Java Programming II

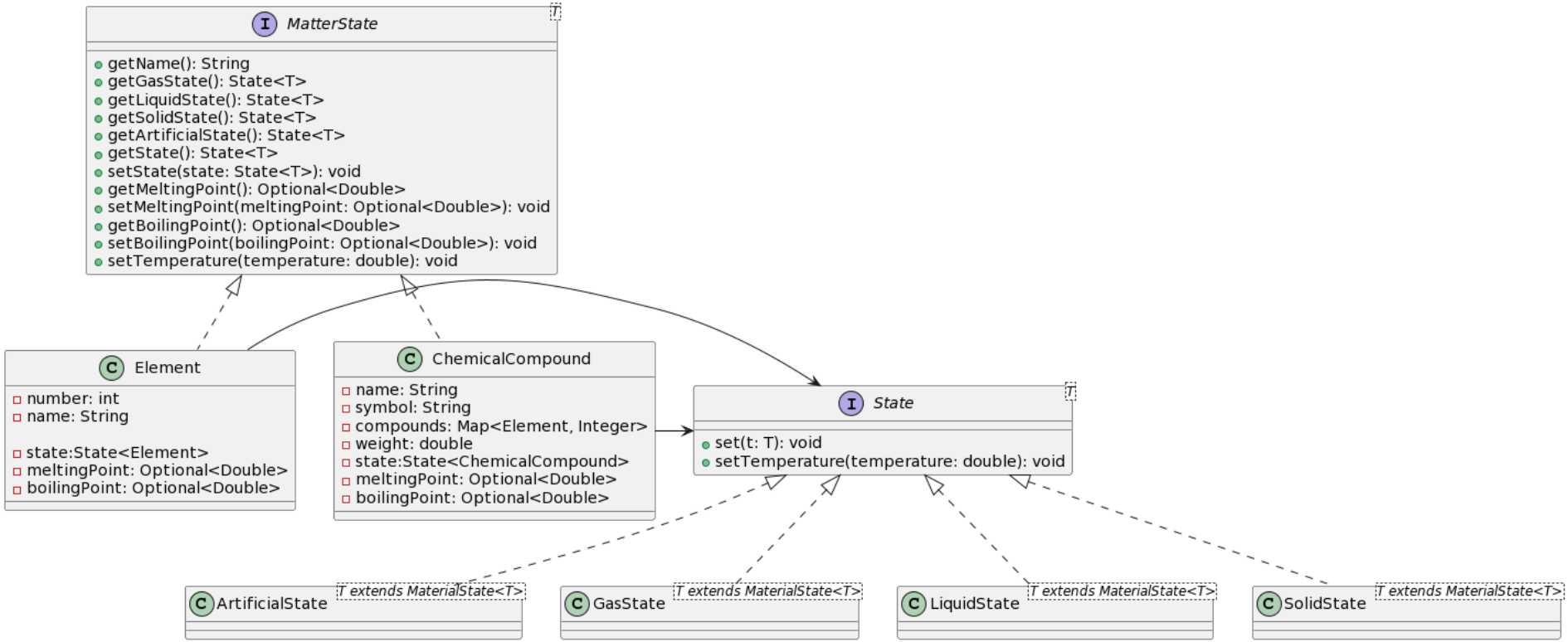
Lab9

514770-1
Fall 2023
11/21/2023
Kyoung Shin Park
Computer Engineering
Dankook University

Lab9

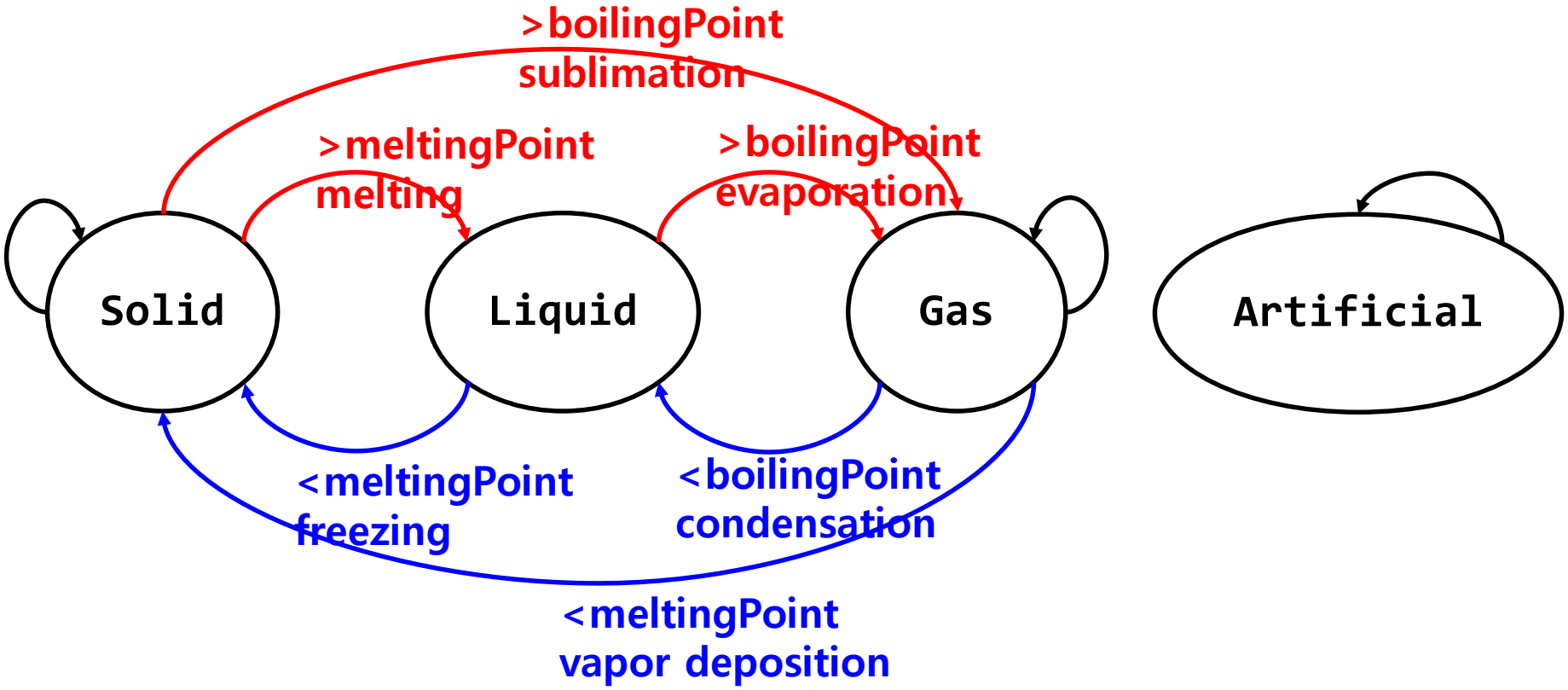
- Practice to write a program that **Element** or **ChemicalCompound** change its **State** by temperature at *meltingPoint* or *boilingPoint*.
  - **State<T>, MatterState<T> interface**
  - **ArtificialState, GasState, LiquidState, SolidState**

Lab9


MatterState
T
getName(): String
getGasState(): State<T>
getLiquidState(): State<T>
getSolidState(): State<T>
getArtificialState(): State<T>
getState(): State<T>
setState(state: State<T>): void
getMeltingPoint(): Optional<Double>
setMeltingPoint(meltingPoint: Optional<Double>): void
getBoilingPoint(): Optional<Double>
setBoilingPoint(boilingPoint: Optional<Double>): void
setTemperature(temperature: double): void
Element
number: int
name: String
state:State<Element>
meltingPoint: Optional<Double>
boilingPoint: Optional<Double>
ChemicalCompound
name: String
symbol: String
compounds: Map<Element, Integer>
weight: double
state:State<ChemicalCompound>
meltingPoint: Optional<Double>
boilingPoint: Optional<Double>
State
T
set(t: T): void
setTemperature(temperature: double): void
ArtificialState
T extends MaterialState<T>
GasState
T extends MaterialState<T>
LiquidState
T extends MaterialState<T>
SolidState
T extends MaterialState<T>

Lab9

- **State** Finite State Machine(FSM)



Lab9

```
public interface State<T> {
    void set(T element);
    void setTemperature(double temperature);
}
public interface MatterState<T> {
    String getName();
    State<T> getArtificialState();
    State<T> getGasState();
    State<T> getLiquidState();
    State<T> getSolidState();
    State<T> getState();
    void setState(State<T> state);
    Optional<Double> getMeltingPoint();
    void setMeltingPoint(Optional<Double> meltingPoint);
    Optional<Double> getBoilingPoint();
    void setBoilingPoint(Optional<Double> boilingPoint);
    void setTemperature(double temperature);
}
```

Lab9

```
public class Element implements MatterState<Element> {
    private int number;
    private String name;
…
    private State<Element> state;
    private Optional<Double> meltingPoint;
    private Optional<Double> boidingPoint;
    @Override
    public void setState(State<Element> state) {
        this.state = state;
        this.state.set(this);
    } …
    @Override
    public void setTemperature(double temperature) {
        this.state.setTemperature(temperature);
    }
…
}
```

Lab9

```
public class ChemicalCompound implements
MatterState<ChemicalCompound> {
    private String name;
    private String symbol;
…
    private State<ChemicalCompound> state;
    private Optional<Double> meltingPoint;
    private Optional<Double> boidingPoint;
    @Override
    public void setState(State<ChemicalCompound> state) {
        this.state = state;
        this.state.set(this);
    }…
    @Override
    public void setTemperature(double temperature) {
        this.state.setTemperature(temperature);
    }
}
```

Lab9

```
public class LiquidState<T extends MatterState<T>> implements
State<T> {
    private T t;
    @Override
    public void set(T t) {
        this.t = t;
    }
    @Override
    public void setTemperature(double temperature) {
      boolean changeState = false;
      if (t.getMeltingPoint().isPresent()) {
          if (temperature <= t.getMeltingPoint().get()) {
            System.out.println("Liquid is solidifying.");
            t.setState(t.getSolidState());
            changeState = true;
          }
      }
```

Lab9

```
        if (!changeState && t.getBoilingPoint().isPresent()) {
          if (temperature >= t.getBoilingPoint().get()) {
              System.out.println("Liquid is vaporizing.");
              t.setState(t.getGasState());
              changeState = true;
          }
        }
        if (changeState) {
          System.out.println(t.getName() + " state changed: " +
t.getState());
        }
    }
    @Override
    public String toString() {
        return "liq";
    }
}
```

Lab9 - JSON

- PeriodicElementsDetails.csv use Kevin (- 273.15 to C)
- PeriodicElementsDetails.json & ChemicalCompoundDetails.json
- FileImporter<E> interface
  - ChemicalCompoundJSONImporter
  - PeriodicElementJSONImporter
- JSONDeserializer<E> interface
  - DoubleOptionalDeserializer
  - ChemicalCompoundDeserializer
  - PeriodicElementDeserializer
- JSONSerializer<E> interface
  - DoubleOptionalSerializer
  - ChemicalCompoundSerializer
  - PeriodicElementSerializer

```
public class DoubleOptionalDeserializer implements
JsonDeserializer<Optional<Double>> {
    @Override
    public Optional<Double> deserialize(JsonElement json, Type typeOfT,
JsonDeserializationContext context) {
        if (json.isJsonPrimitive() &&
json.getAsJsonPrimitive().isNumber()) {
            return Optional.of(json.getAsDouble());
        } else {
            return Optional.empty();
        }
    }
}
public class DoubleOptionalSerializer implements
JsonSerializer<Optional<Double>> {
    @Override
    public JsonElement serialize(Optional<Double> src, Type typeOfSrc,
JsonSerializationContext context) {
        if (src.isPresent()) {
            return new JsonPrimitive(src.get());
        } else {
            return JsonNull.INSTANCE;
        }
    }
}
```

```
public static void main(String[] args) {

System.out.println("\nload Element from json file");
List<Element> list = new
ElementJSONImporter().importFile("PeriodicElementsDetail.json");
// state change of Element at temperature 22 ~ 5000 & 22 ~ -300
double temperature = 22;
for (temperature = 22; temperature < 5000; temperature += 10) {
   System.out.println("\n\nElement state at temperature = " +
temperature);
   for (Element e: list) { e.setTemperature(temperature); }
}
for (double temperature = 22; temperature > -300; temperature -= 10) {
   System.out.println("\n\nElement state at temperature = " +
temperature);
   for (Element e: list) { e.setTemperature(temperature); }
}
System.out.println("\nload ChemicalCompound from json file");
List<ChemicalCompound> list2 = new
ChemicalCompoundJSONImporter().importFile("ChemicalCompoundDetail.json
");
// state change of ChemicalCompound at 22C ~ 1500C & 22C ~ -300C
…
}
```

Submit to e-learning

- Add your code (e.g., additional method, class, routine, etc) in the Lab9 assignment.
- Submit the Lab9 assignment (JAVA23-2-Lab9-YourID-YourName.zip including the report) to e-learning.