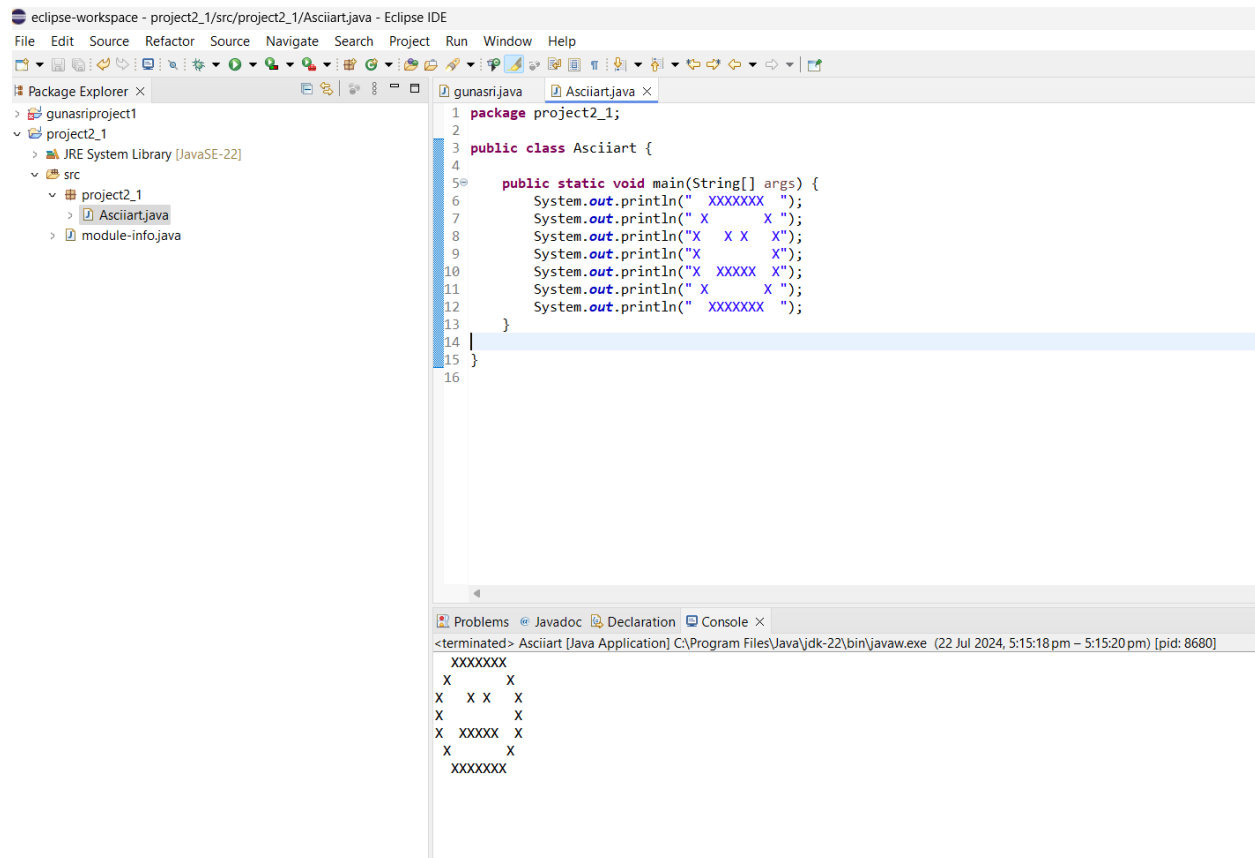


PROJECT 2:

PROJECT 2.1



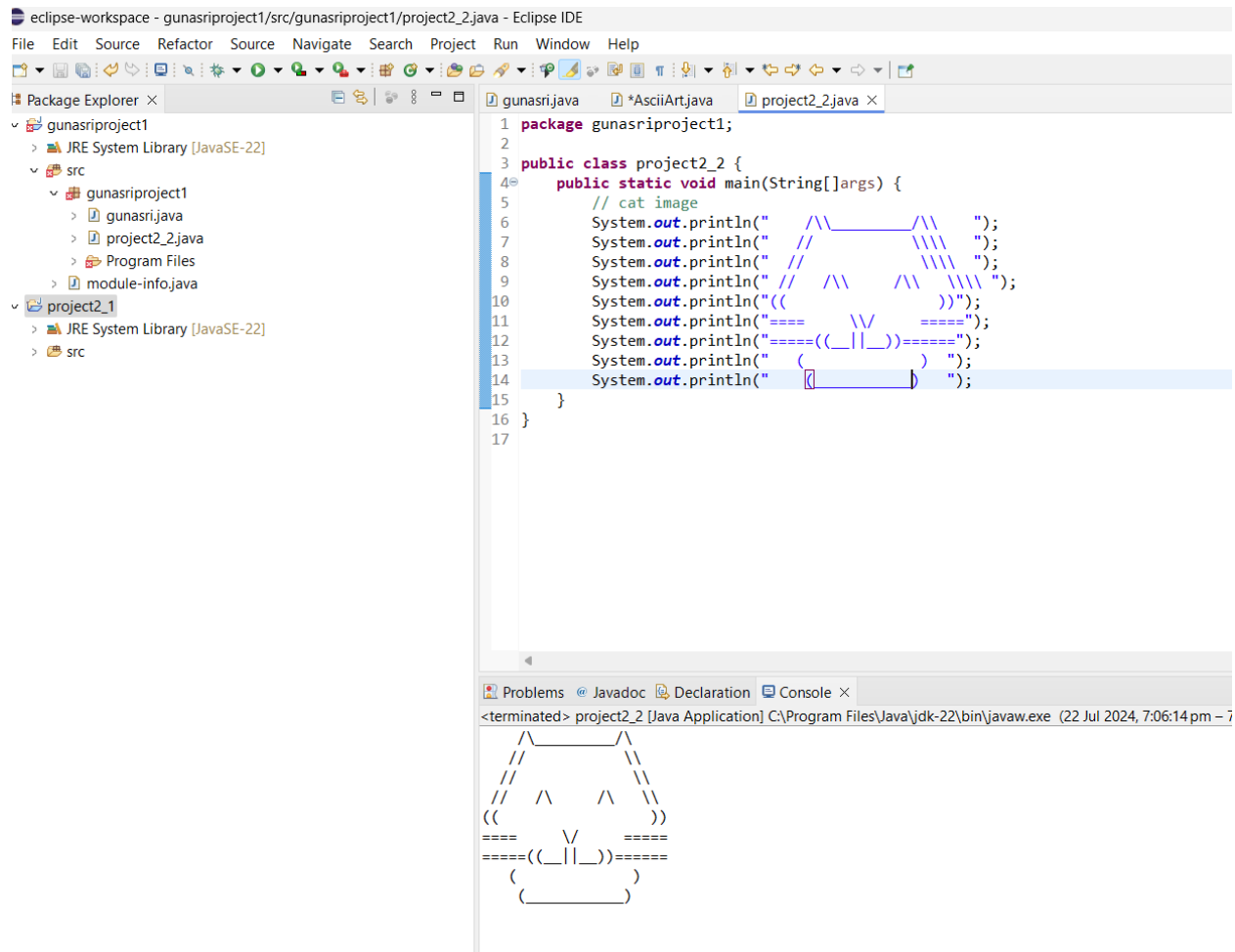
The screenshot shows the Eclipse IDE interface. The Package Explorer on the left displays the project structure: gunasriproject1 > project2_1 > src > project2_1 > Asciiart.java. The main editor window shows the code for Asciiart.java, which defines a package project2_1 and a public class Asciiart with a main method. The console at the bottom shows the output of the program, which is a 7x7 ASCII art pattern of 'X' characters.

```
1 package project2_1;
2
3 public class Asciiart {
4
5     public static void main(String[] args) {
6         System.out.println(" XXXXXX ");
7         System.out.println(" X   X ");
8         System.out.println("X  X X  X");
9         System.out.println("X       X");
10        System.out.println("X XXXXX X");
11        System.out.println(" X   X ");
12        System.out.println(" XXXXXX ");
13    }
14 }
15
16
```

<terminated> Asciiart [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (22 Jul 2024, 5:15:18 pm – 5:15:20 pm) [pid: 8680]

```
XXXXXX
X       X
X  X X  X
X       X
X XXXXX X
X       X
XXXXXX
```

PROJECT 2.2:



```
1 package gunasriproject1;
2
3 public class project2_2 {
4     public static void main(String[] args) {
5         // cat image
6         System.out.println("  /\_/\  ");
7         System.out.println(" //  \\// ");
8         System.out.println(" //  \\// ");
9         System.out.println(" //  /\_/\  ");
10        System.out.println("((  ))");
11        System.out.println("==== V ====");
12        System.out.println("====( _| _ )====");
13        System.out.println(" (      ) ");
14        System.out.println("  (____)  ");
15    }
16 }
17
```

<terminated> project2_2 [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (22 Jul 2024, 7:06:14 pm - 7

```
  /\_/\
 //  \\
 //  \\
 //  /\_/\
((  ))
==== V ====
====( _| _ )====
 (      )
  (____)
```

PROJECT 2.3

DOCUMENT TO CREATE SNAKE BOX FACTORY SOFTWARE DESIGN

1. Object: SnakeBox

Properties:

Dimensions: The size of the box, typically including length, width, and height.

MaterialQuality: The quality of the cardboard used to make the box.

SnakeType: The specific type of snake that will be placed inside the box.

Behaviors:

CalculateBoxVolume(): Computes the volume of the box based on its dimensions.

CheckMaterialQuality(): Assesses the quality of the cardboard to ensure it meets standards.

AssignSnakeType(SnakeType type): Assigns a specific snake type to the box and adjusts any related settings.

2. Object: Snake

Properties:

Species: The species of the snake.

Size: The size of the snake, which may influence the size of the box required.

HealthStatus: The current health status of the snake.

Behaviors:

ChangeHealthStatus(String status): Updates the health status of the snake.

Grow(Size newSize): Adjusts the size attribute of the snake as it grows.

GenerateReport(): Creates a report on the snake's current health and characteristics.

3. Object: Order

Properties:

OrderID: A unique identifier for each order.

CustomerDetails: Information about the customer placing the order, such as name and address.

OrderStatus: The current status of the order (e.g., Processing, Shipped, Delivered).

Behaviors:

UpdateOrderStatus(String newStatus): Changes the status of the order to reflect its current state.

GenerateInvoice(): Creates an invoice based on the details of the order.

TrackShipment(): Provides tracking information for the shipment of the order.