      }
10     public void fly(){
11         System.out.println("... flying");
12     }
13 }
14
```

Problems Javadoc Declaration Console Git Staging History

<terminated> ZooApp (1) [Java Application] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe (3 Feb 2019, 16:56:47)

```
Location:London; Compounds:10
Location:Paris; Compounds:20
animal does not exist
Dolphine
... swimming
```

Java Basics for AnyLogic

The University of Nottingham
UNITED KINGDOM · CHINA · MALAYSIA

Java Basics for AnyLogic

- General remarks
  - You do not have to learn full OO programming
    - You need to understand Java data types, expression, and statement syntax
  - Please note:
    - Java is case-sensitive: MyVar is different to myVar!
    - Spaces are not allowed in names: "My Var" is an illegal name!
    - Each statement has to be finished with ";": MyVar=150;
    - Each function has to have parenthesis: time(), add(a)
    - Mind integer division: 3/2=1, not 1.5
    - Boolean values are only true and false, you cannot use 1 and 0
    - Dot "." brings you "inside" the object: agent.event.restart()
    - Array elements have indexes from 0 to n-1

Java Basics for AnyLogic

- Primitive Types
  - double: Represents real numbers: 1.43, 3.6E18, -14.0
  - int: Represents integer numbers: 12, 16384, -5000
  - boolean: Represents Boolean (true/false) values

- Compound Types –Classes
  - String: Represents textual strings, e.g. "MSFT", "Hi there!", etc.
  - ArrayList; LinkedList: Represents collections of objects
  - HyperArray: Represents multi-dimensional array
  - …many others. See AnyLogic and Java Class References

Java Basics for AnyLogic

- Arithmetic operations
  - Notation: +; –; *; /; % (remainder)
  - In integer divisions, the fraction part is lost, e.g. 3/2=1, and 2/3=0
  - Multiplication operators have priority over addition operators
  - The "+" operator allows operands of type String

- Comparison operations
  - Notation: >; >=; <; <=; ==; !=

- Boolean operations
  - Notation: && (AND); || (OR); ! (NOT)

Java Basics for AnyLogic

- Conditional operator
  - Notation: condition ? value-if-true : value-if-false
- Assignments and shortcuts
  - Notation: =; +=; -=; *=; /=; %=; ++; -- (a+=b is the same as a=a+b)
- Please note:
  - Within most of operators, left-to-right precedence holds
  - Parentheses may be used to alter the precedence of operations

Java Basics for AnyLogic

- Method call
  - To call a method, type its name followed by parenthesis; if necessary, put parameters separated by commas within the parenthesis
    - Examples:
      - x=time(); moveTo(getX(),getY()+100); traceln("Population is increasing");

- Accessing object fields and methods
  - To access a field or method of a model element (statechart, timer, animation), use the model element name followed by dot "." followed by the field/method name
    - Examples:
      - statechart.fireEvent("go"); sum=sum+agents.get(i).x;

Java Basics for AnyLogic

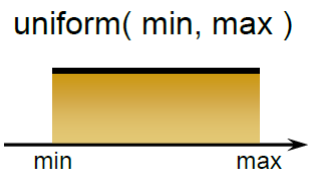
- Replicated objects are stored in a collection
  - Items are indexed from 0 to n-1
  - Getting the current size of the collection:
    - people.size()
  - Obtaining i-th item of the collection:
    - people.get(i)
  - Adding a new object to the collection:
    - add_people();
  - Removing an object from the collection:
    - remove_people(person);

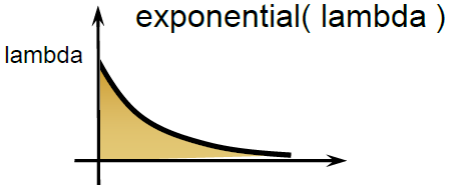
Java Basics for AnyLogic

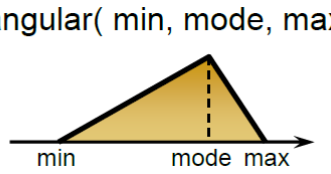
- Built-in Functions
  - System functions
    - time(); getOwner(); pause(); isStateActive(…); traceln(...); etc.
  - Mathematical functions
    - Basic: sqrt; sin; cos; tan; exp; log; round; zidz; xidz; etc.
    - Array: add; sub; mul; sum; avg; min; max; get; etc.
  - Special functions
    - Random numbers: uniform; exponential; bernoulli; beta; etc.
    - Time related: delay; etc.
  - And more…
    - See AnyLogic Class Reference

Java Basics for AnyLogic

- Probability Distributions
  - **Uniform:** Used to represent a random variable with constant likelihood of being in any small interval between min and max. Its density does not depend on the value of x.
  - **Exponential:** Used to represent the time between random occurrences. The unique property is history independence, i.e. it has the same set of probabilities when shifted in time.
  - **Triangular:** Used when no or little data is available to represent e.g. a process duration.







Java Basics for AnyLogic

- Common contextual variables that are used by code snippets
  - In statistics:
    - "item" indicates current agent
  - In "On Message Received" handler for agent:
    - "msg" indicates received message
  - In Dynamic properties of an Agent's replicated line property:
    - "index" indicates current person's index
  - In "Parameters" properties of Agent populations (used to set properties of agents within population):
    - "index" indicates the index of the current agent in the population

For more useful advice see Nathaniel Osgood's "AnyLogic and Java" presentation (url)

The Laptop Challenge

OO DES Example

The University of Nottingham
UNITED KINGDOM · CHINA · MALAYSIA

The Laptop Challenge

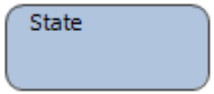
- Considering different power states

The Laptop Challenge

- Build a conceptual model of a laptop using a state chart
  - When the laptop is on and the user is working (i.e. is pressing the keyboard keys and moving the mouse, the laptop stays on)
  - After 5 minutes of user inactivity the laptop turns off the screen to save power
  - If the user remains inactive for another 10 minutes, the laptop switches to sleep mode to further minimise the battery usage
  - To wake the laptop up you need to press the power button
  - In any of the three states "On", "Screen Off" and "Sleep" the laptop consumes the battery power
  - When the battery level falls down to 10% the laptop is forced to shut down regardless of the state

The Laptop Challenge

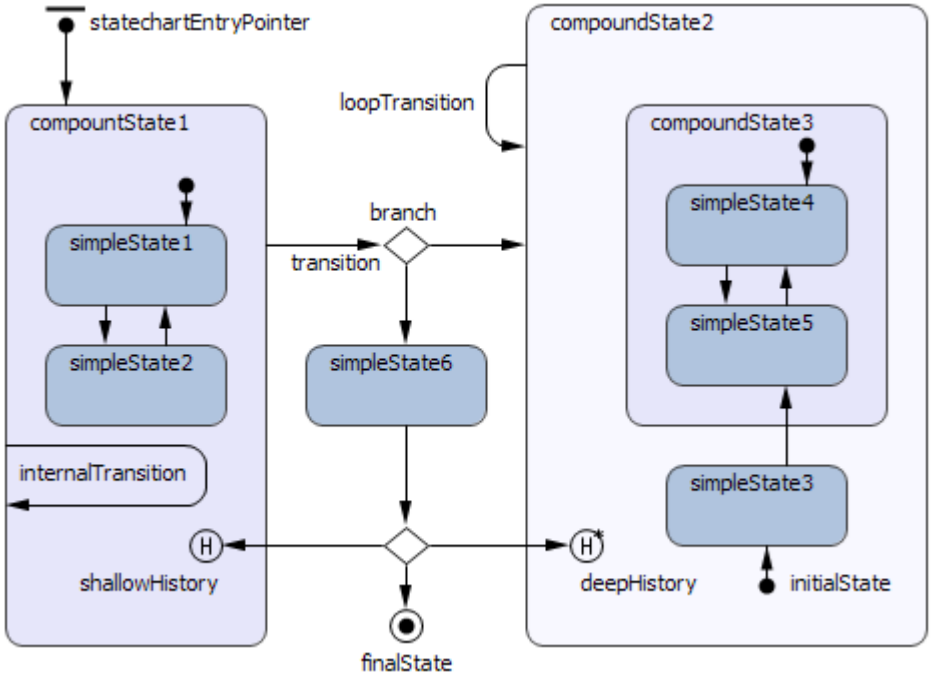
- Typical elements of a state chart diagram
  - States
    - Represents a location of control with a particular set of reactions to conditions and/or events



    - Examples
      - Cup can be in state **full** or **empty**
      - Person can be in state **idle** or **busy**
  - Transitions
    - Movement between states, triggered by a specific event

The Laptop Challenge

- Typical elements of a state chart diagram



Questions