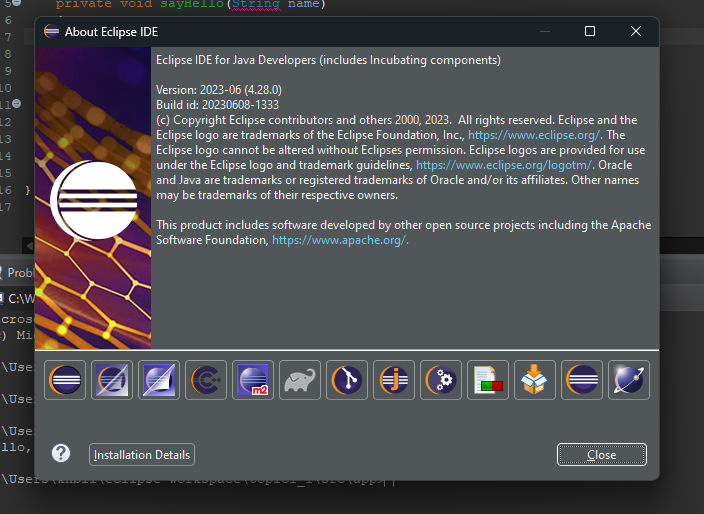
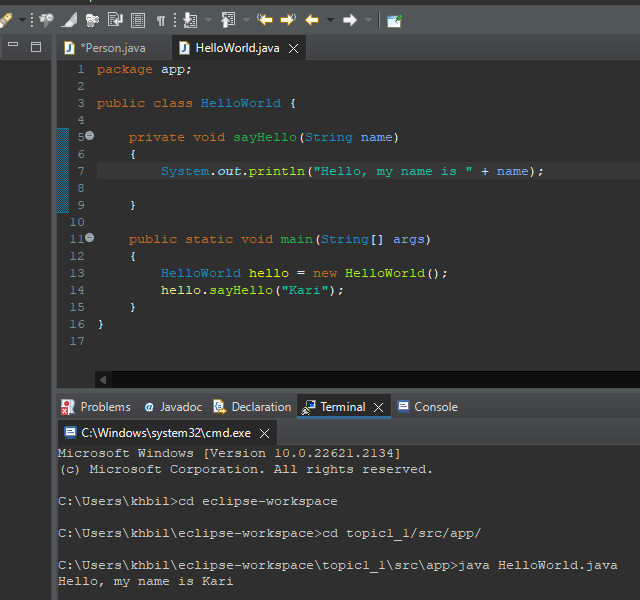
Topic 1

KariAnn Harjo

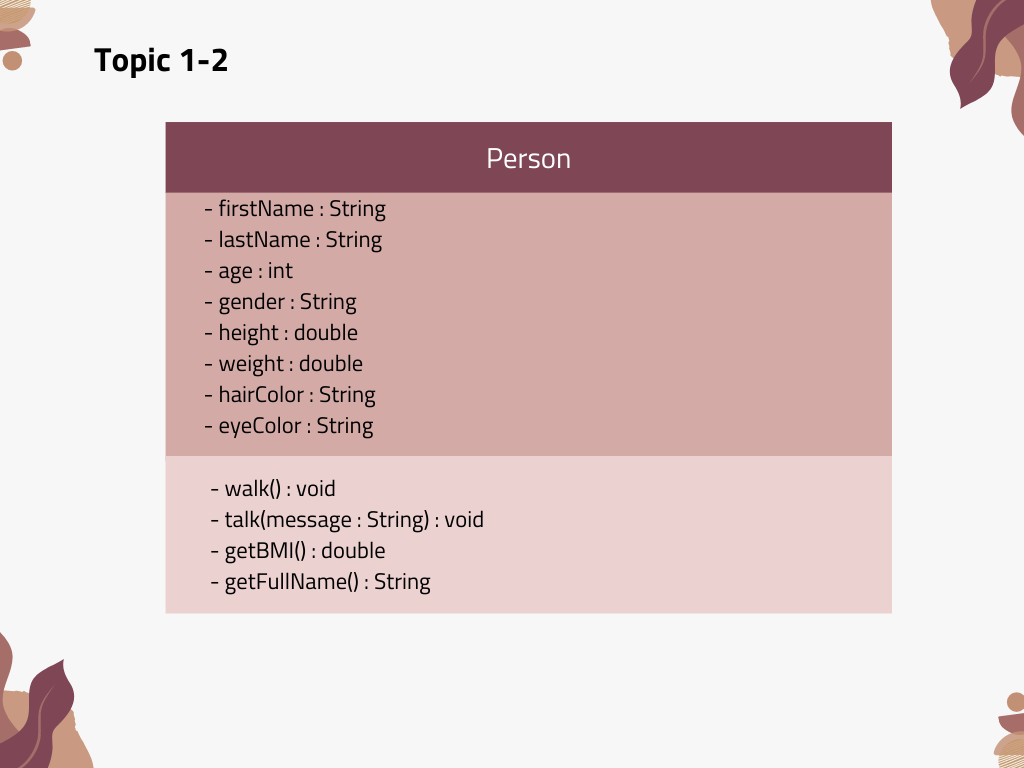
08/20/2023

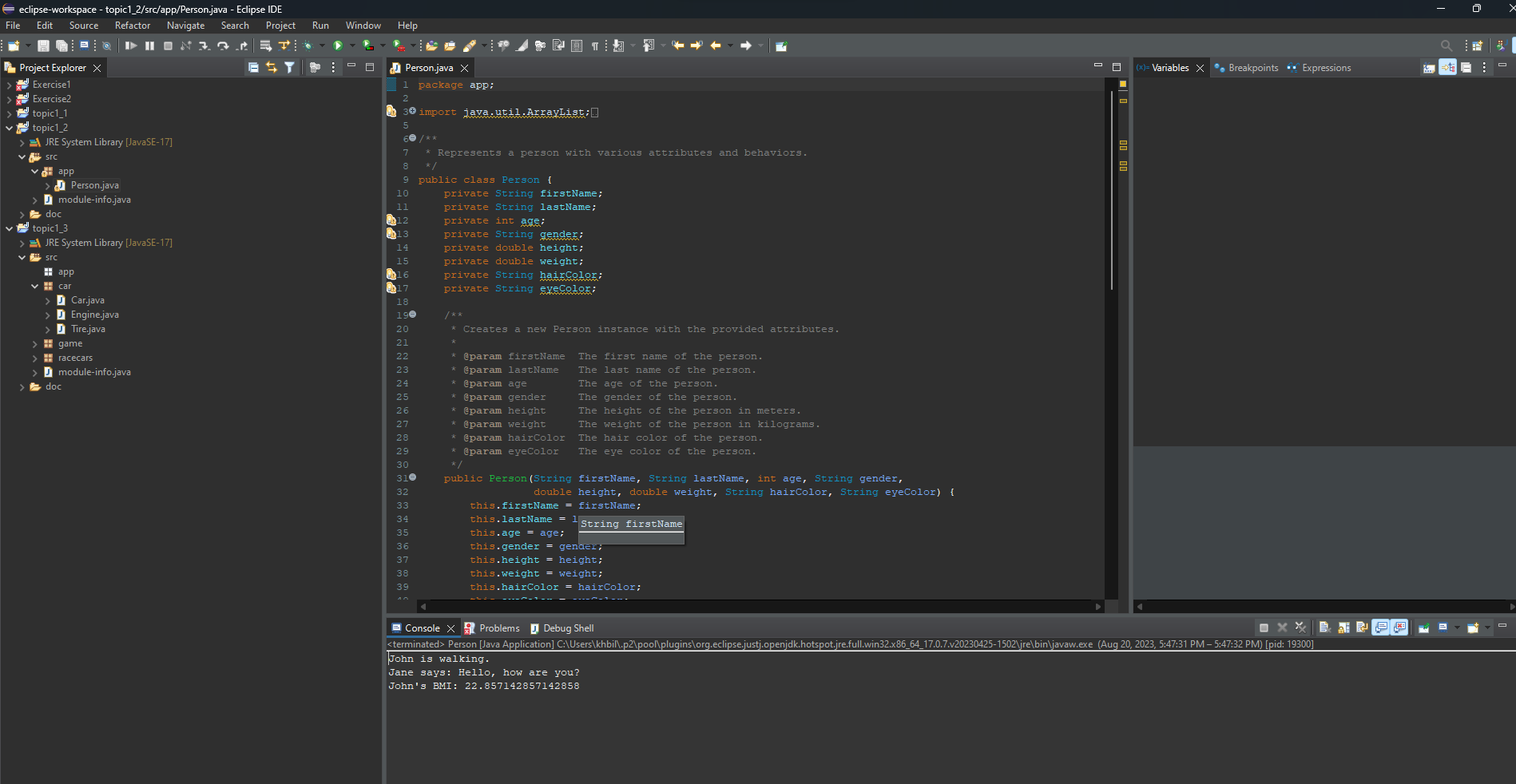
Topic 1\_1



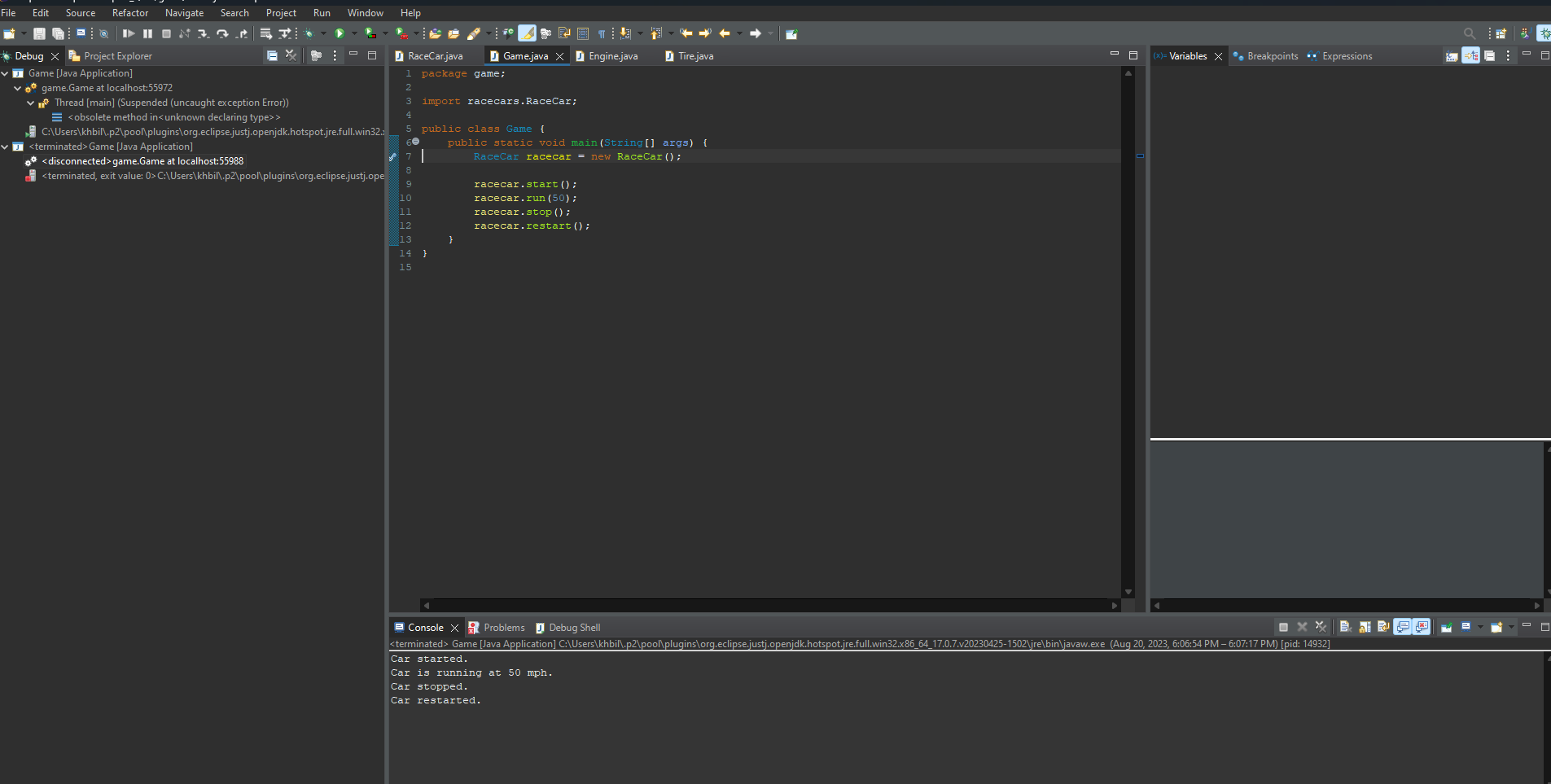


Topic 1\_2 – Person Class

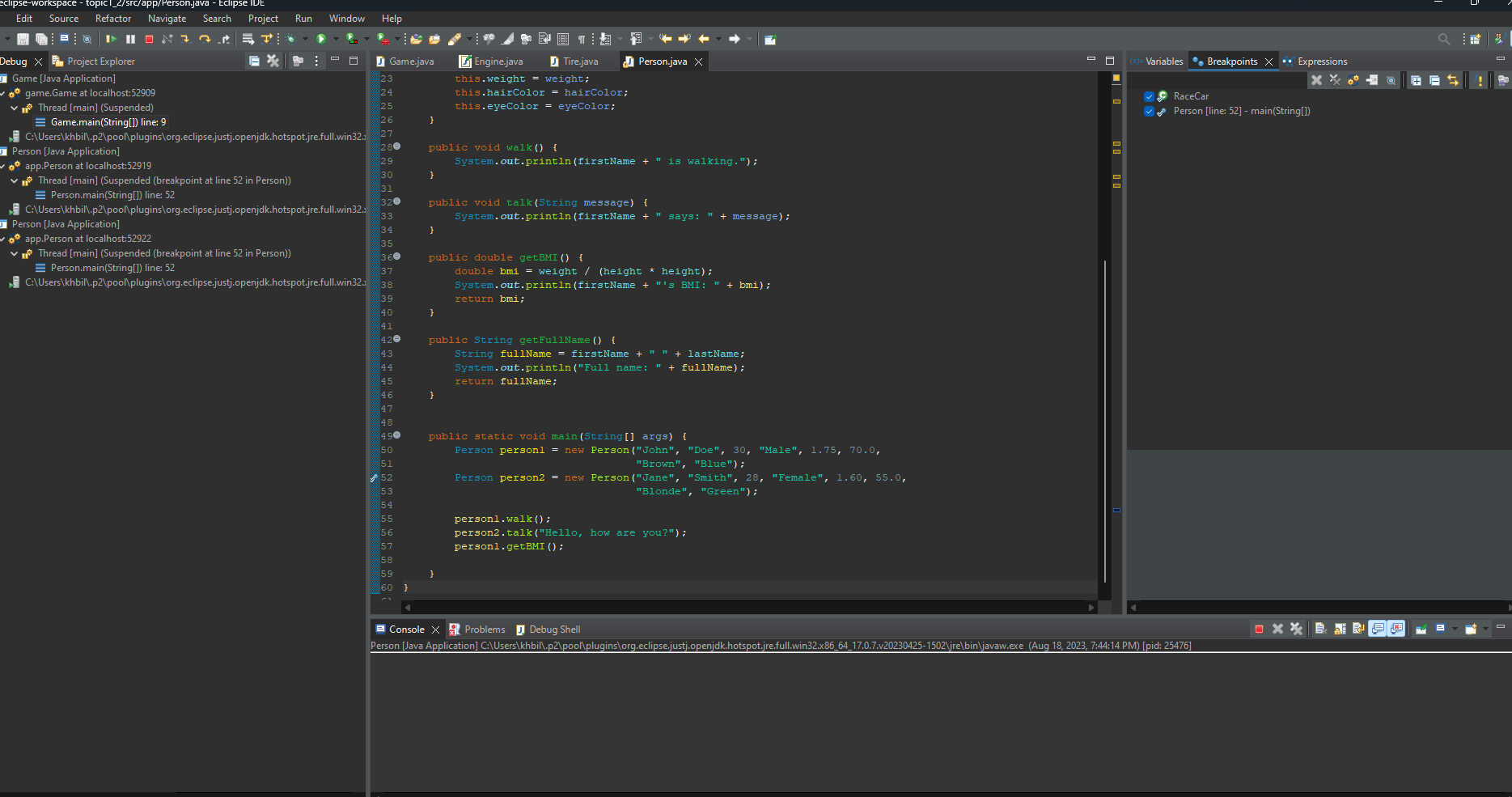


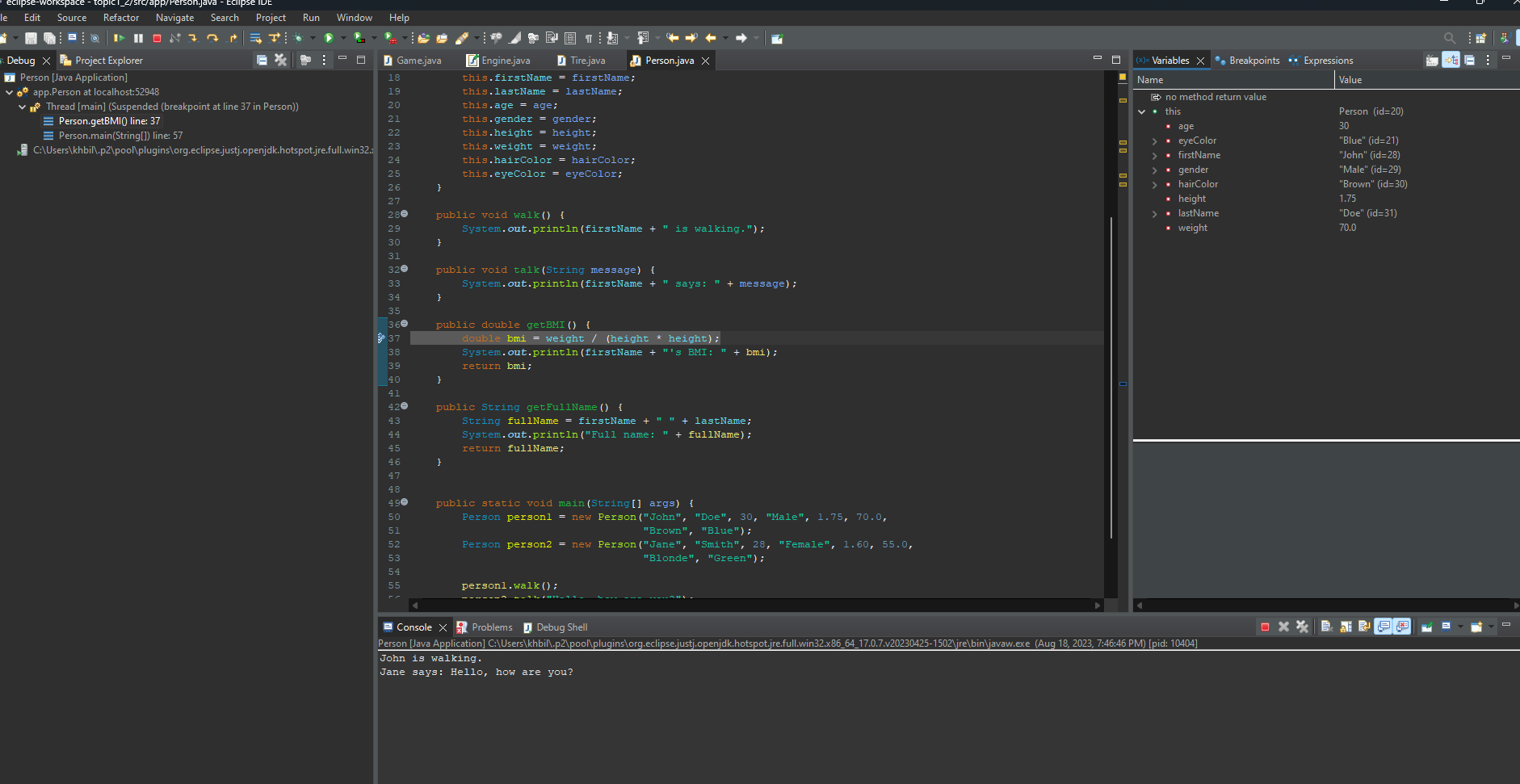


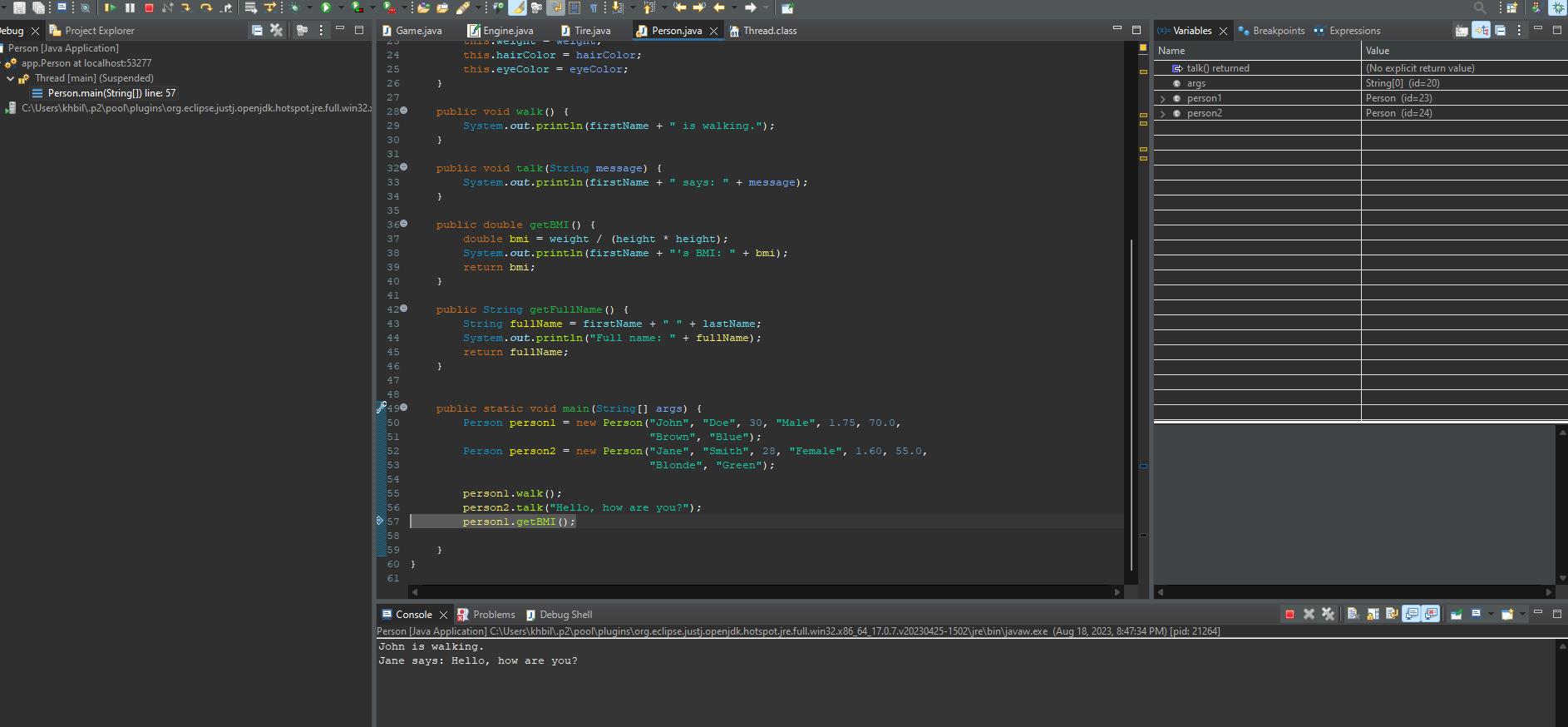
Topic 1\_3 – RaceCar

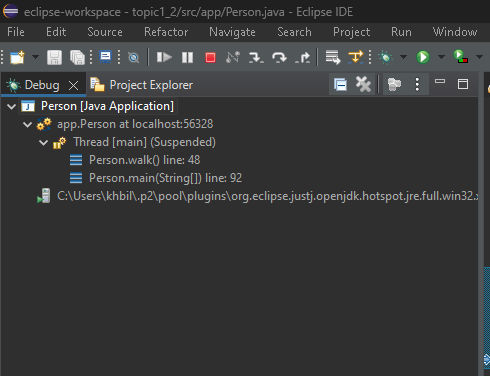


Topic 1\_4 – Debugging









Research Questions

1. In the context of the current class project, an association exists between the Salable class and the ShoppingCart class. This association signifies that a shopping cart contains salable products. The multiplicity of this association would be "0 to many," as a shopping cart can hold multiple salable products, but it's possible for a cart to be empty.

2. The statement "class abstraction is the separation of class implementation from the use of a class" refers to the practice of concealing the internal implementation details of a class while exposing its functionality to users. Users interact with the class through its public methods and attributes without needing to know how those methods are implemented. An example of this abstraction can be seen in a BankAccount class:

public class BankAccount {

private String accountNumber;

private double balance;

public BankAccount(String accountNumber) {

this.accountNumber = accountNumber;

this.balance = 0.0;

}

public void deposit(double amount) {

if (amount > 0) {

balance += amount;

}

}

public void withdraw(double amount) {

if (amount > 0 && balance >= amount) {

balance -= amount;

}

}

public double getBalance() {

return balance;

}

}