Daisy: A Block-Based Environment for Learning Data Modeling

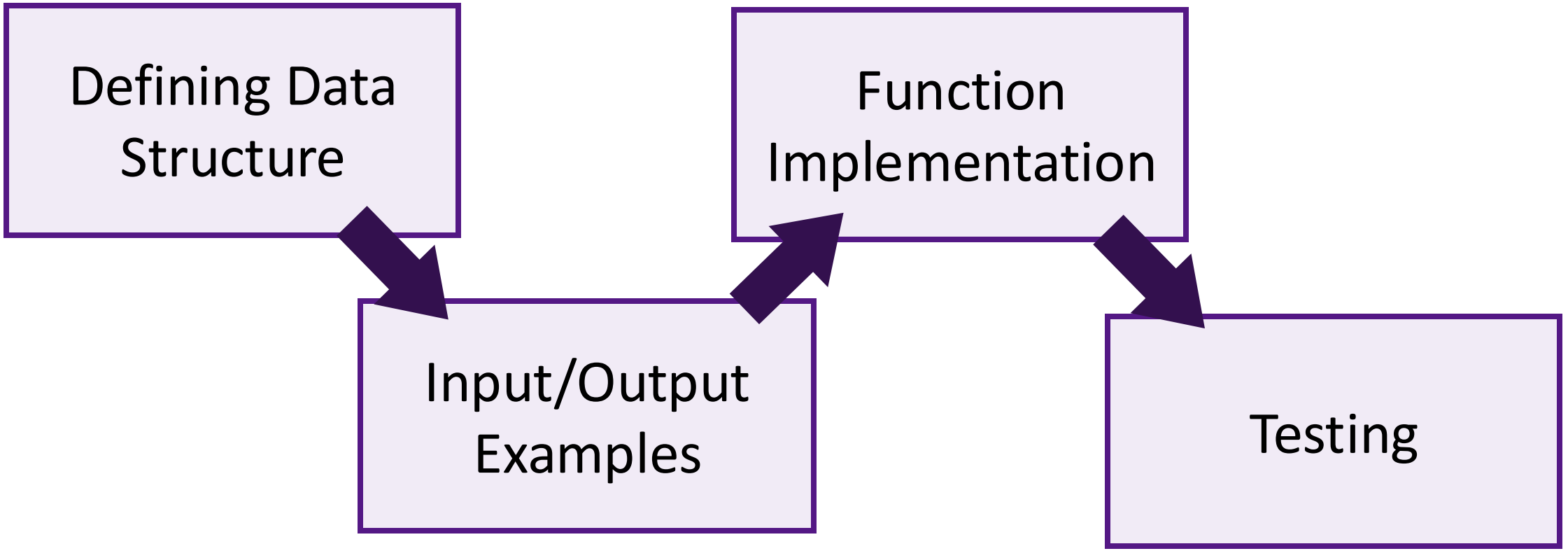
Jessica Belicia Cahyono, Youyou Cong, Hidehiko Masuhara

Tokyo Institute of Technology
(Institute of Science Tokyo from October 2024)

IFL 2024
Nijmegen, The Netherlands

Background

Problem Solving through Programming



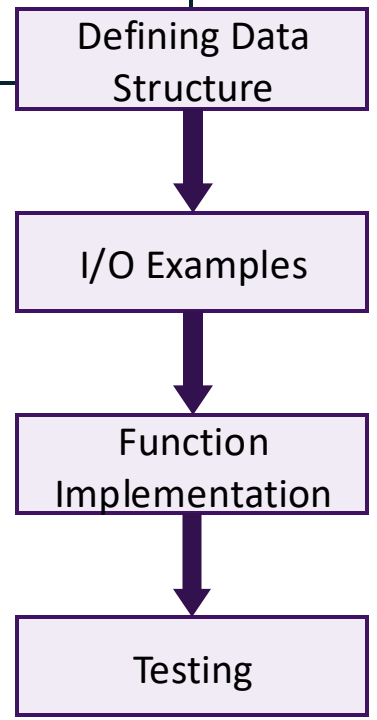
Background

Problem Solving through Programming

Example Problem

Define the function `area` to calculate the area of a shape.

The shape is either a square or a triangle.



Background

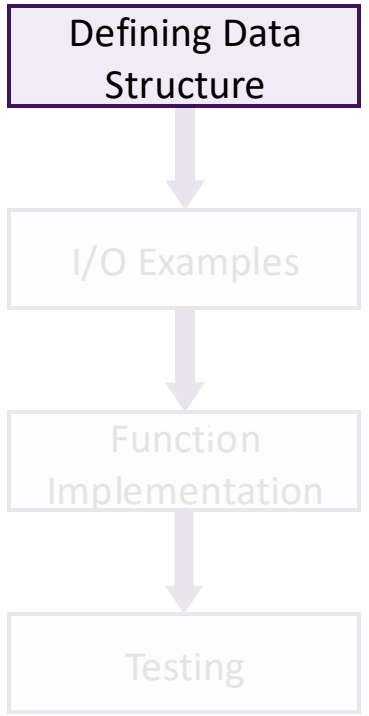
Problem Solving through Programming

Example Problem

Define the function `area` to calculate the area of a shape.
The shape is either a square or a triangle.

```
enum Shape:
  case Square(side: Int)
  case Triangle(base: Int, height: Int)
```

Scala 3



Background

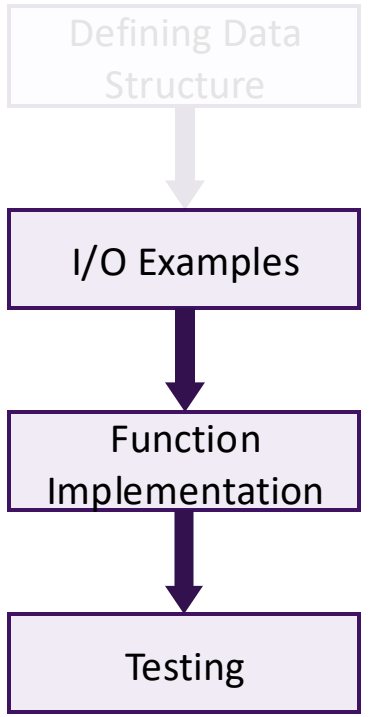
Problem Solving through Programming

Example Problem

Define the function `area` to calculate the area of a shape.
The shape is either a square or a triangle.

```
enum Shape:
  case Square(side: Int)
  case Triangle(base: Int, height: Int)

shape1 = Shape.Square(4)
shape2 = Shape.Triangle(1,2)

def area(shape: Shape): Double =
  shape match
    case Shape.Square(s)     => s * s
    case Shape.Triangle(b,h) => 0.5 * b * h

area(shape1) == 16
area(shape2) == 1
```

Scala 3



Background

Problem Solving through Programming

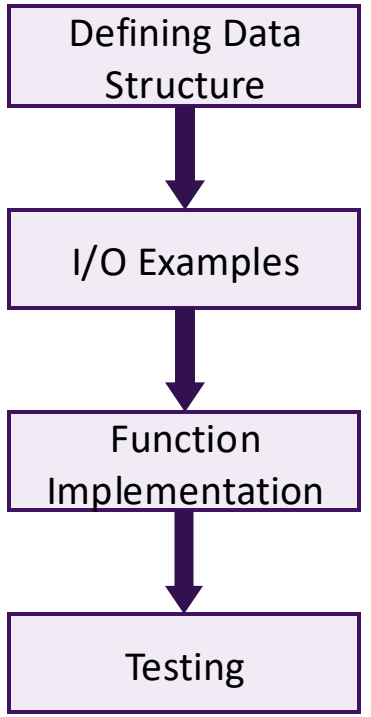
Example Problem

Define the function `area` to calculate the area of a shape.
The shape is either a square or a triangle.

```
enum Shape:
  case Square(side: Int)
  case Triangle(base: Int, height: Int)

shape1 = Shap
shape2 = Shap

def area(shape: Shape): Double =
  shape match
    case Shape.Square(s)      => s * s
    case Shape.Triangle(b,h) => 0.5 * b * h

area(shape1) == 16
area(shape2) == 1
```

Data Definition

Scala 3



Background

Problem Solving through Programming

Example Problem

Define the function `area` to calculate the area of a shape.
The shape is either a square or a triangle.

```
enum Shape:
  case Square(side: Int)
  case Triangle(base: Int, height: Int)
```

Data Definition

```
shape1 = Shap
shape2 = Shap

def area(shape: Shape): Double =
  shape match
    case Shape.Square(s)     => s * s
    case Shape.Triangle(b,h) => 0.5 * b * h

area(shape1) == 16
area(shape2) == 1
```

Scala 3

Data Modeling

- I/O Examples
- Function Implementation
- Testing

Background

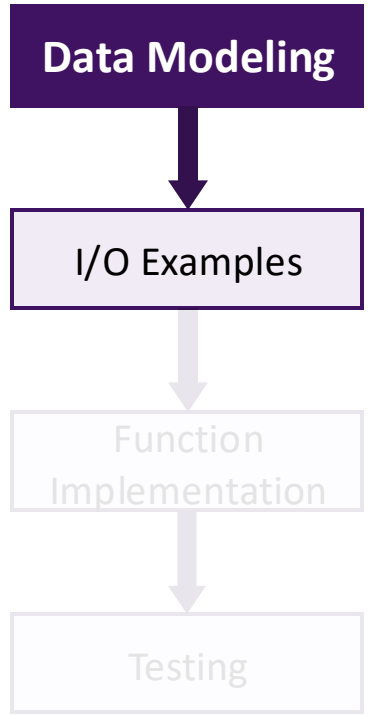
Importance of Data Modeling Process

Data modeling **drives the planning of the rest of the program**

[Felleisen+, 2018]

```
enum Shape:
  case Square(side: Int)
  case Triangle(base: Int, height: Int)

shape1 = Shape.Square(4)
shape2 = Shape.Triangle(1,2)

def area(shape: Shape): Double =
  shape match
    case Shape.Square(s)     => s * s
    case Shape.Triangle(b,h) => 0.5 * b * h

area(shape1) == 16
area(shape2) == 1
```

Scala 3



Background

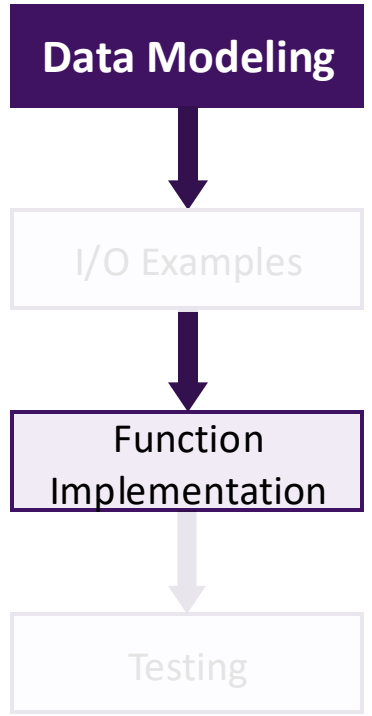
Importance of Data Modeling Process

Data modeling **drives the planning of the rest of the program**

[Felleisen+, 2018]

```
enum Shape:
  case Square(side: Int)
  case Triangle(base: Int, height: Int)

shape1 = Shape.Square(4)
shape2 = Shape.Triangle(1,2)

def area(shape: Shape): Double =
  shape match
    case Shape.Square(s)      => s * s
    case Shape.Triangle(b,h) => 0.5 * b * h

area(shape1) == 16
area(shape2) == 1
```

Scala 3



Problem: Data Modeling is Difficult for Novices

1. Data modeling process is too abstract

2. Need for knowledge of programming language syntax & ADT concept

3. Lack of feedback on the outcome

Problem #1

Data Modeling is Too Abstract

Problem Description

Define the function `area` to calculate the area of a shape. The shape is either a square or a triangle.

ADT

Scala 3

```
enum Shape:
   case Square(side: Int)
   case Triangle(base: Int,
                 height: Int)
```

What should I do?

Problem #2

Need for Syntax & Concept Knowledge

Problem Description

Define the function area to calculate the area of a shape. The shape is either a square or a triangle.

ADT

Scala 3

```
enum Shape:
   case Square(side: Int)
   case Triangle(base: Int,
                 height: Int)
```

I don't know the difference between constructor and data type

How should I define a data type in Scala?

Problem #3

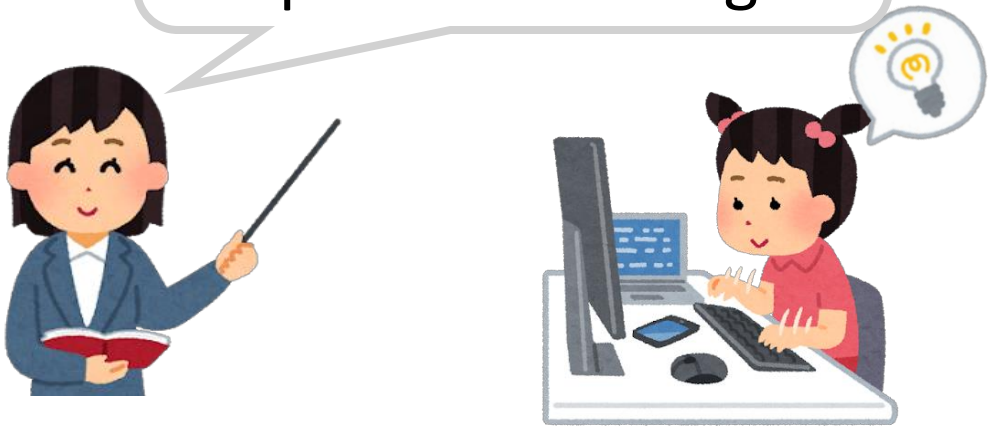
Lack of Feedback

Incorrect ADT

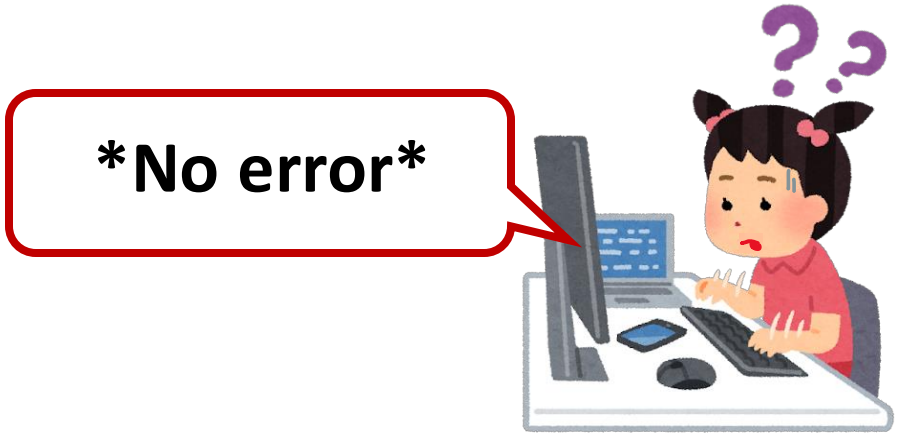
```
enum Shape:
    case Square(side: Int)
    case Triangle(base: Int)
```

Scala 3

Teacher



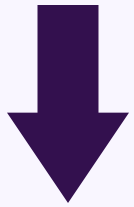
Compiler



Goal: Assist Data Modeling Learning

Contributions:

Data modeling is too abstract



✅ **Refined data modeling process**

Need for knowledge of syntax & concept

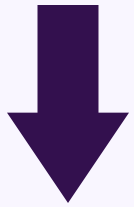
Lack of feedback on outcome

✅ **Daisy:**
a block-based environment for solving data modeling exercise

Goal: Assist Data Modeling Learning

Contributions:

Data modeling is too abstract



✅**Refined data modeling process**

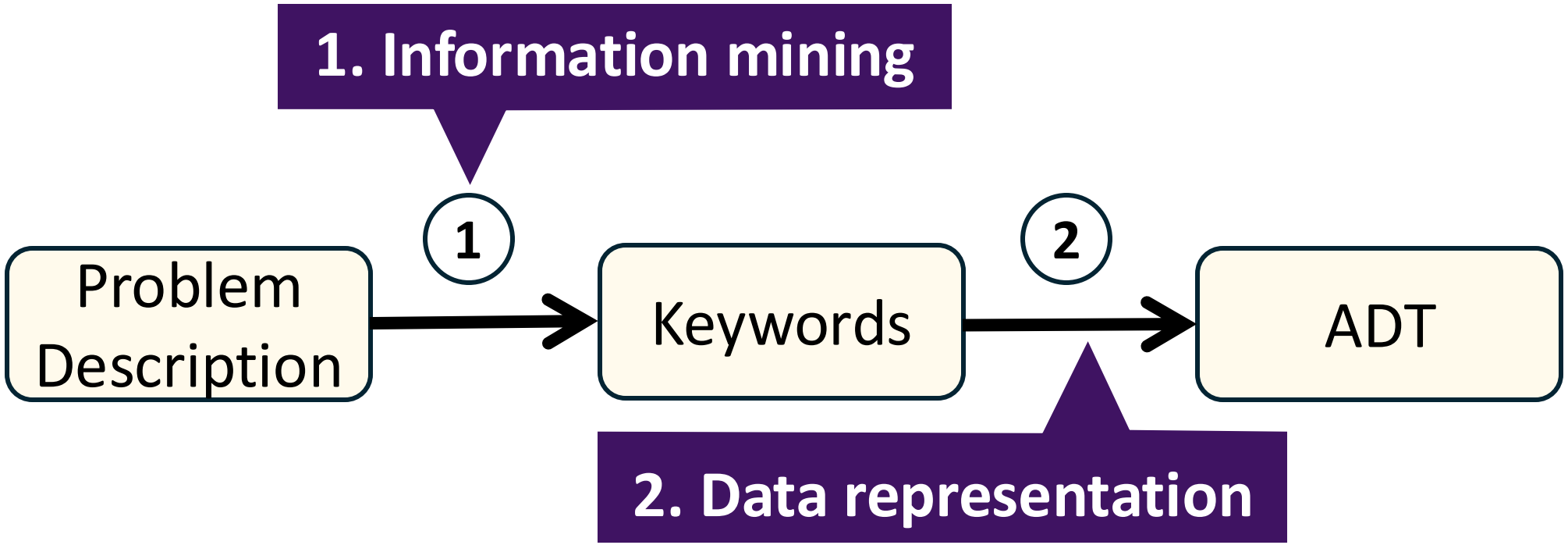
Need for knowledge of syntax & concept

Lack of feedback on outcome

✅Daisy:
a block-based environment
for solving data modeling exercise

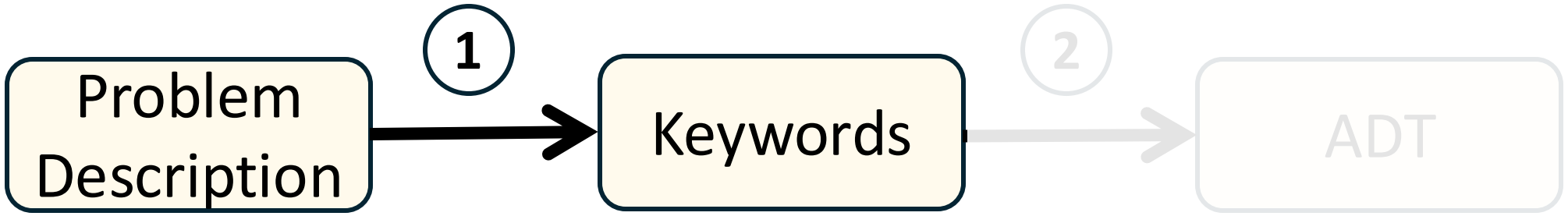
Refining Data Modeling Process

Refining Data Modeling Process

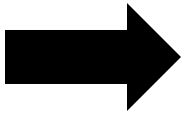


Refining Data Modeling Process

1. Information Mining

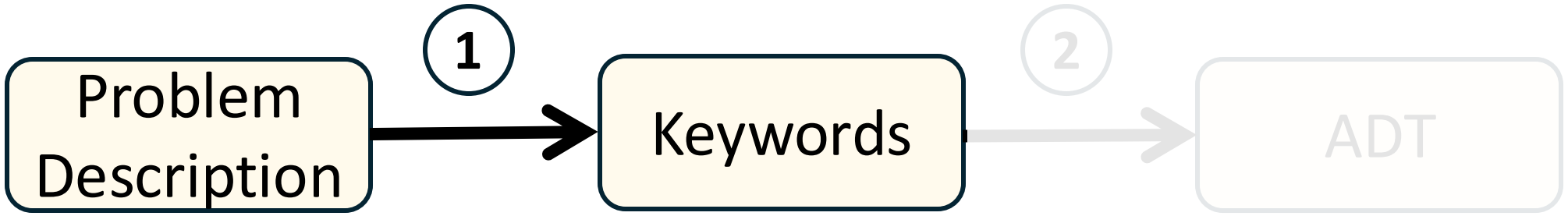


Define the function `area` to calculate the area of a shape. The shape is either a square or a triangle.

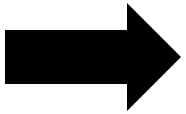


Refining Data Modeling Process

1. Information Mining



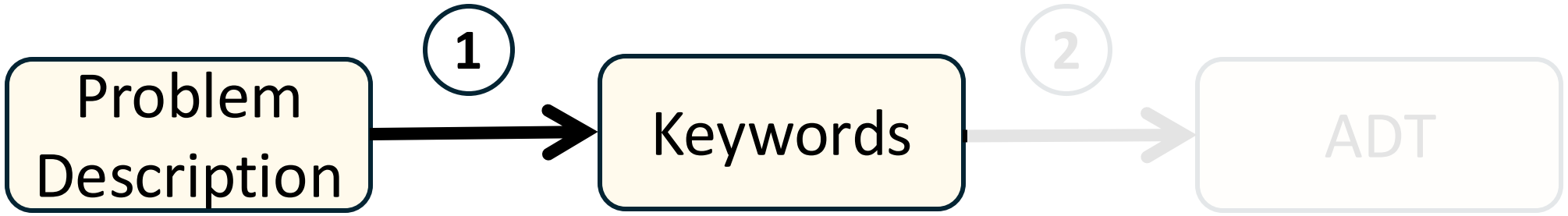
Define the function area to calculate the area of a **shape**.
The shape is either a square or a triangle.



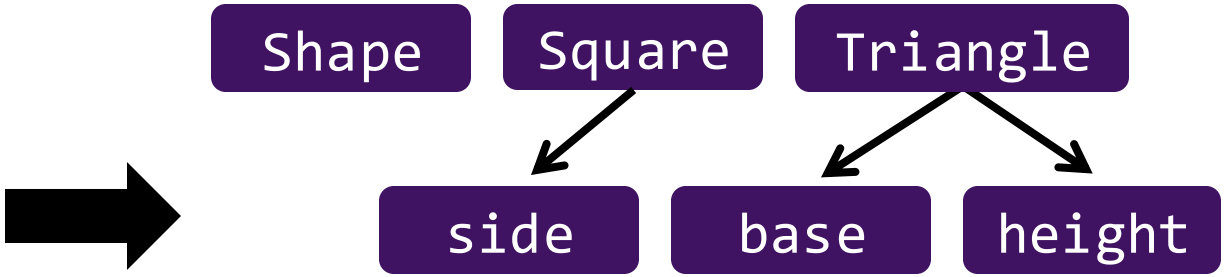
`Shape` `Square` `Triangle`

Refining Data Modeling Process

1. Information Mining

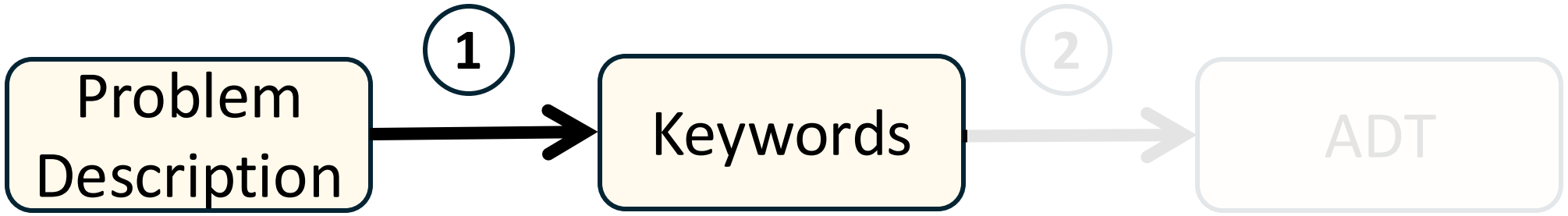


Define the function area to calculate the area of a **shape**.
The shape is either a square or a triangle.

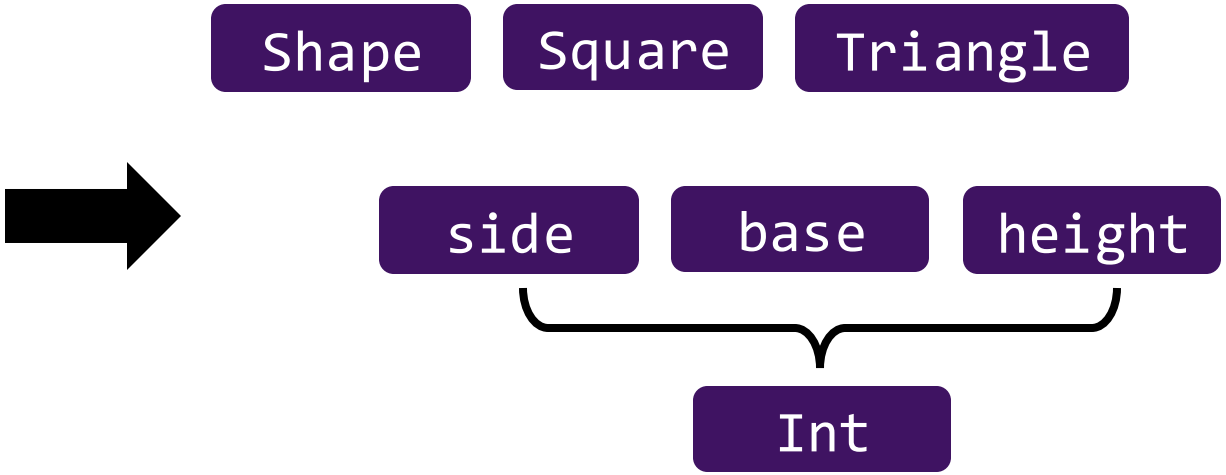


Refining Data Modeling Process

1. Information Mining

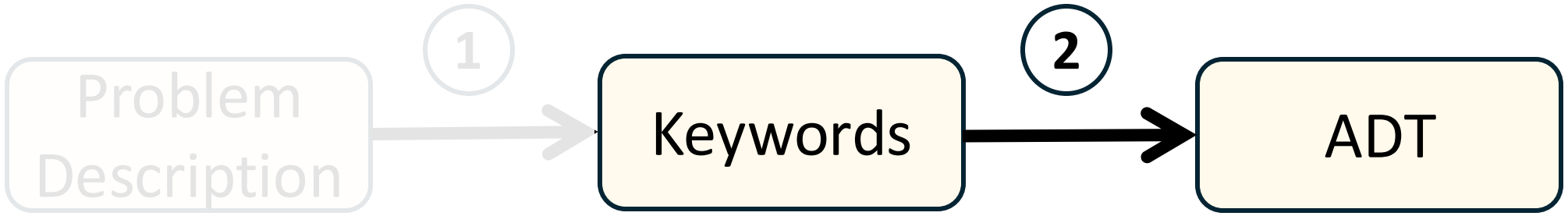


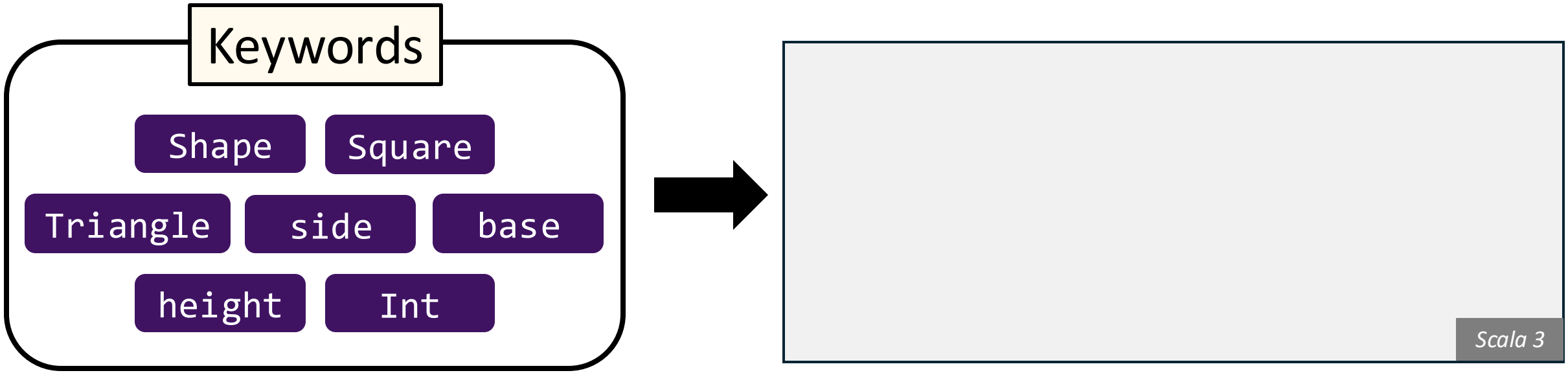
Define the function area to calculate the area of a **shape**.
The shape is either a square or a triangle.



Refining Data Modeling Process

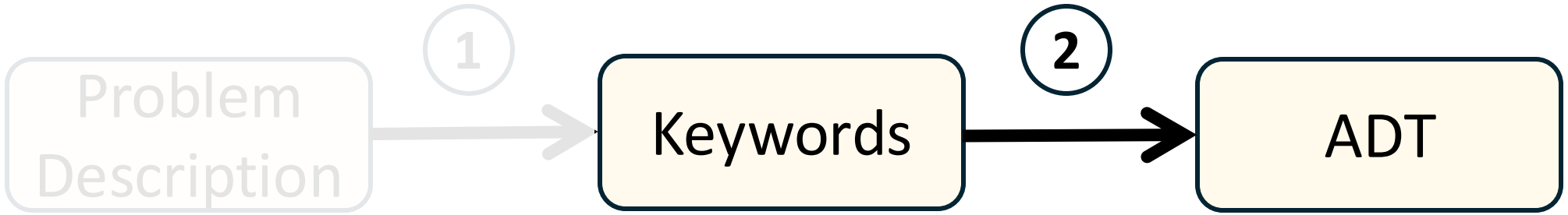
2. Data Representation

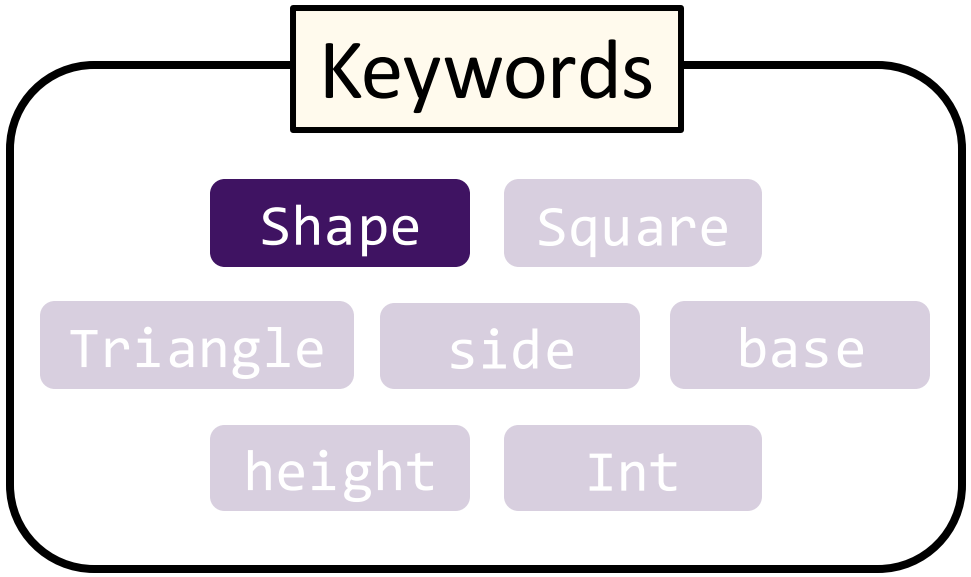




Refining Data Modeling Process

2. Data Representation



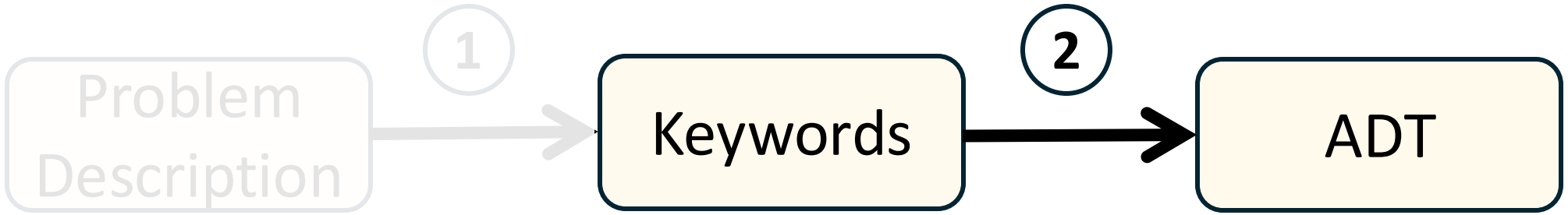


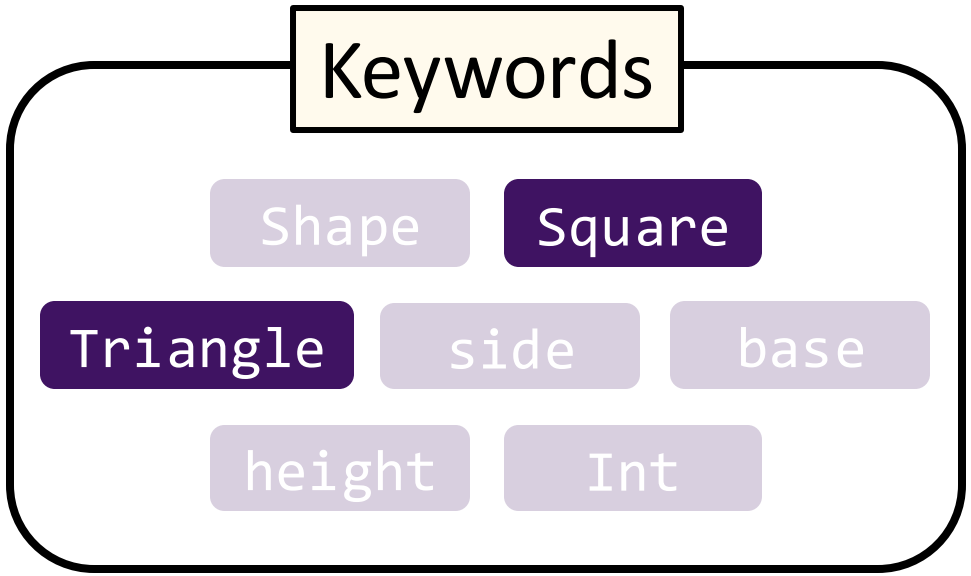

```
enum Shape :
```

Scala 3

Refining Data Modeling Process

2. Data Representation



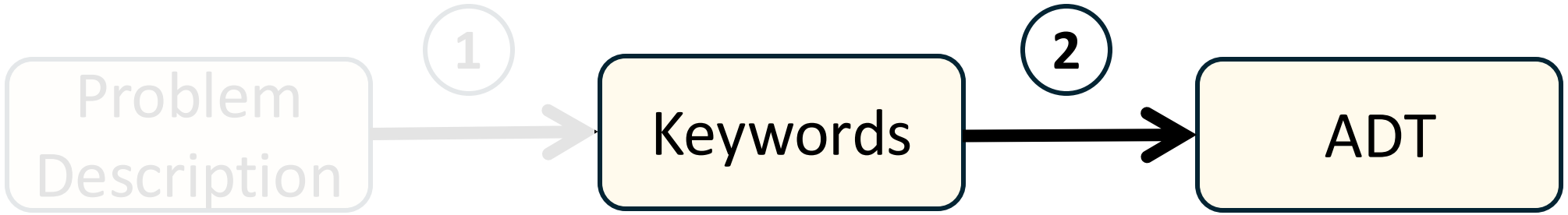


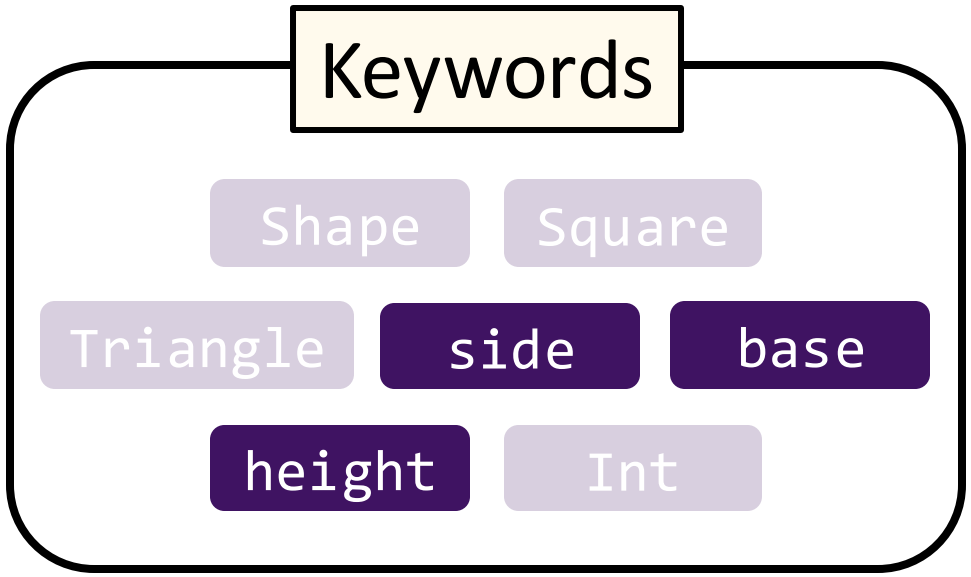

```
enum Shape :
    case Square
    case Triangle
```

Scala 3

Refining Data Modeling Process

2. Data Representation



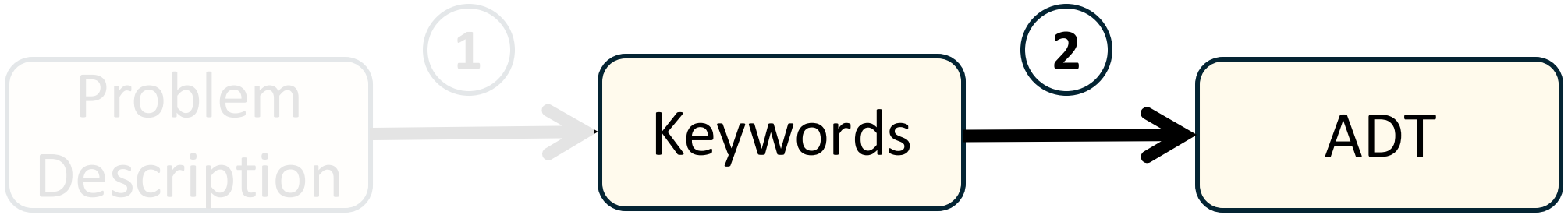


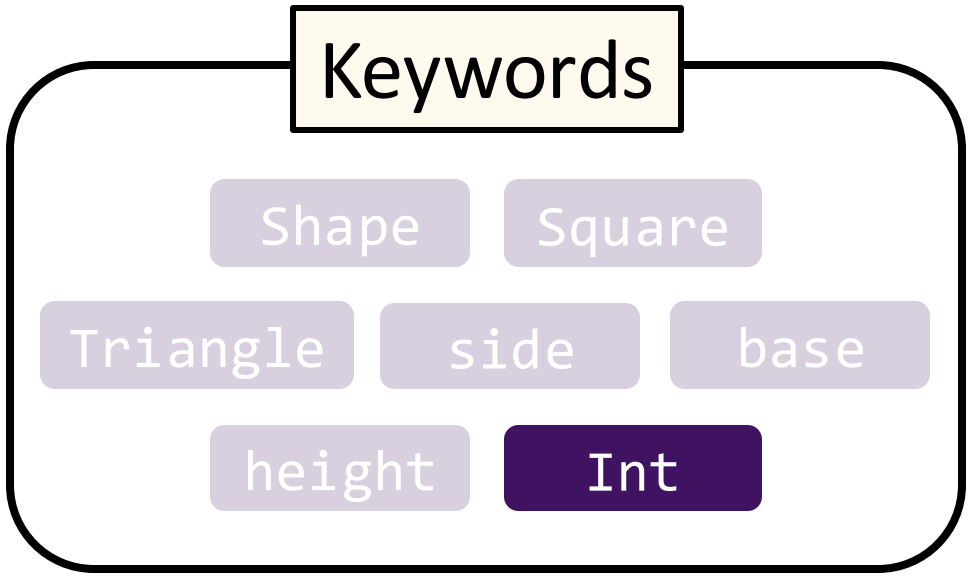

```
enum Shape :
    case Square ( side )
    case Triangle ( base , height )
```

Scala 3

Refining Data Modeling Process

2. Data Representation

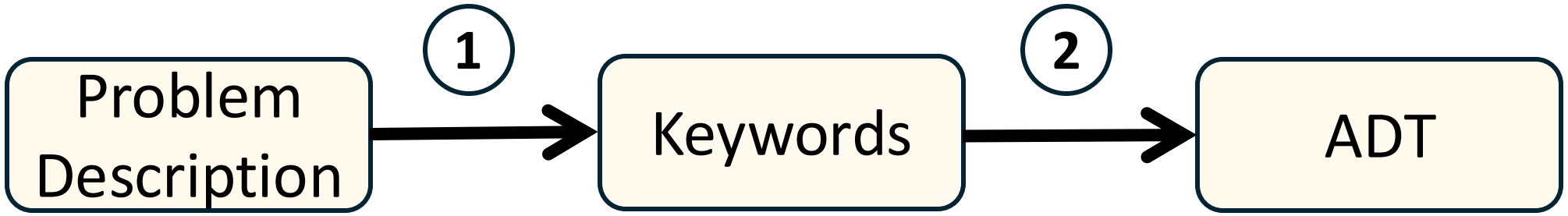




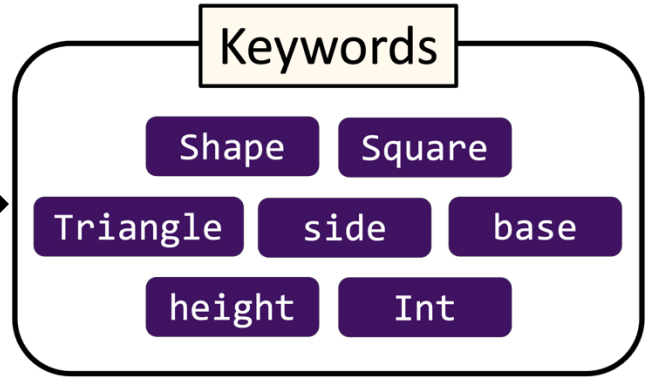

```
enum Shape :
    case Square ( side : Int )
    case Triangle ( base : Int ,
                    height : Int )
```

Scala 3

Refining Data Modeling Process



Define the function area to calculate the area of a shape. The shape is either a square or a triangle.




```
enum Shape:
  case Square(side: Int)
  case Triangle(base: Int,
                height: Int)
```

Scala 3

Goal: Assist Data Modeling Learning

Contributions:

Data modeling is too abstract

✅Refined data modeling process

Need for knowledge of syntax & concept

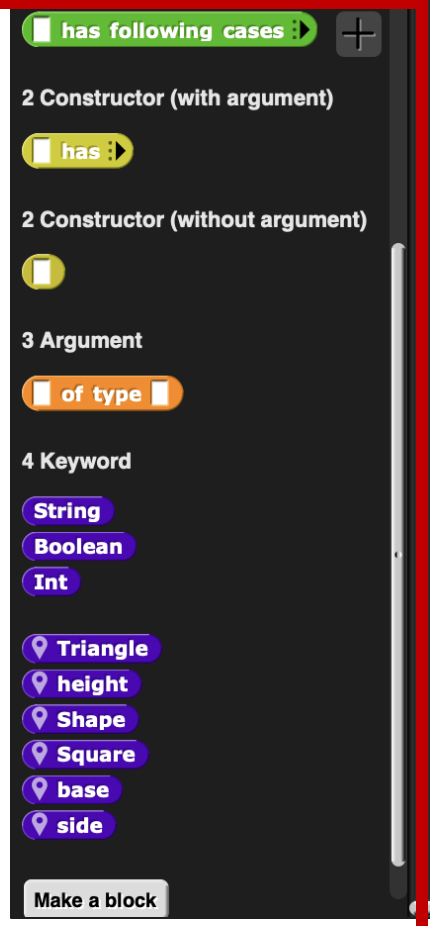
Lack of feedback on outcome

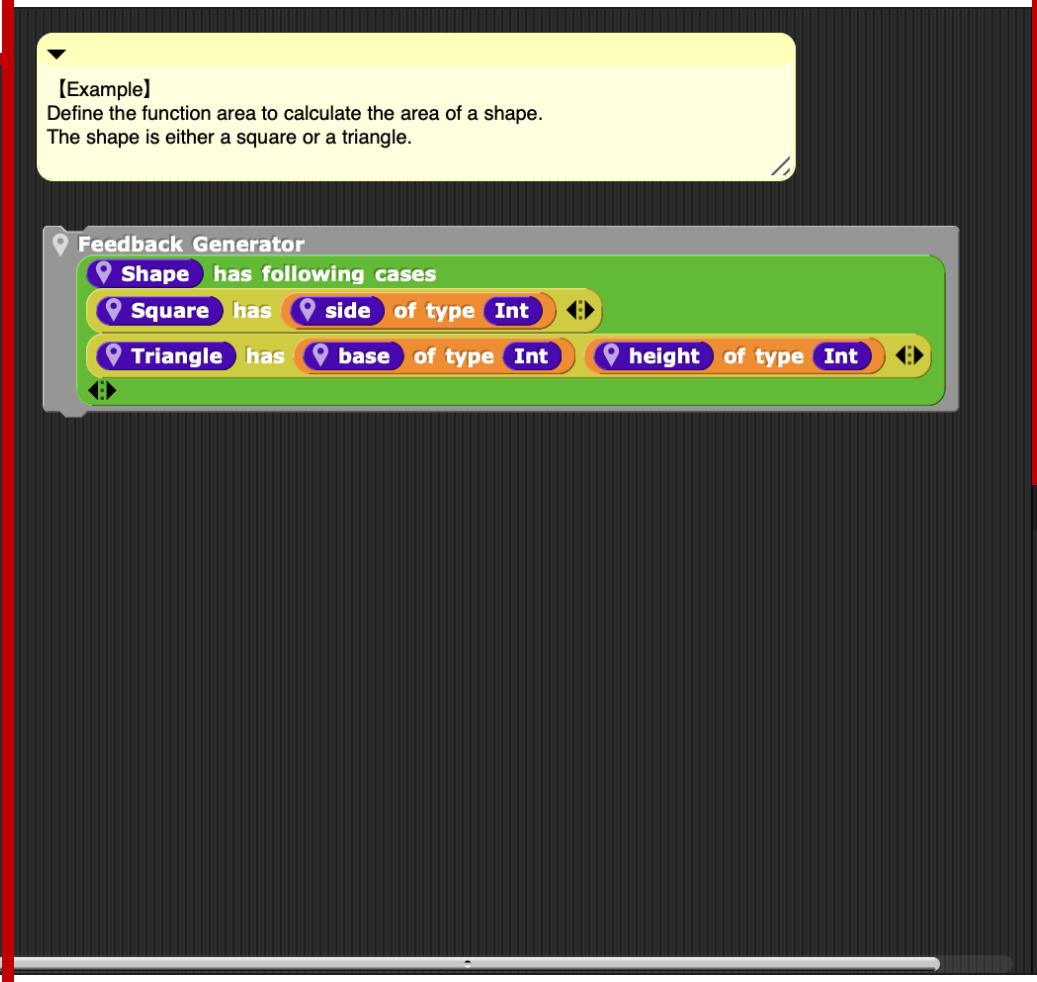
✅**Daisy:**
a block-based environment for solving data modeling exercise

The Daisy Environment

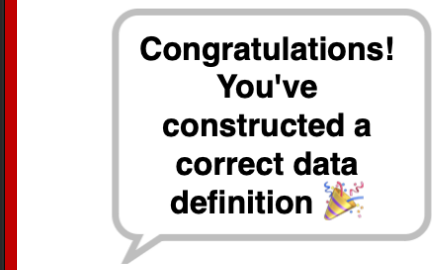
(data modeling is easy to learn)

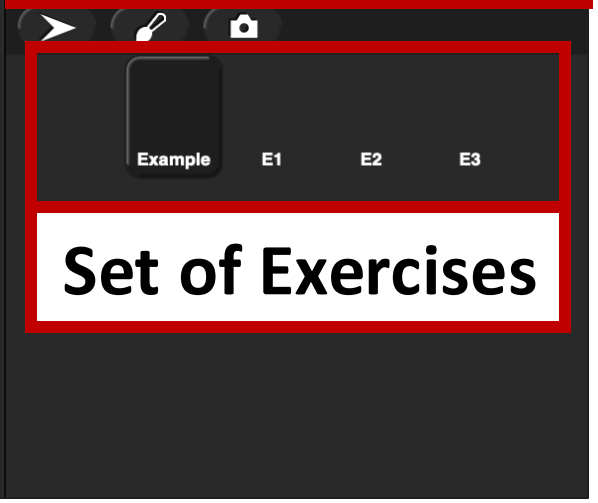
Blocks





Feedback





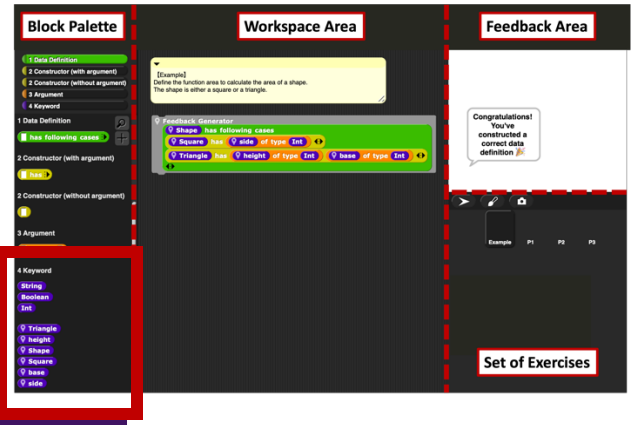
Set of Exercises

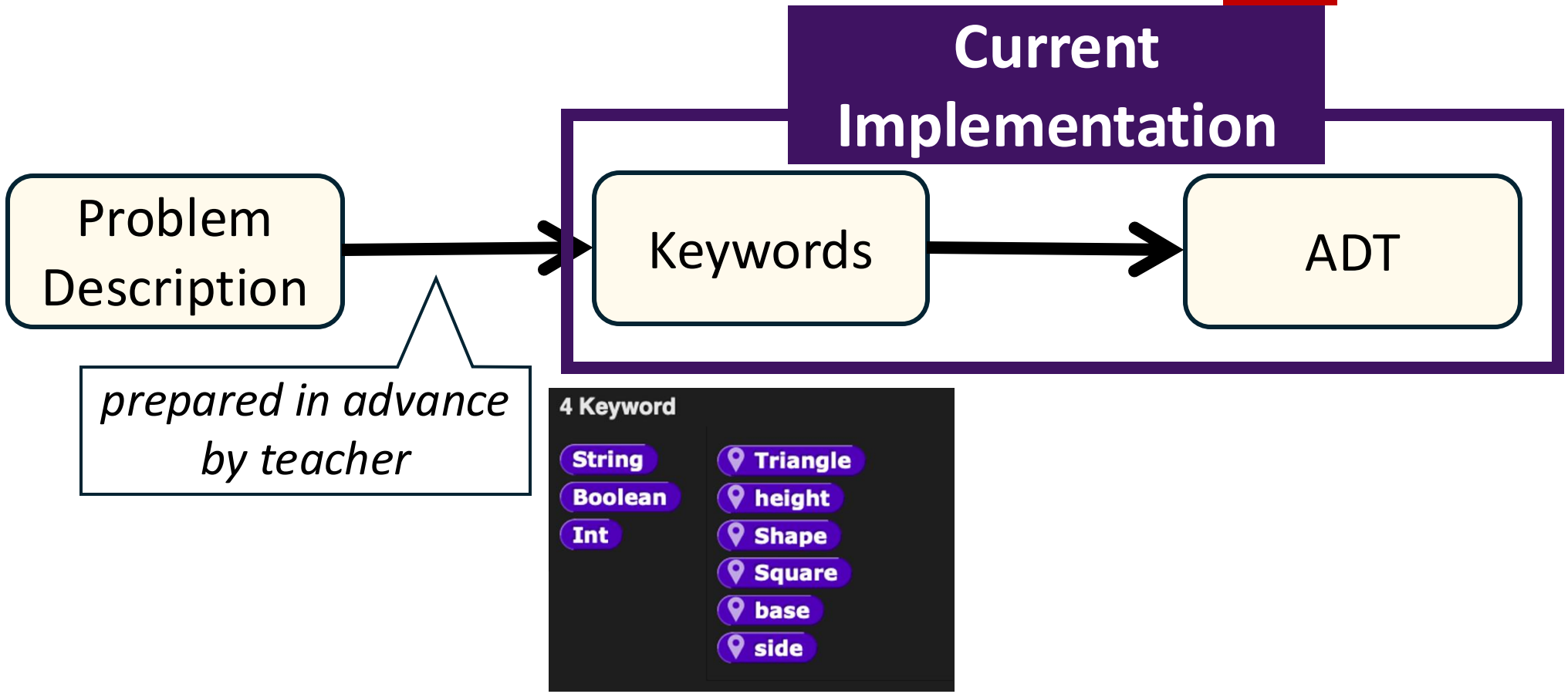
Features of Daisy Environment

- ✓ Support for data modeling
- ✓ Use of natural language
- ✓ Automatic feedback generation

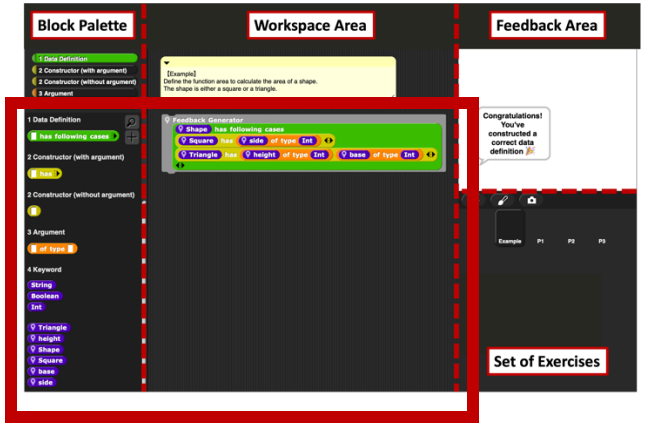
Features of Daisy

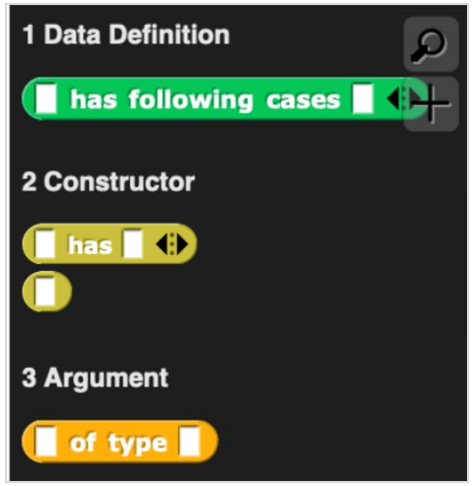
Support for Data Modeling





Features of Daisy
Use of Natural Language

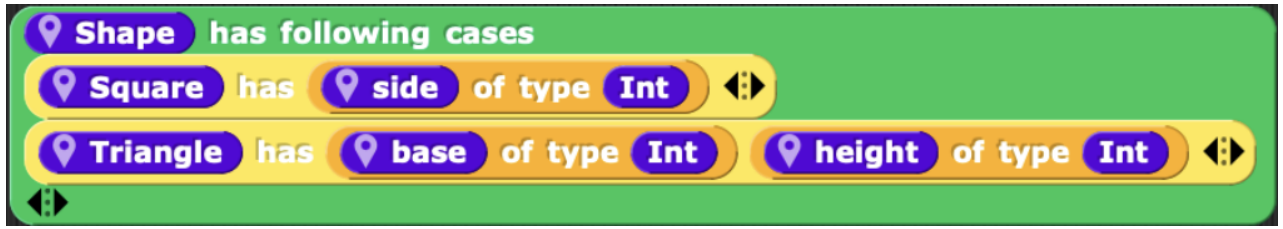




+

Int
Triangle
Shape
height

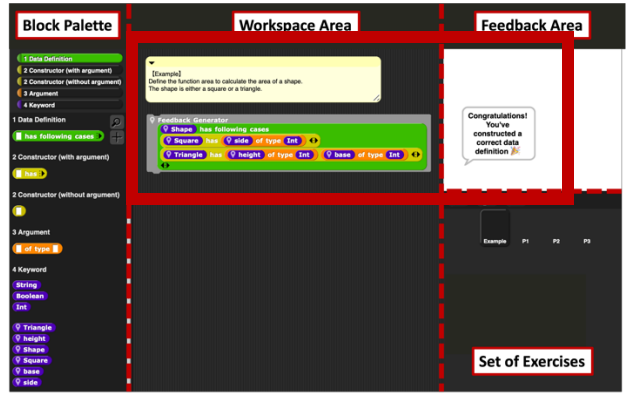
Square
base
side



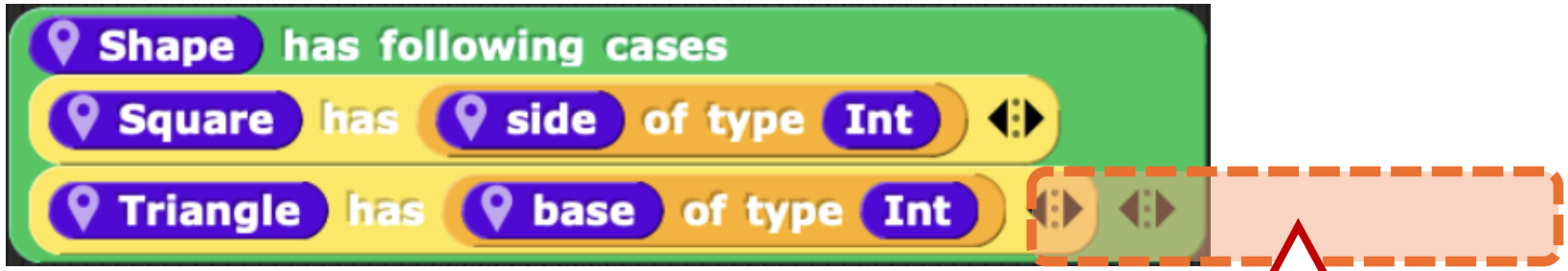
✓ Focus on the essential task of constructing ADT

Features of Daisy

Feedback Generation

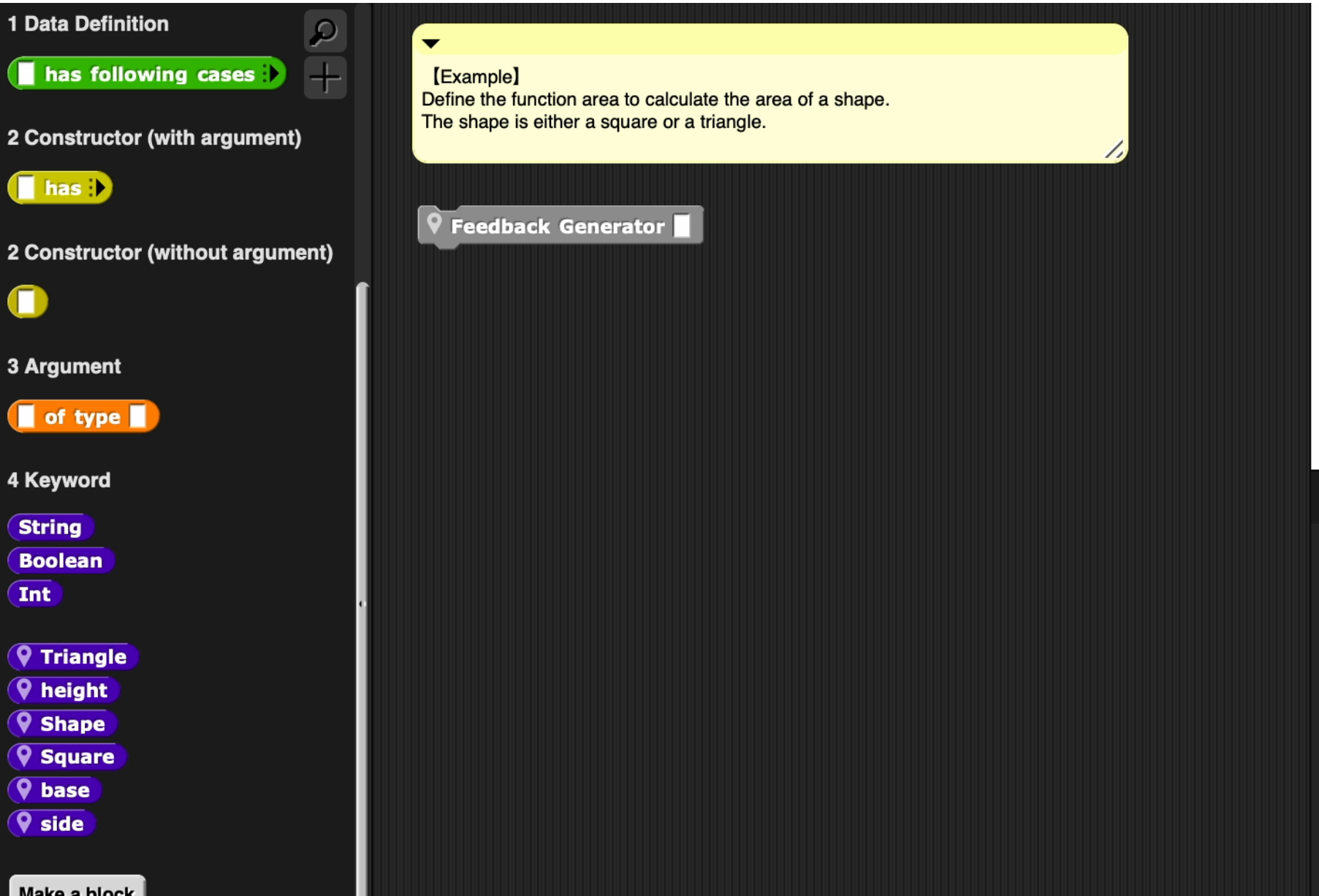


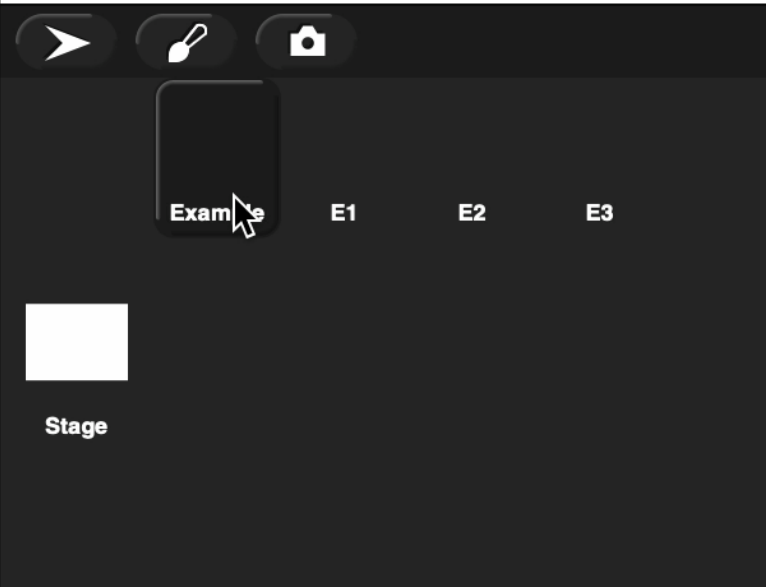
Comparison with the correct answer provided by teacher



Warning:
Incorrect **argument number** of Triangle

The Daisy Environment: Demo





Preliminary Experiment

Participants

27 undergraduate students with basic knowledge of FP

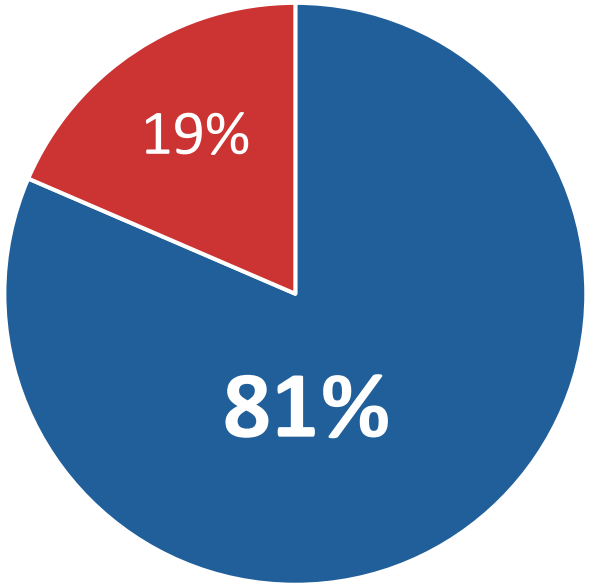
Tasks

- Solve three data modeling problems using Daisy (JP ver)
- Fill out questionnaire

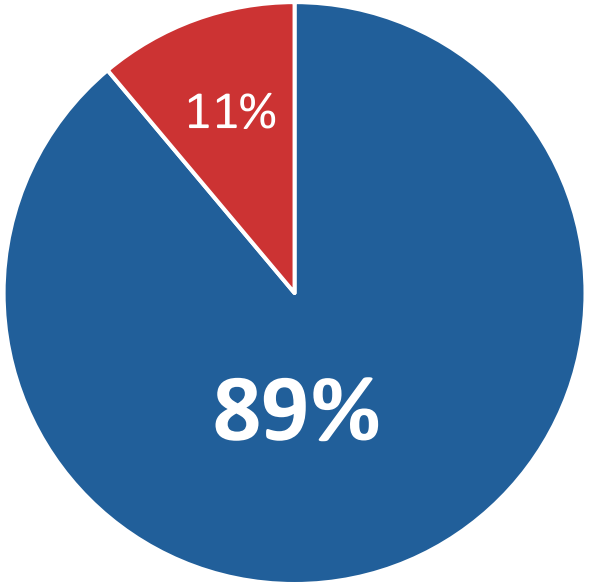
Results

1. Helpfulness of Daisy

Block Components



Automatic Feedback



Helpful

Not helpful

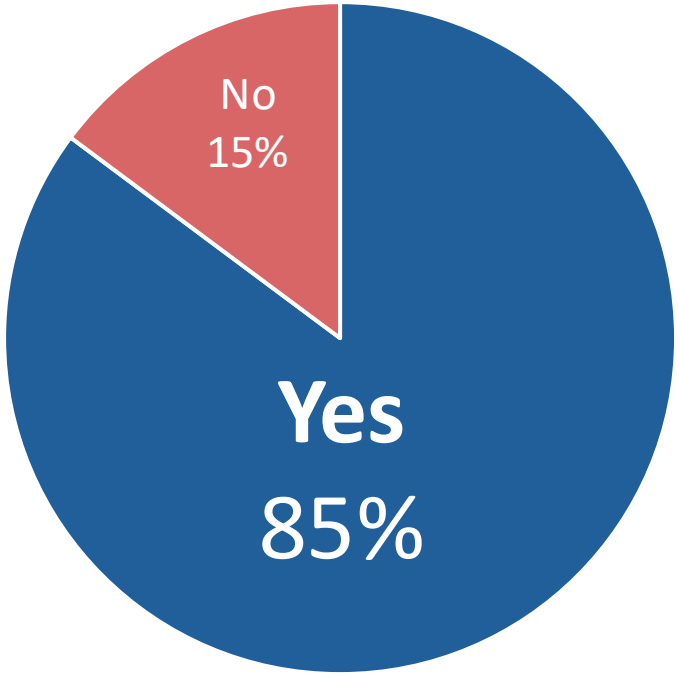
✅ **> 80%** students said Daisy is helpful

Results

2. Block Components

When constructing ADT,

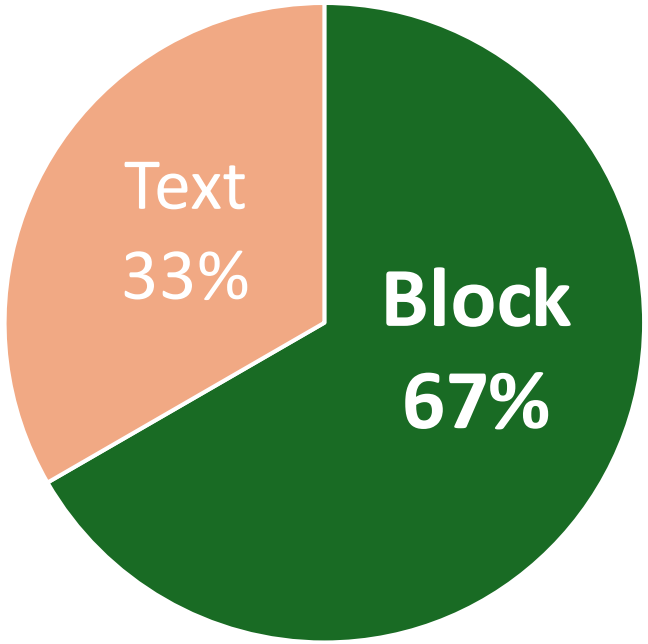
provision of constructor and argument names makes tasks easier



Results

2. Block Components

When constructing ADT, which one is **easier to use**?



Block makes constructing ADT ***easy*** and more ***intuitive***

Dragging-and-dropping block is ***troublesome***

Results

3. Automatic Feedback: Positive Comments

20 participants received feedback

Most of them received it on the recursive problem

The feedback helps me ***find the mistake faster***

The ***content*** of the feedback is on point!

Results

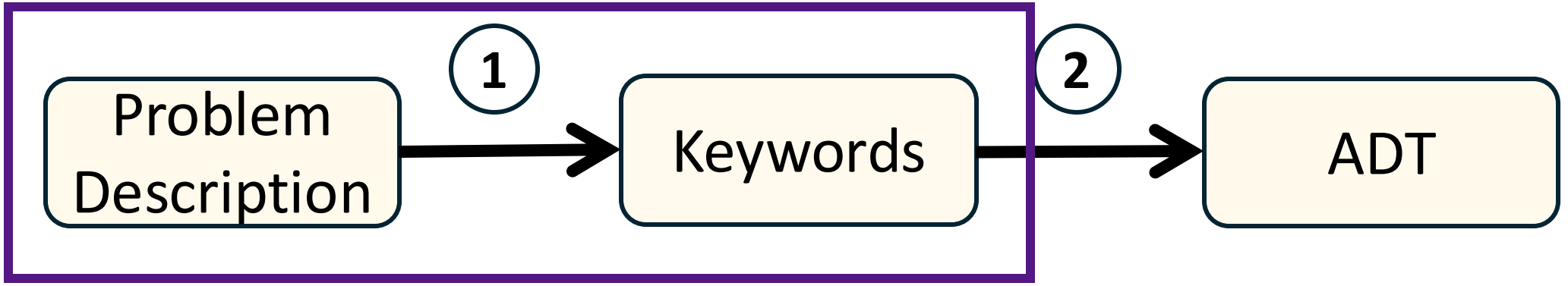
3. Automatic Feedback: Suggestions

Better to ***keep previous messages*** when generating new feedback

It would be nice if I can get the exact ***error location***

Future Work

1. Support Information Mining Step



2. Improve Feedback
3. Quantitative Assessment

Summary

Goal: Assist data modeling learning

Contribution: (1) Refine Data Modeling Process
(2) Develop Daisy

To-do:

- Implement information mining
- Improve feedback
- Assess Daisy

} **comments are welcome!**

Reference

Matthias Felleisen, Robert Bruce Findler, Matthew Flatt, and Shriram Krishnamurthi. 2018. *How to Design Programs: An Introduction to Programming and Computing*. The MIT Press.

Junya Nose, Youyou Cong, and Hidehiko Masuhara. 2022. Mio: A Block-Based Environment for Program Design. In *Proceedings of the 2022 ACM SIGPLAN International SPLASH-E Symposium (SPLASH-E '22), December 05, 2022, Auckland, New Zealand*. ACM, New York, NY, USA, 62-69. https://doi.org/10.1145/3563767.3568127

Appendix

Exercises in Experiment: Non-Recursive

Problem 1: Money

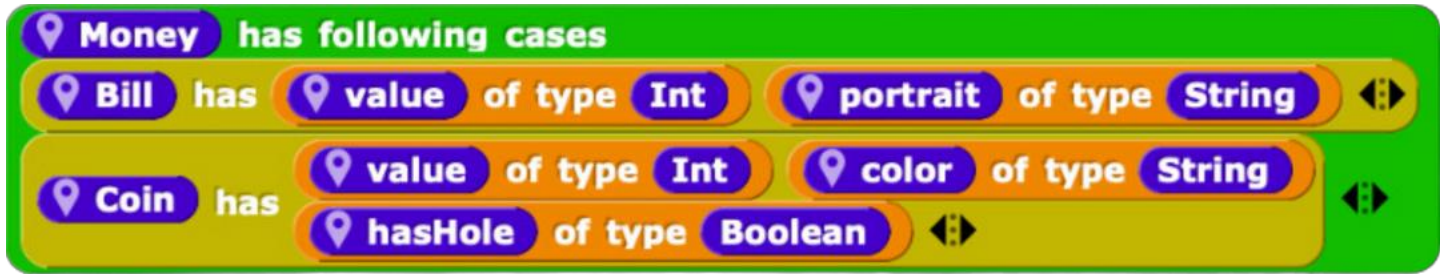
Define the data type Money that represents cash used in Japan.

Cash is either a banknote or a coin. A banknote has the information of its amount and the name of the person whose portrait is on it. A coin has the information of its amount, color, and whether it has a hole or not.

Keyword

String	Coin
Int	Portrait
Boolean	hasHole
Money	Bill
color	value

Answer



Exercises in Experiment: Non-Recursive

Problem 2: Device

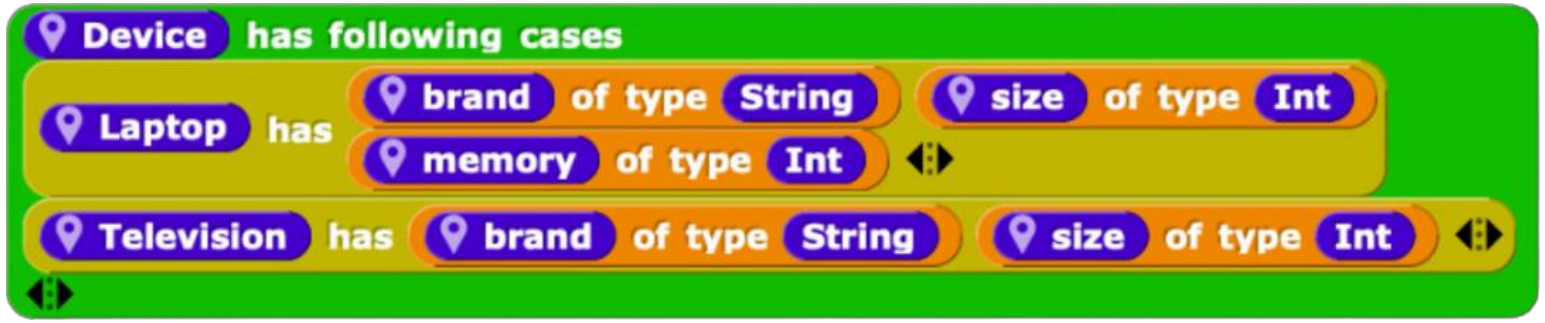
Define the data type Device that represents devices.

A device is either a laptop or a television.

Keyword

String
size
Int
Laptop
Boolean
memory
brand
Television
Device

Answer



Exercises in Experiment: Recursive

Problem 3: Path

Define the data type Path that represents the file location in a file system.
For example, /Home/Download/ex1.scala is a path to the ex1.scala file.
The path is either a file or a directory. A file has information of its name and extension. A directory has the information of its name and the file or directory inside it. We assume that each directory can only have one directory or file.

Keyword

String	File
Int	Path
Boolean	extension
name	Directory
content	

Answer