

Daisy: A Block-Based Environment for Learning Data Modeling

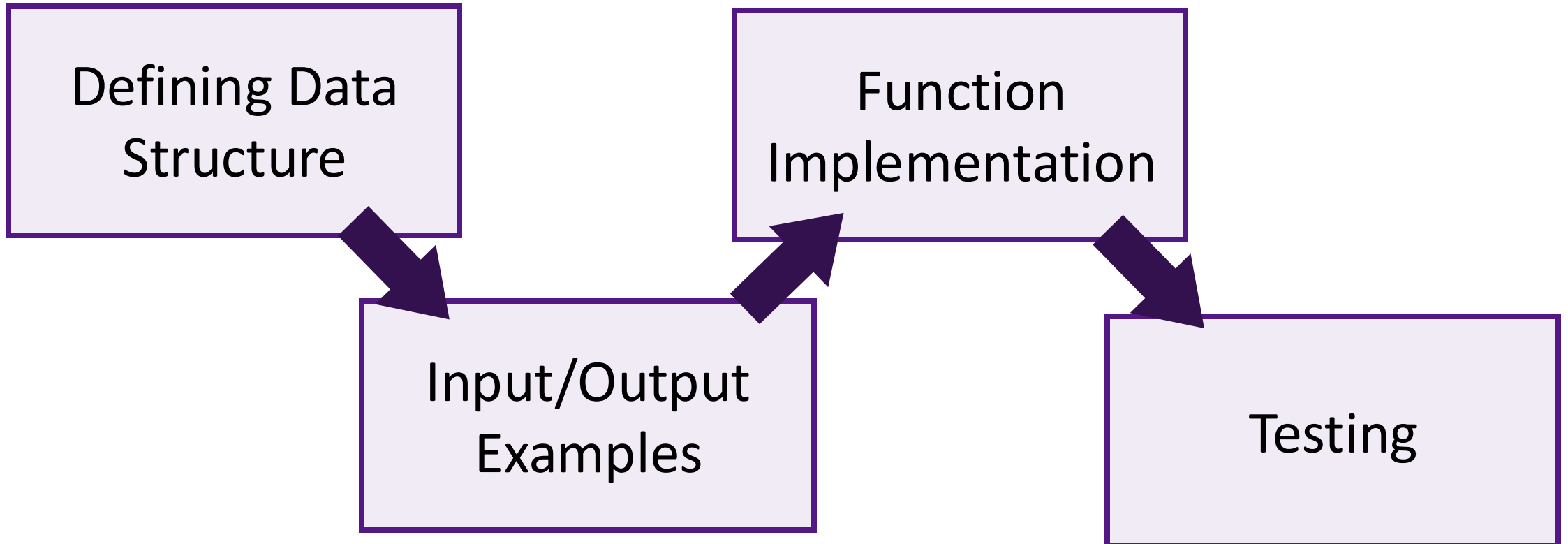
Jessica Belicia Cahyono, Youyou Cong, Hidehiko Masuhara

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

IFL 2024
Nijmegen, The Netherlands

Background

Problem Solving through Programming

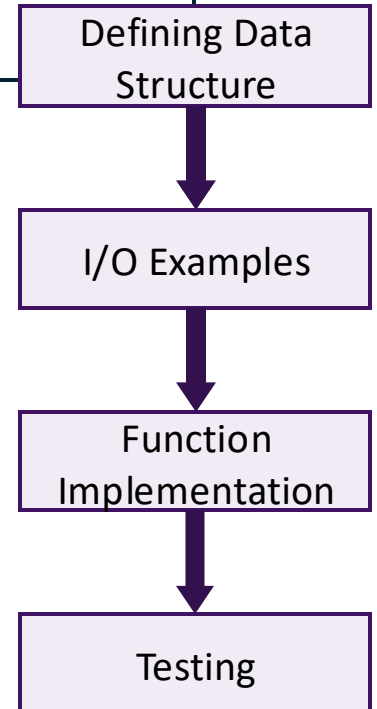


Background

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

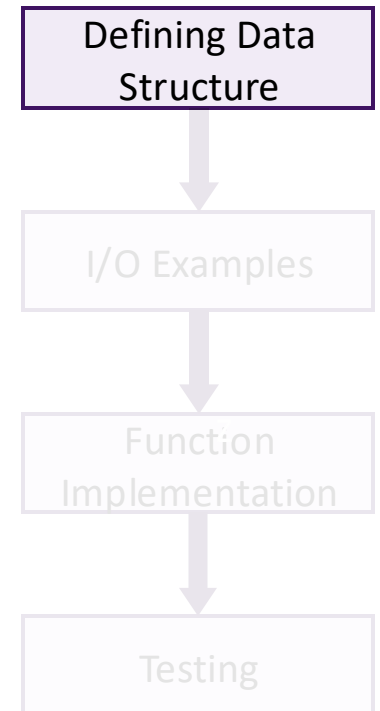
Problem Solving through Programming

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.

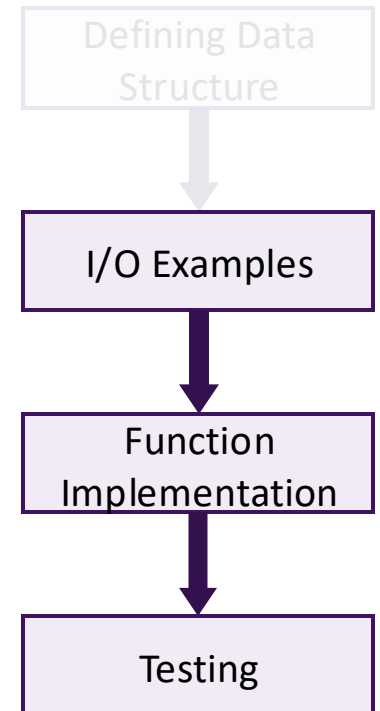
```
enum Shape:
  case Square(side: Int)
  case Triangle(base: Int, height: Int)

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

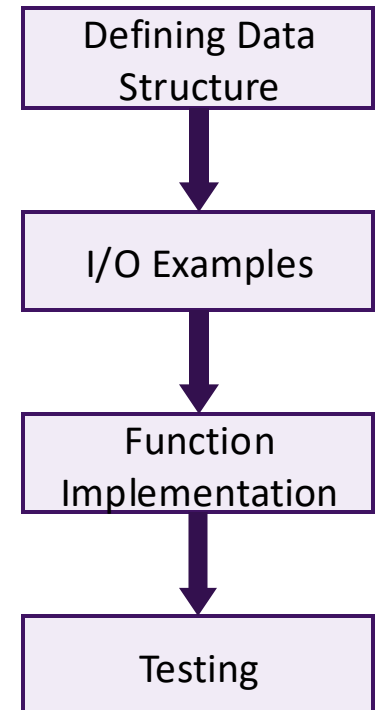
Define the function `area` to calculate the area of a shape. The shape is either a square or a triangle.

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

Data Definition

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

Data Definition

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

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

Data Modeling

I/O Examples

Function
Implementation

Testing

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

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

Scala 3



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

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

Scala 3



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

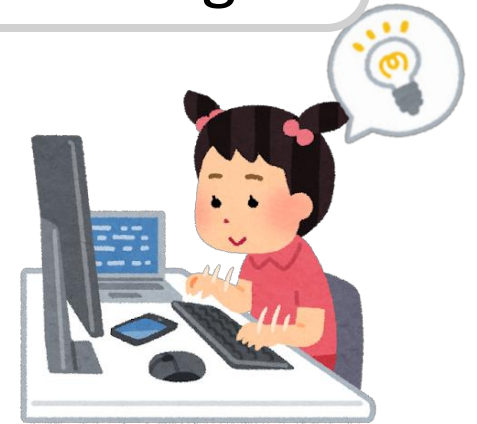
```
enum Shape:  
  case Square(side: Int)  
  case Triangle(base: Int)
```

Scala 3

Teacher



You need more than one argument to represent a triangle



Compiler

No error




Goal: Assist Data Modeling Learning

Contributions:

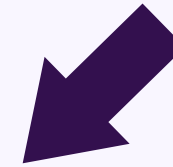
Data modeling is too abstract



 **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

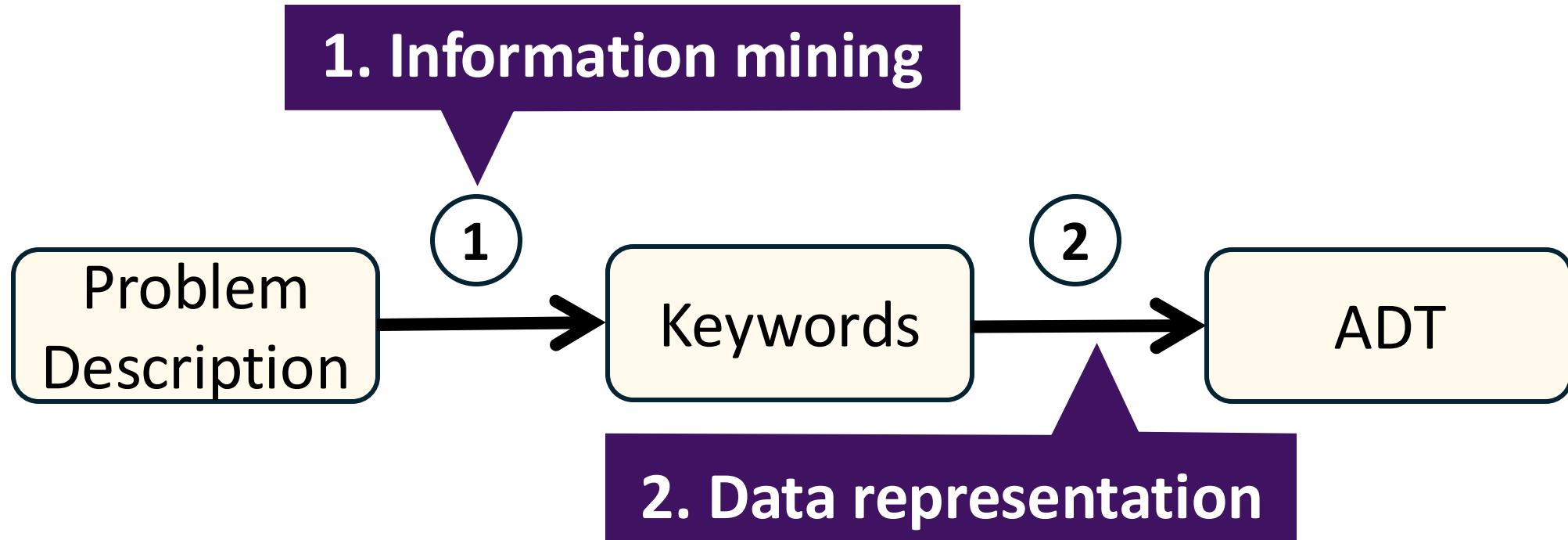


 **Daisy:**
a block-based environment for solving data modeling exercise

Refining Data Modeling Process

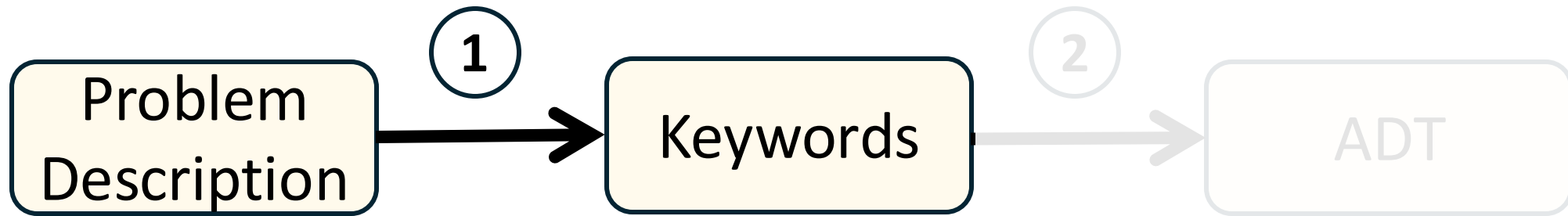


Refining Data Modeling Process

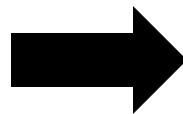


Refining Data Modeling Process

1. Information Mining

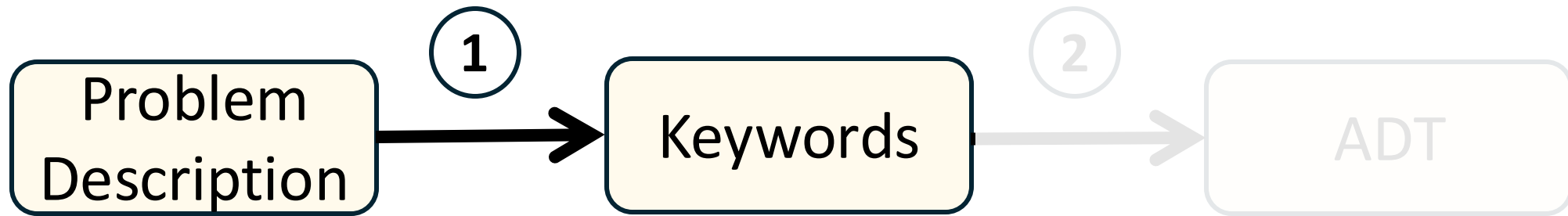


Define the function area to calculate the area of a shape. The shape is either a square or a triangle.

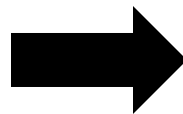


Refining Data Modeling Process

1. Information Mining

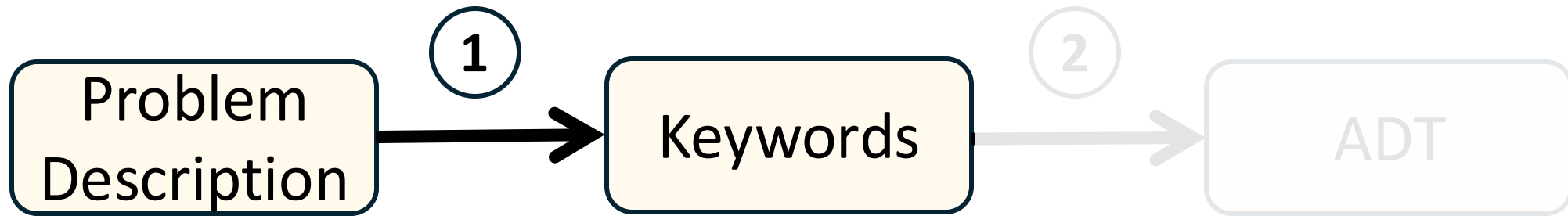


Define the function area to calculate the area of a **shape**.
The shape is either a square or a triangle.

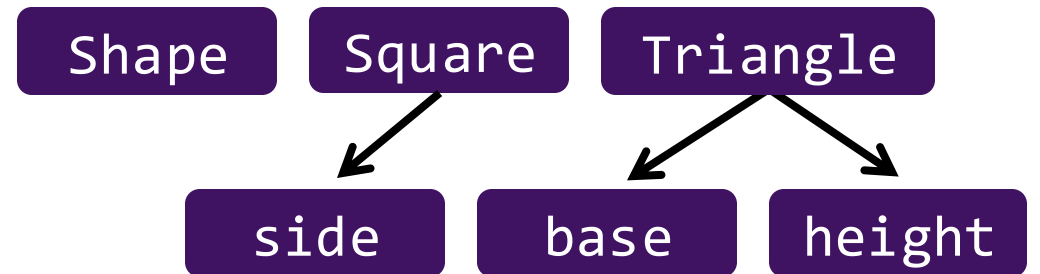


Refining Data Modeling Process

1. Information Mining

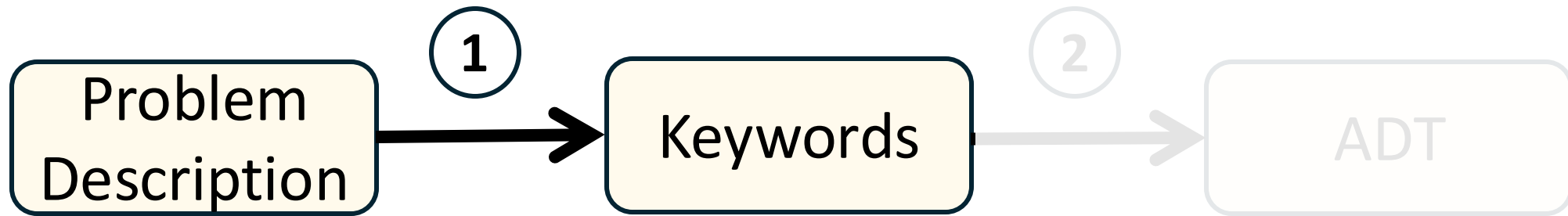


Define the function area to calculate the area of a **shape**.
The shape is either a square or a triangle.

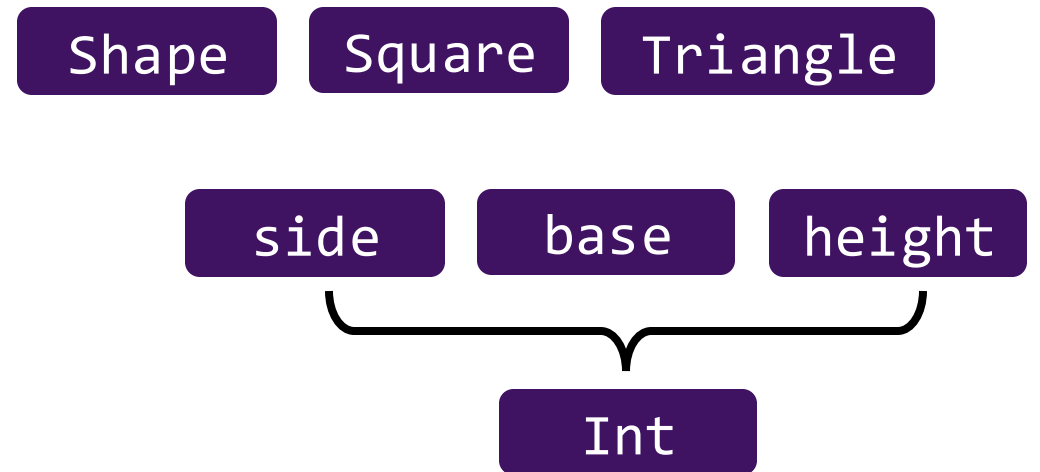
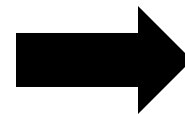


Refining Data Modeling Process

1. Information Mining

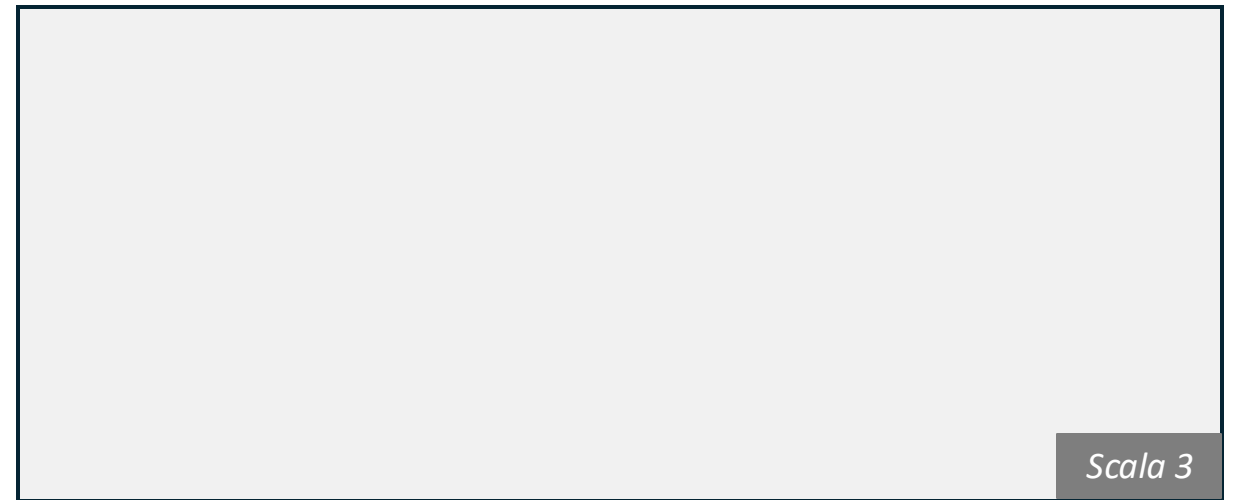
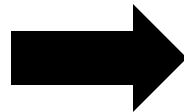
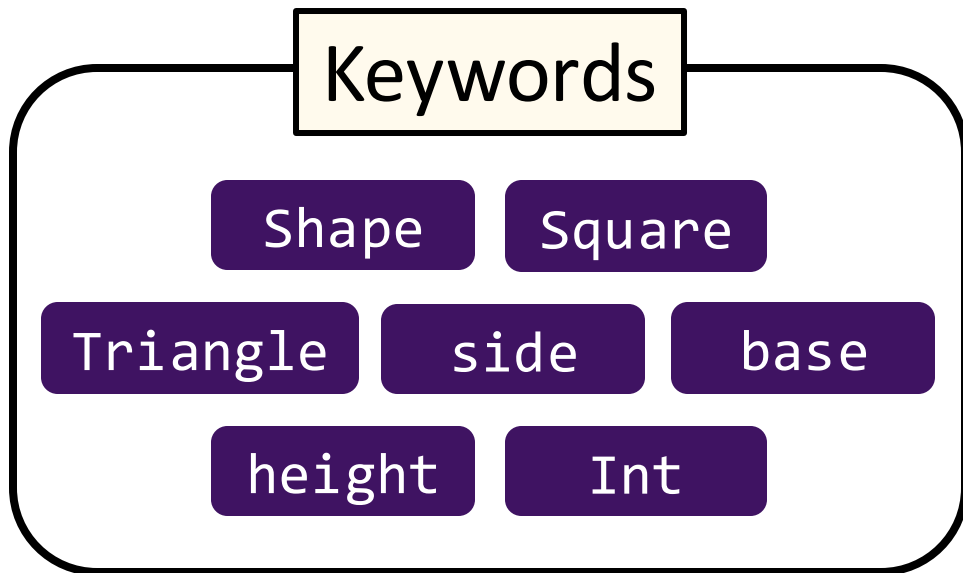
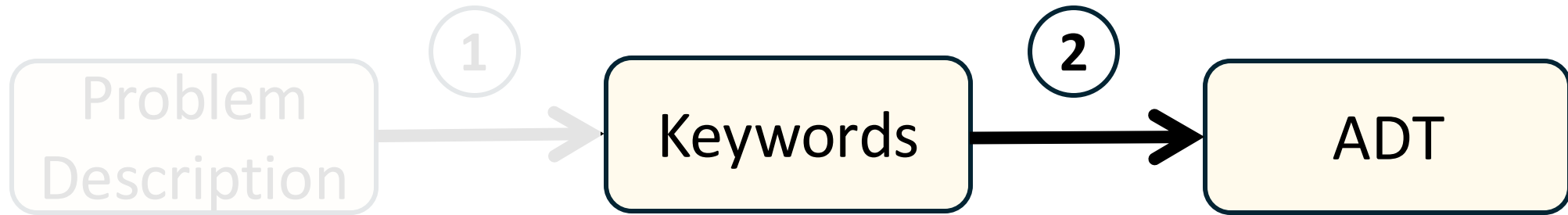


Define the function `area` to calculate the area of a **shape**.
The shape is either a **square** or a **triangle**.



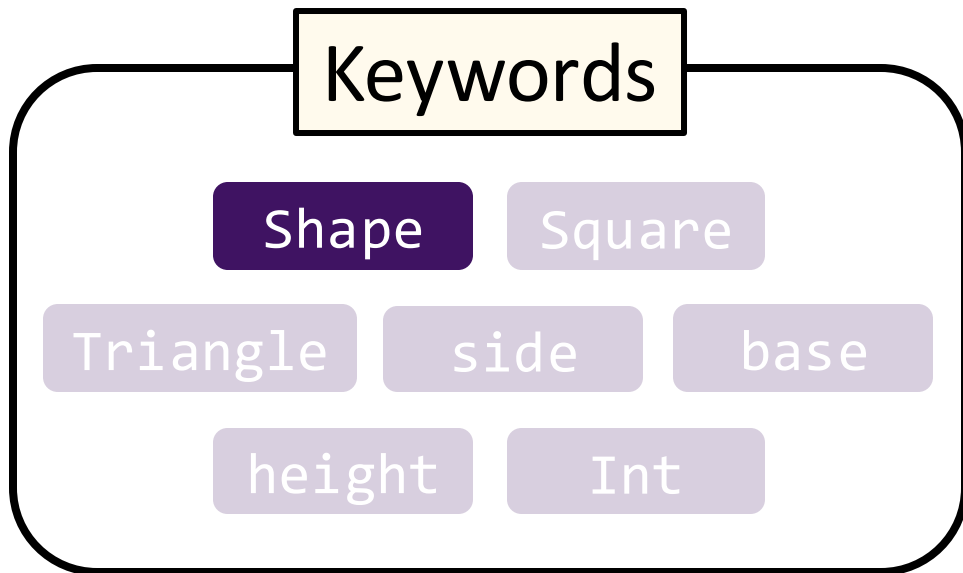
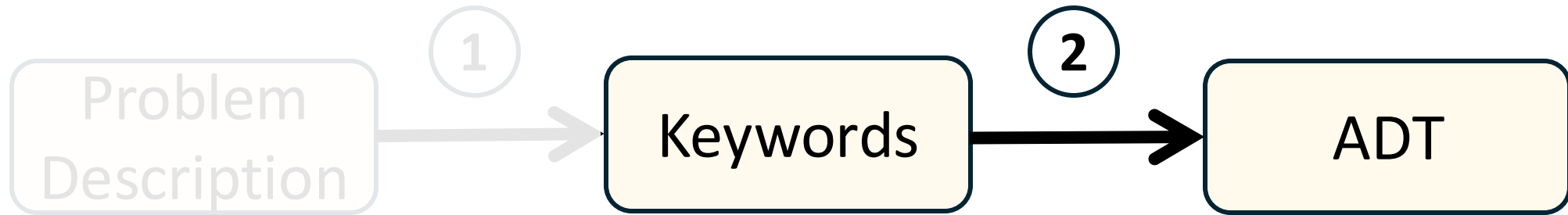
Refining Data Modeling Process

2. Data Representation



Refining Data Modeling Process

2. Data Representation

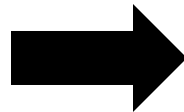
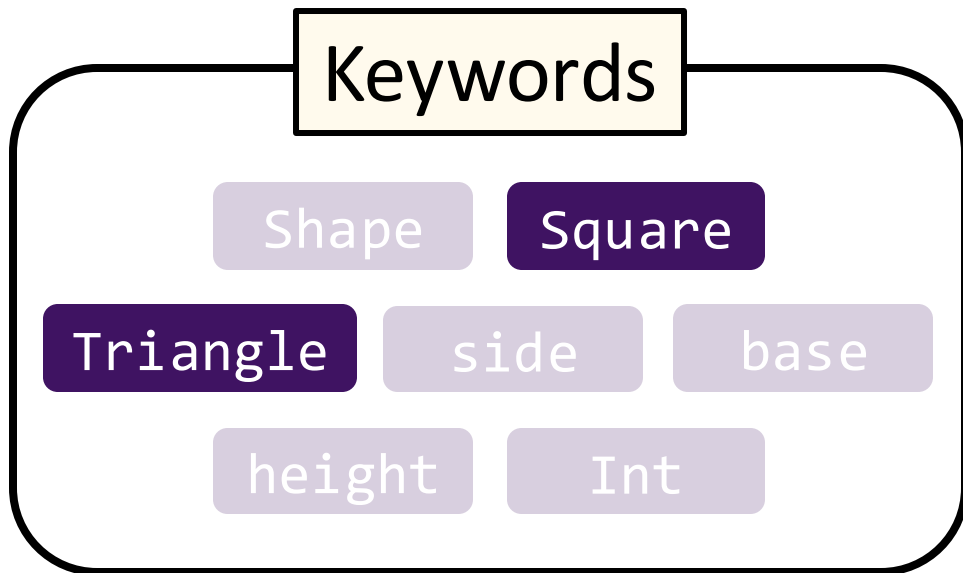
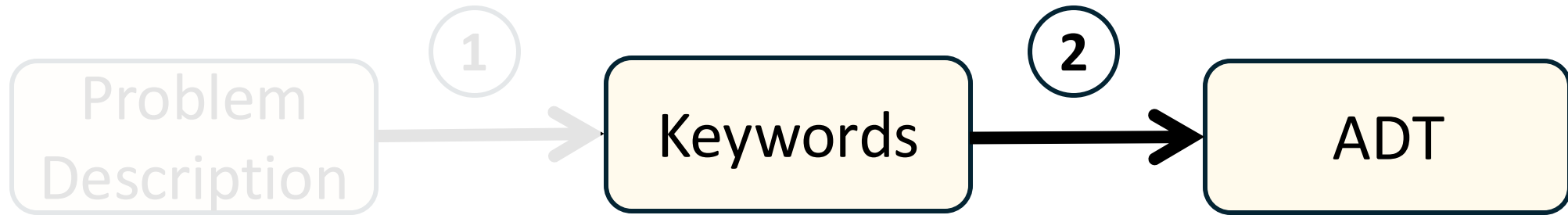


```
enum Shape :
```

Scala 3

Refining Data Modeling Process

2. Data Representation

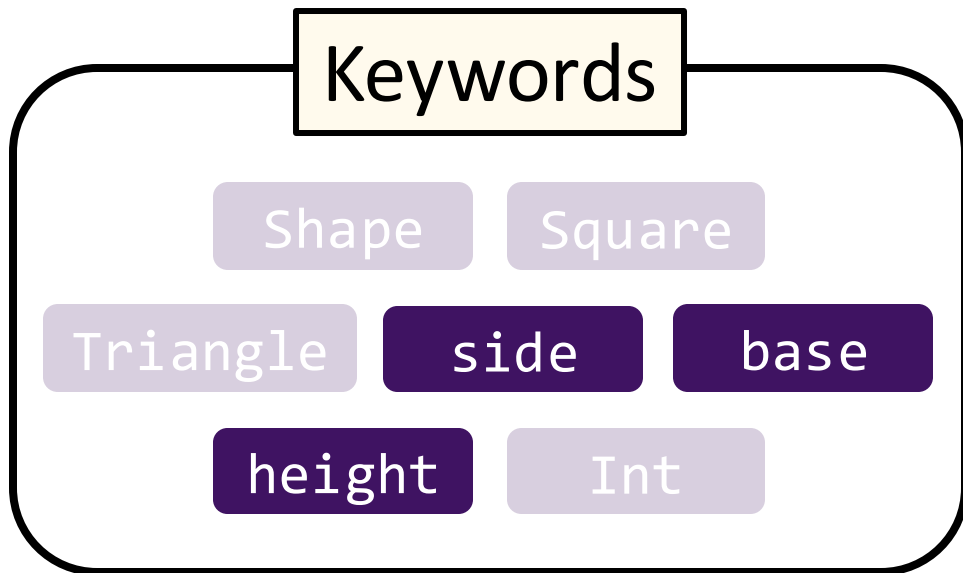
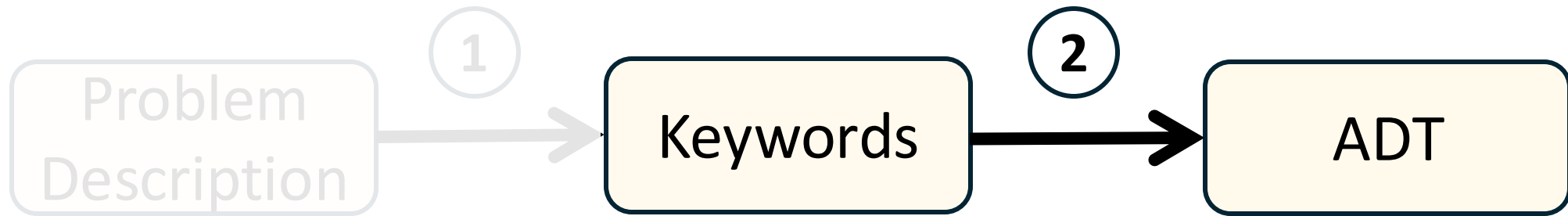


```
enum Shape :  
  case Square  
  case Triangle
```

Scala 3

Refining Data Modeling Process

2. Data Representation

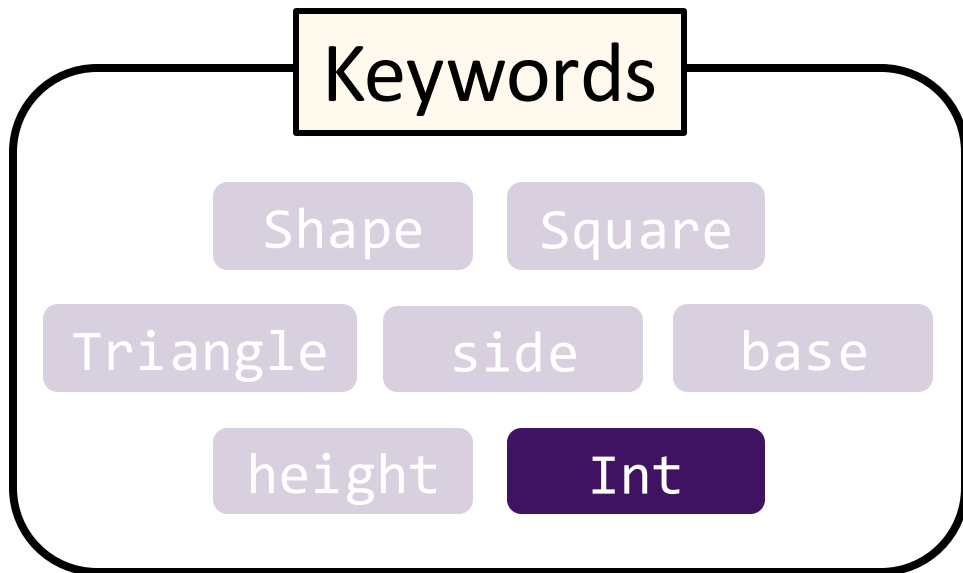
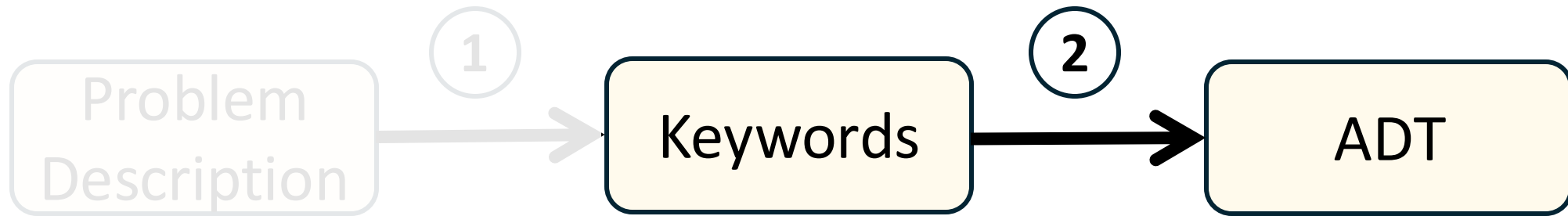


```
enum Shape :  
  case Square ( side )  
  case Triangle ( base , height )
```

Scala 3

Refining Data Modeling Process

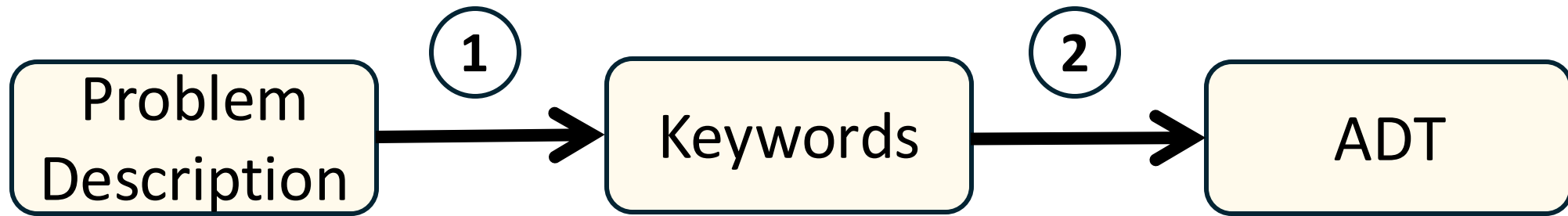
2. Data Representation



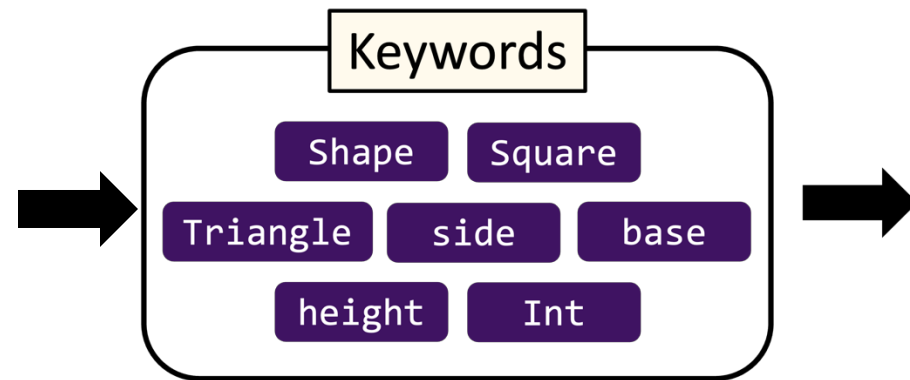
```
enum Shape :  
  case Square ( side : Int )  
  case Triangle ( base : Int ,  
                 height : Int )
```

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.

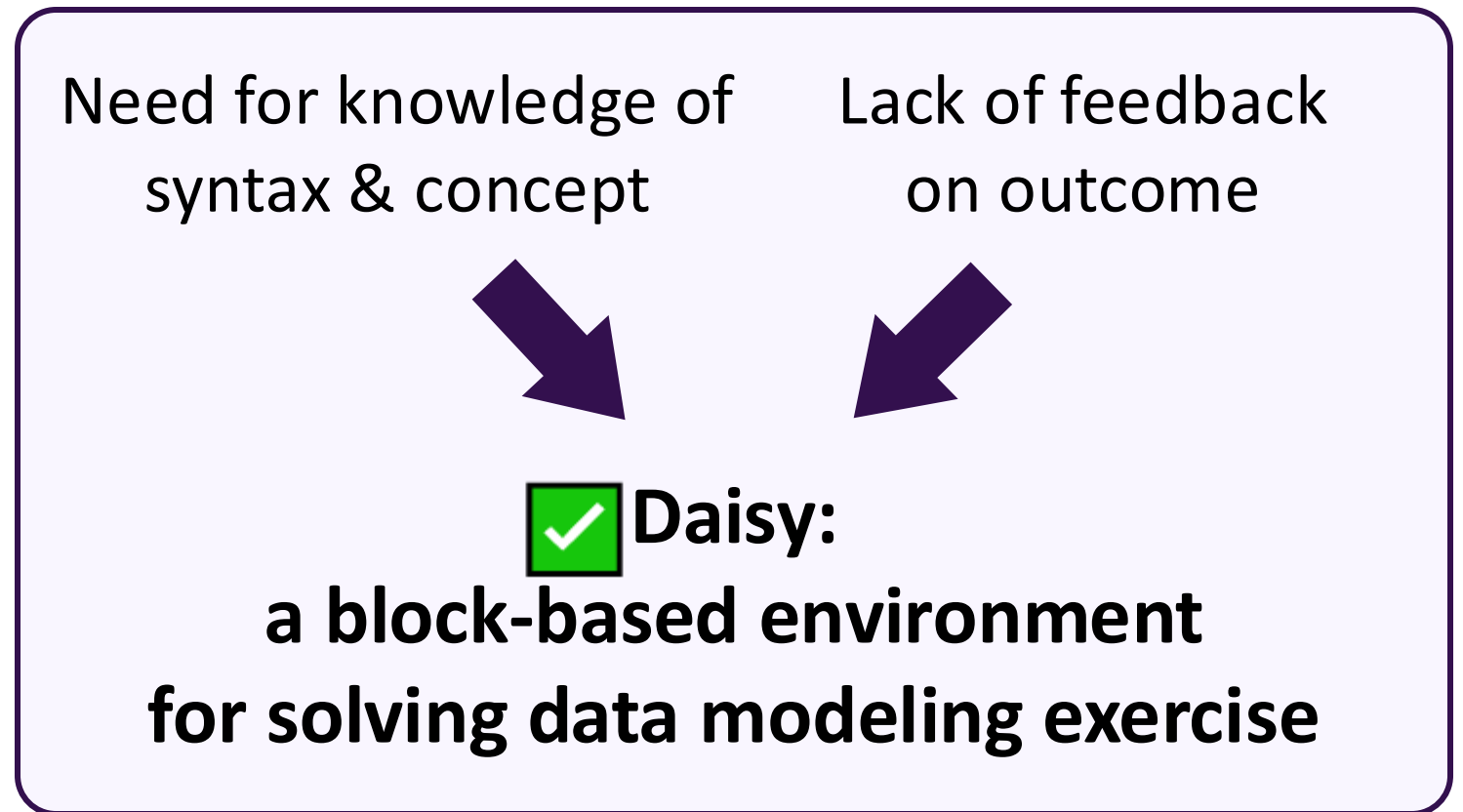
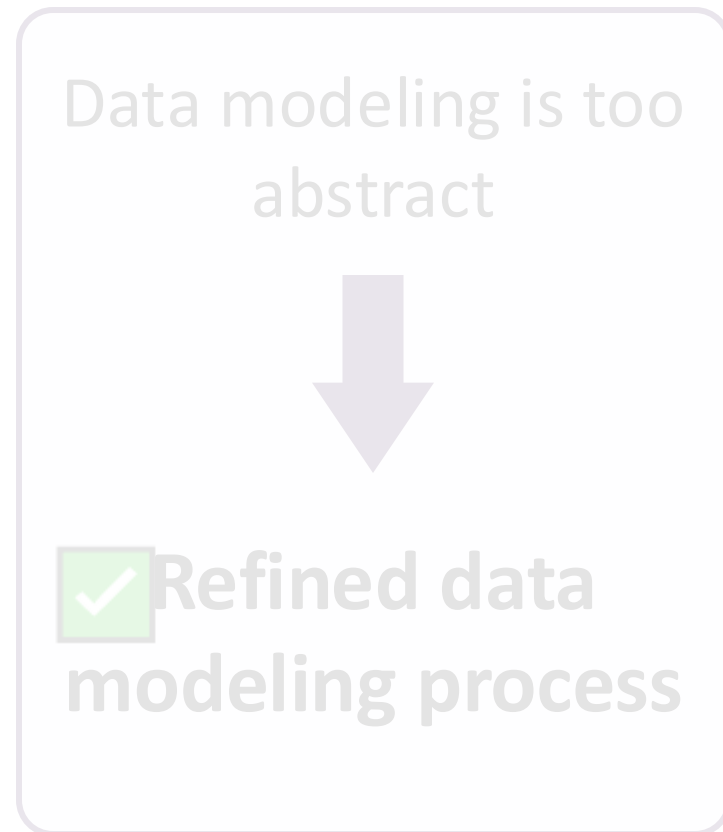


```
enum Shape :  
  case Square ( side : Int )  
  case Triangle ( base : Int ,  
                 height : Int )
```

Scala 3

Goal: Assist Data Modeling Learning

Contributions:



The Daisy Environment

(data modeling is easy to learn)

Blocks

has following cases ▶ +

2 Constructor (with argument)

has ▶

2 Constructor (without argument)

3 Argument

of type

4 Keyword

String

Boolean

Int

Triangle

height

Shape

Square

base

side

Make a block

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

Feedback Generator

Shape has following cases

Square has side of type Int

Triangle has base of type Int height of type Int

Feedback

Congratulations!
You've
constructed a
correct data
definition 🎉

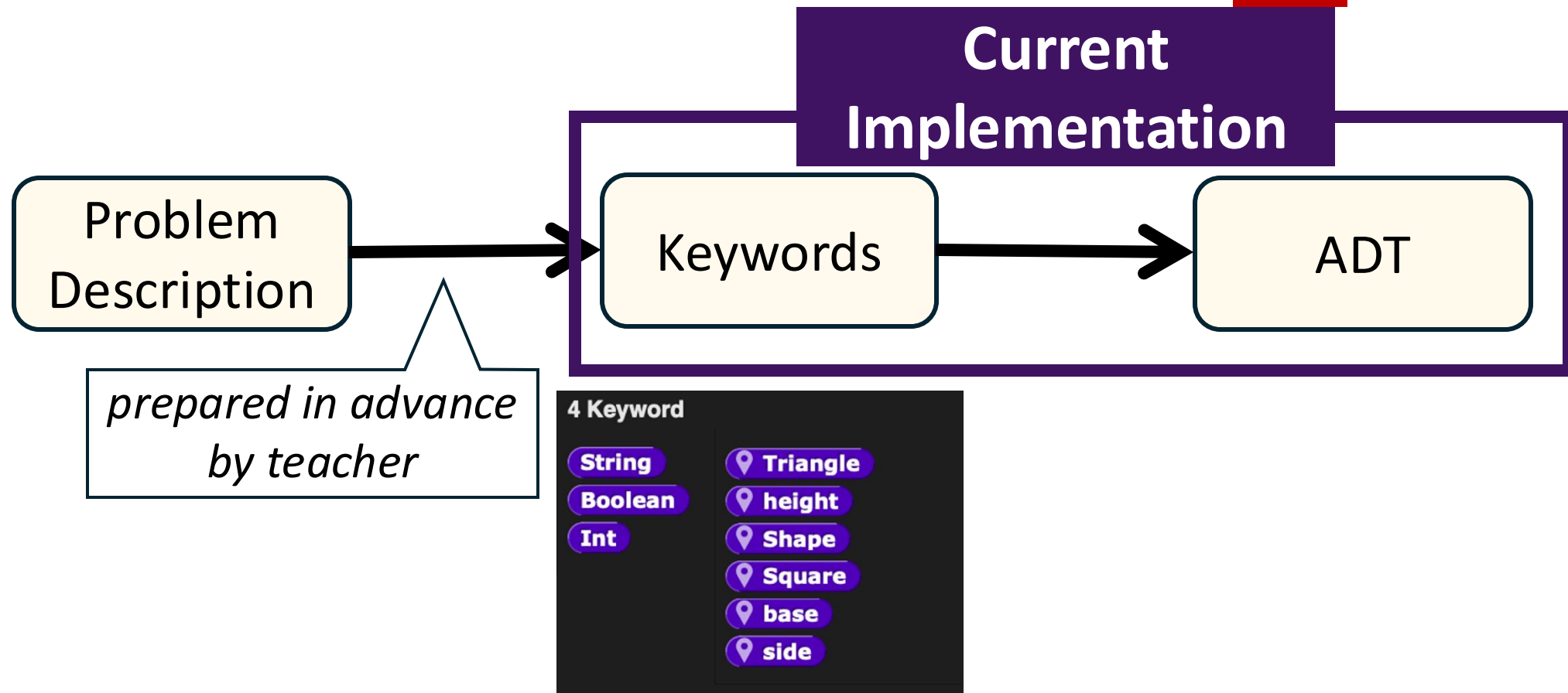
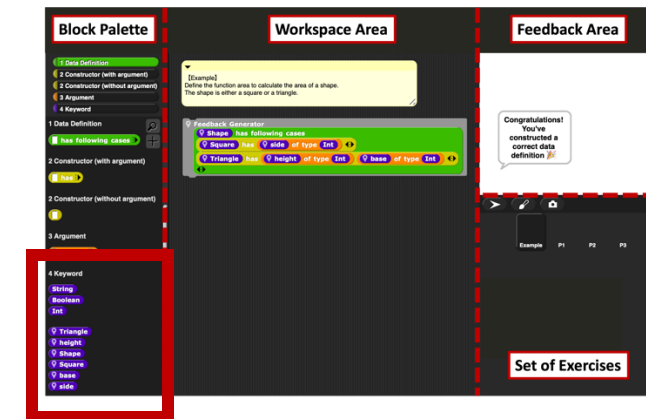
Example E1 E2 E3

Set of Exercises

Features of Daisy Environment

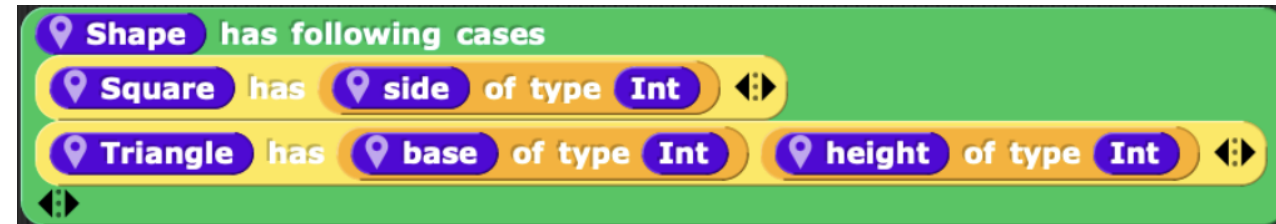
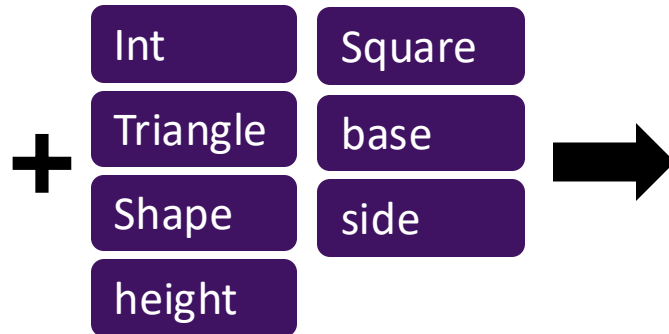
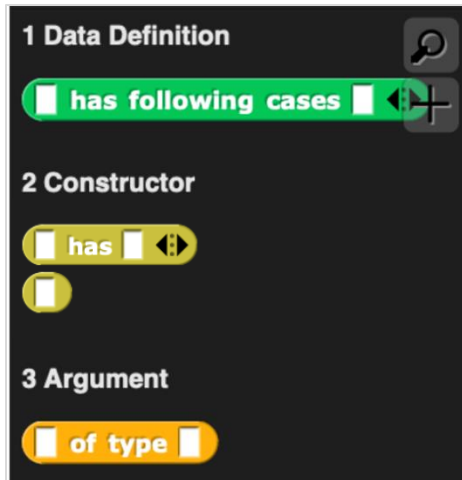
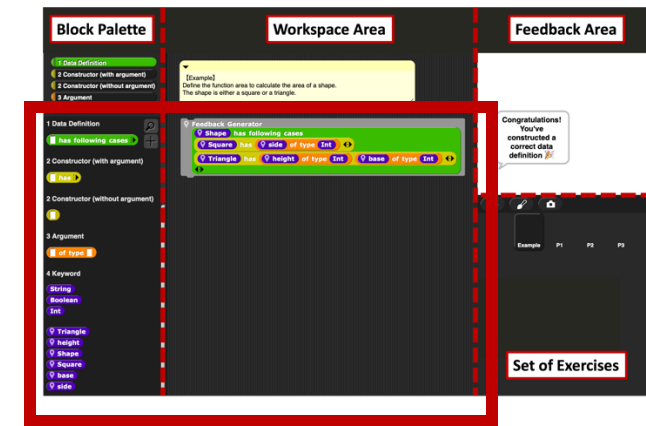
- ✓ Support for data modeling
- ✓ Use of natural language
- ✓ Automatic feedback generation

Features of Daisy Support for Data Modeling

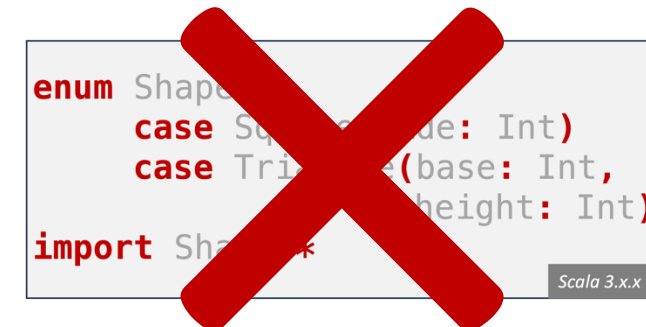


Features of Daisy

Use of Natural Language

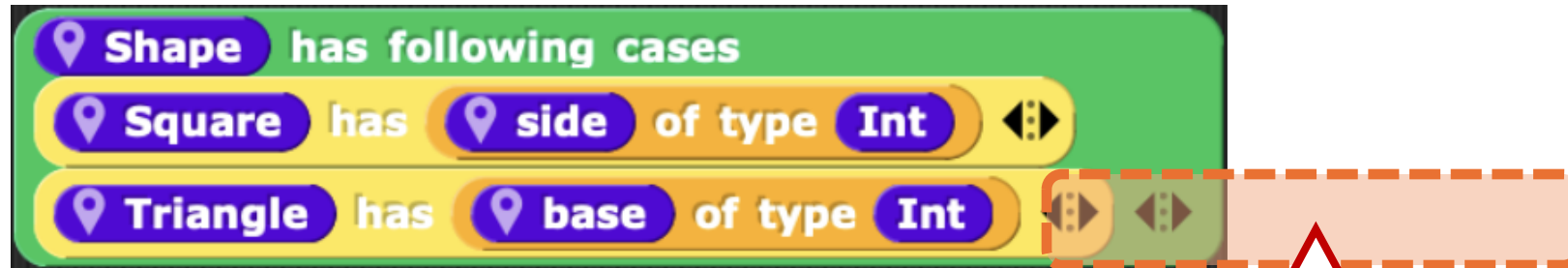
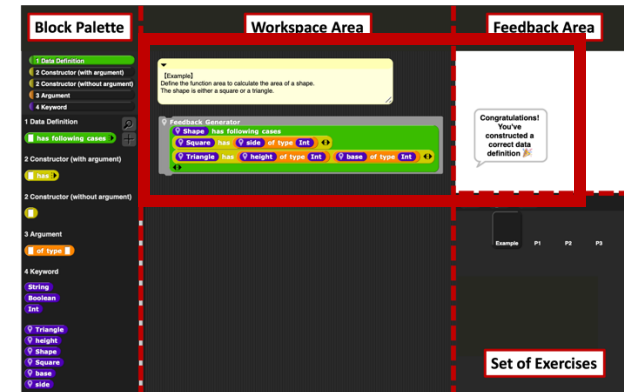


✓ Focus on the essential task of constructing ADT



Features of Daisy Feedback Generation

Comparison with the correct answer
provided by teacher



Warning:
Incorrect **argument number** of Triangle

The Daisy Environment: Demo

The image shows a software interface for 'The Daisy Environment'. On the left is a dark sidebar with several categories and items:

- 1 Data Definition**: Includes a search icon, a green button labeled 'has following cases', and a plus icon.
- 2 Constructor (with argument)**: Includes a yellow button labeled 'has'.
- 2 Constructor (without argument)**: Includes a yellow button.
- 3 Argument**: Includes an orange button labeled 'of type'.
- 4 Keyword**: Includes buttons for 'String', 'Boolean', and 'Int'. Below these are location pins for 'Triangle', 'height', 'Shape', 'Square', 'base', and 'side'. At the bottom is a 'Make a block' button.

The main workspace is dark grey and contains:

- A yellow box with a dropdown arrow, containing the text: **[Example]**
Define the function area to calculate the area of a shape.
The shape is either a square or a triangle.
- A grey button labeled 'Feedback Generator' with a location pin icon.

At the bottom right, there is a control panel with a navigation arrow, a key icon, and a camera icon. Below these are four buttons: 'Example', 'E1', 'E2', and 'E3'. A white square labeled 'Stage' is positioned below the 'Example' button.

Preliminary Experiment

Participants

27 undergraduate students with basic knowledge of FP

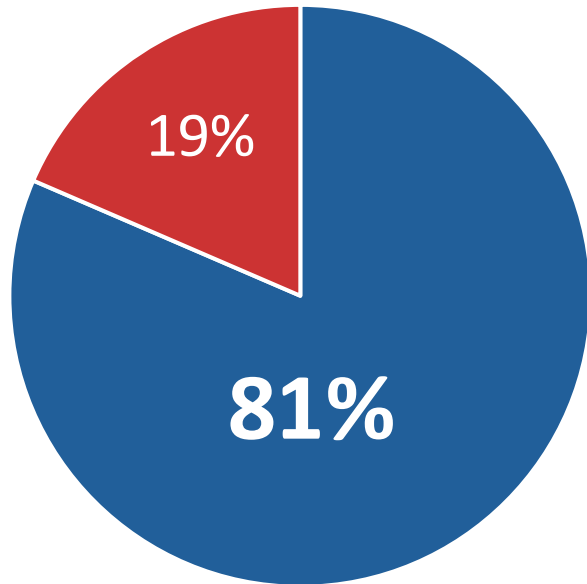
Tasks

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

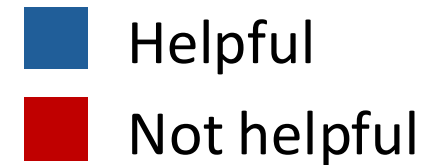
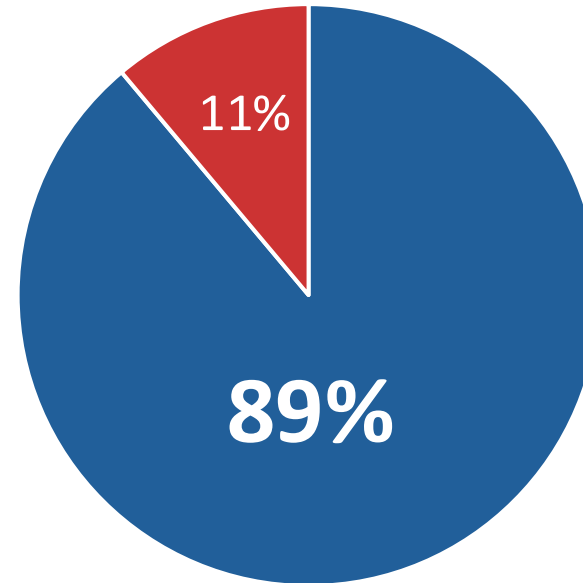
Results

1. Helpfulness of Daisy

Block Components



Automatic Feedback

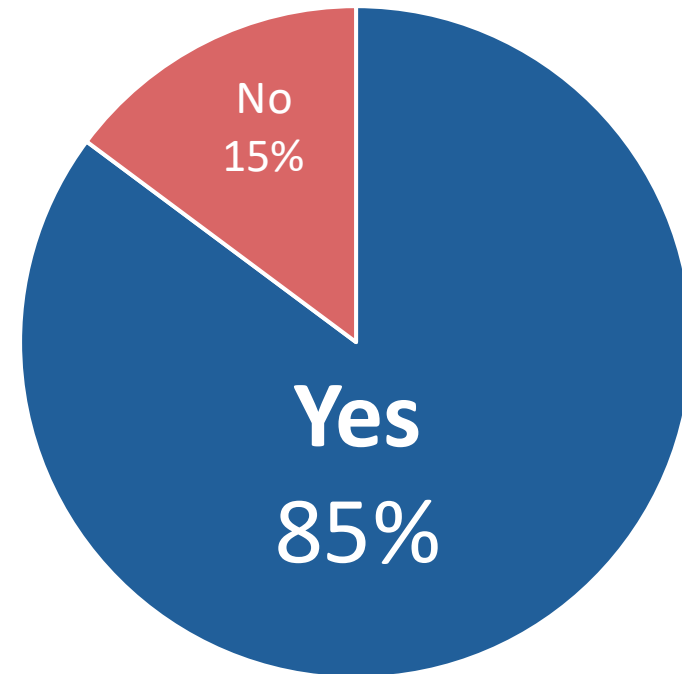


 **> 80%** students said Daisy is helpful

Results

2. Block Components

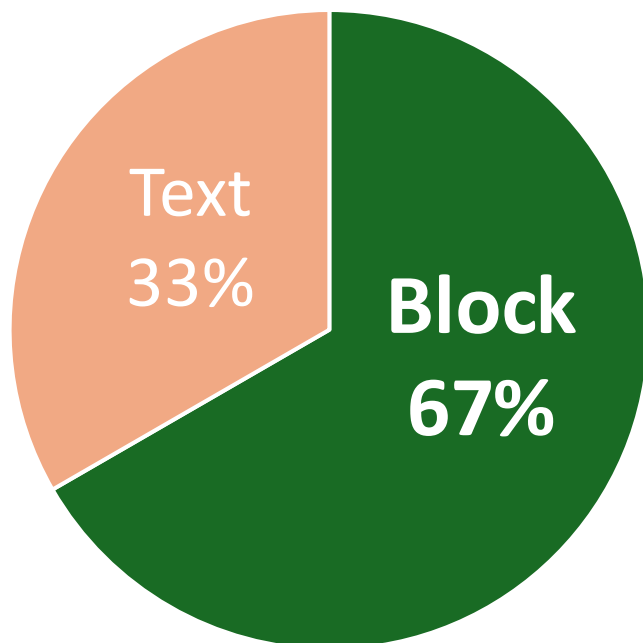
When constructing ADT,
**provision of constructor and
argument names makes
tasks easier**



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



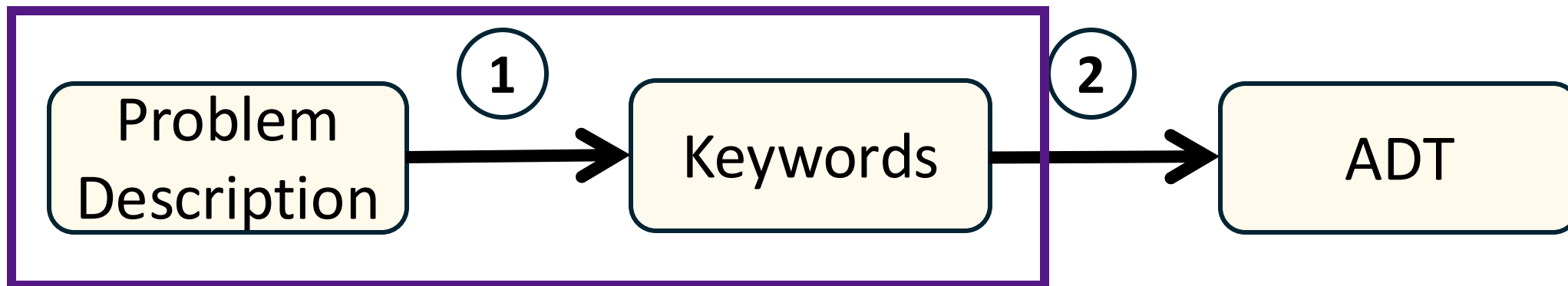
Better to *keep previous messages* when generating new feedback

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

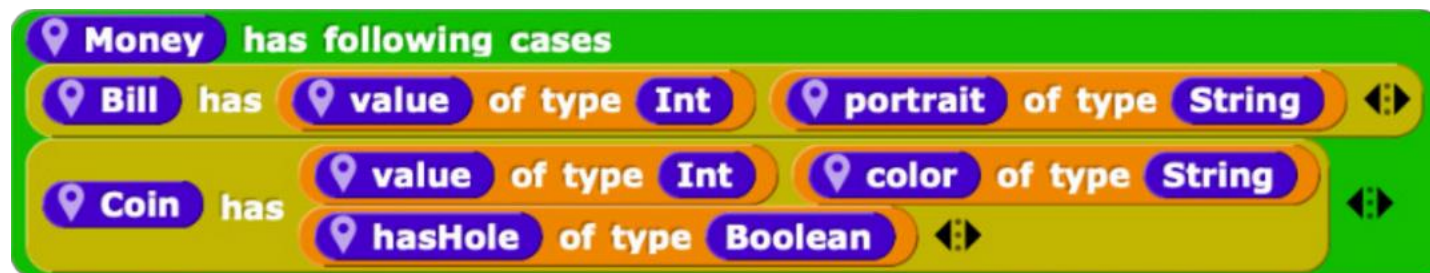
Define the data type Money that represents cash used in Japan.

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

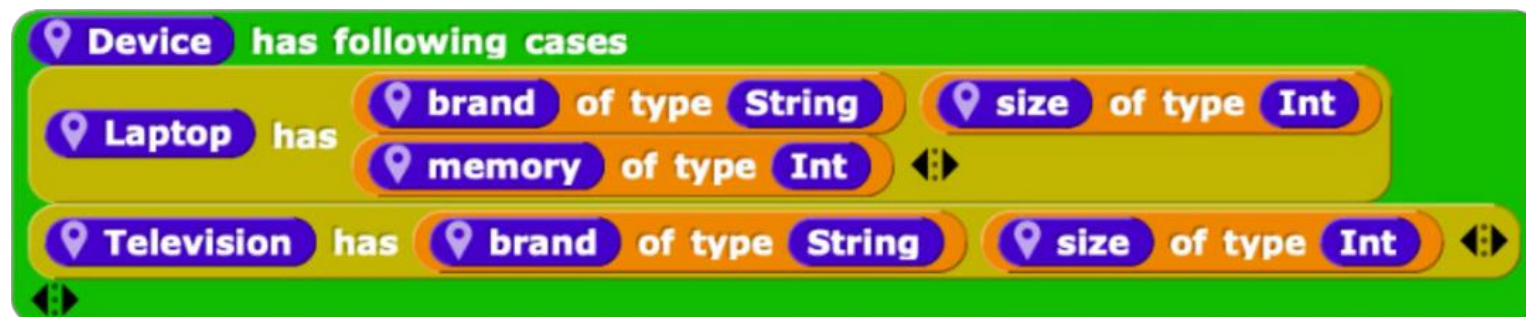
Define the data type Device that represents devices.

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

```
Path has following cases
File has
  name of type String
  extension of type String
Directory has
  name of type String
  content of type Path
```