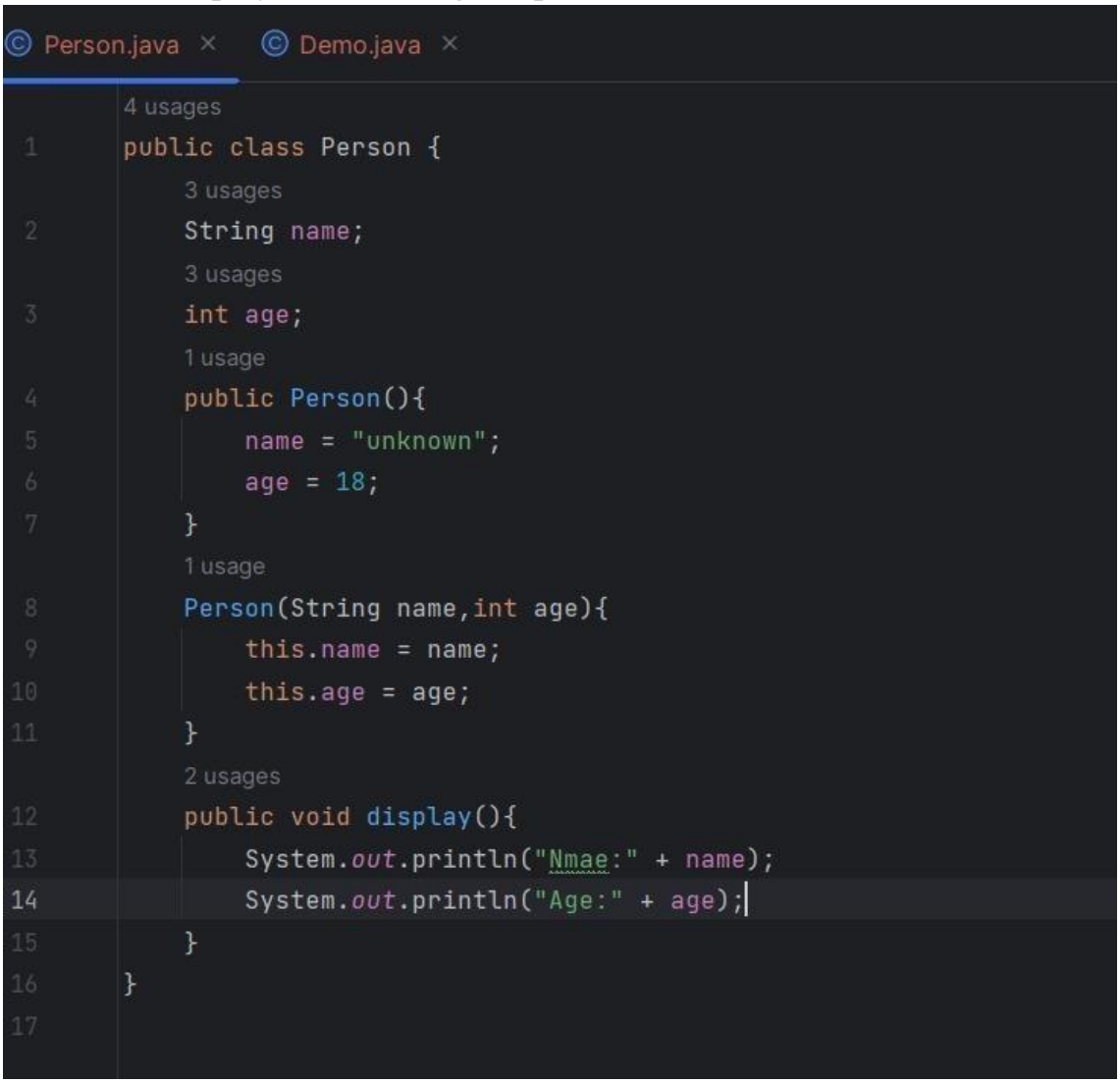
Task 2 Solutions

1.1) Create a class Person with properties (name and age) with following features.

a. Default age of person should be 18;

b. A person object can be initialized with name and age;

c. Method to display name and age of person



```java
public class Demo{
    public static void main(String[] args) {
        Person p1 = new Person();
        p1.display();
        Person p2 = new Person( name: "sam", age: 23);
        p2.display();
    }
}
```

The output:

1.2). Create class Product (pid, price, quantity) with parameterized constructor.

Create a main function in different class (say ProductMain) and perform following task:

a. Accept five product information from user and store in an array

b. Find Pid of the product with the highest price.

c. Create method (with array of product's object as argument) in ProductMain class to calculate and return the total amount spent on all products. (amount spent on single product price of product quantity of product

Product.java × ProductMain.java

```java
4 usages
class Product {
    3 usages
    int pid;
    5 usages
    double price;
    2 usages
    int quantity;
    1 usage
    Product(int pid, double price, int quantity) {
        this.pid = pid;
        this.price = price;
        this.quantity = quantity;
    }
}
```

```java
import java.util.Scanner;
public class ProductMain{
    1 usage
    public static double calculateTotalAmount(Product[] products) {
        double total = 0;
        for (int i = 0; i < products.length; i++) {
            total += products[i].price * products[i].quantity;
        }
        return total;
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Product[] products = new Product[5];
        for (int i = 0; i < 5; i++) {
            System.out.print("Product ID: ");
            int pid = sc.nextInt();
            System.out.print("Price: ");
            double price = sc.nextDouble();
            System.out.print("Quantity: ");
            int quantity = sc.nextInt();
            products[i] = new Product(pid, price, quantity);
        }
        double maxPrice = products[0].price;
        int maxPid = products[0].pid;

        for (int i = 1; i < products.length; i++) {
            if (products[i].price > maxPrice) {
                maxPrice = products[i].price;
                maxPid = products[i].pid;
            }
```

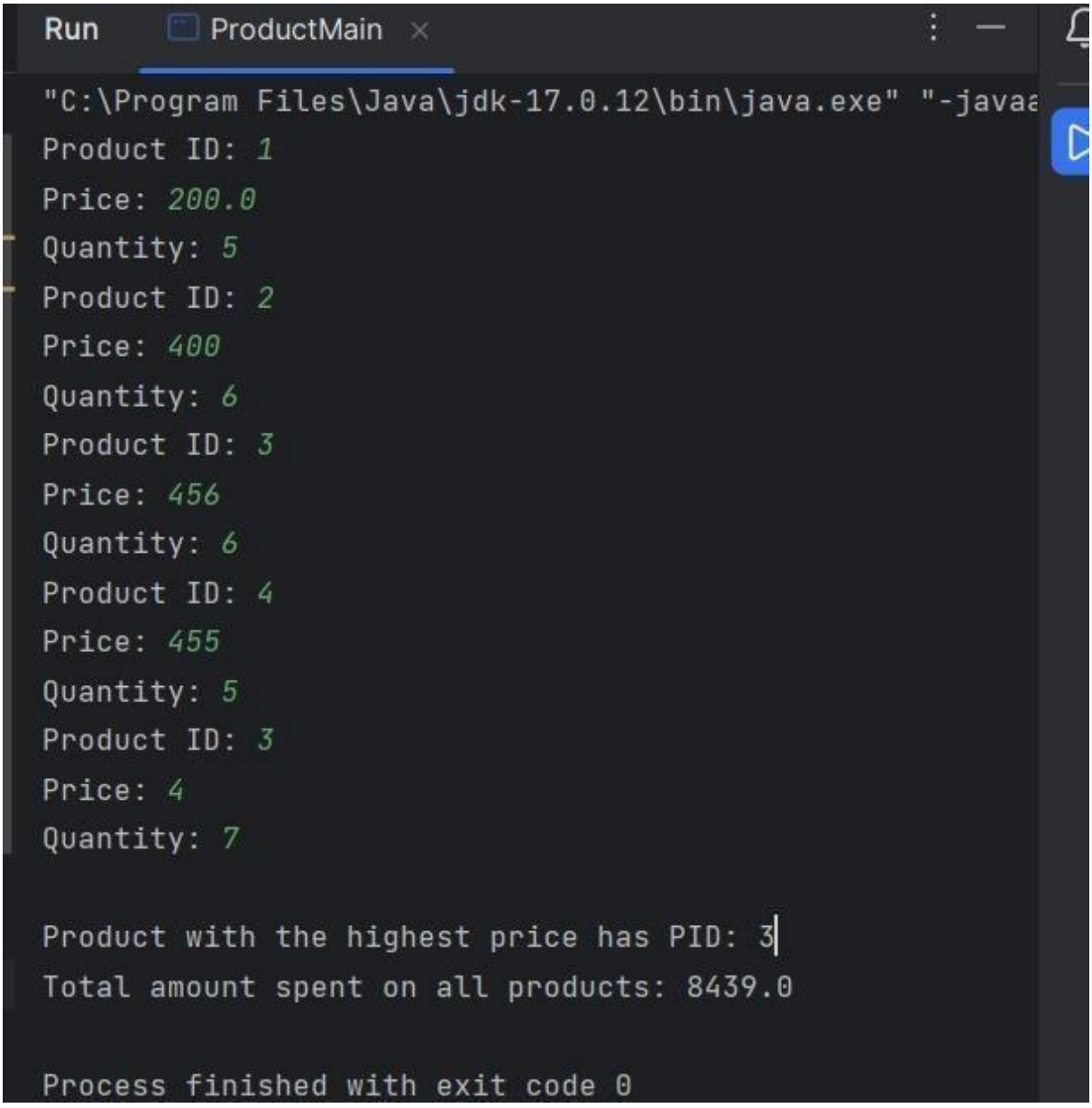
ution > src > ProductMain

Stories
ior IPS Office...

Search

```
            }
        }
        System.out.println("Product with the highest price has PID: " + maxPid);
        double totalAmount = calculateTotalAmount(products);
        System.out.println("Total amount spent on all products: " + totalAmount);
    }
}
```

The output is:



1.3) Create Class Account with data member as Balance. Create two constructors (no argument, and with argument) and perform following task

a. method to deposit the amount to the account.

b. method to withdraw the amount from the account.

c. method to display the Balance

Main.java | Account.java | AccountBalance.java

```
2 usages
public class Account {
    6 usages
    double balance;
    no usages
    public Account(){
        balance = 0.0;
    }
    1 usage
    Account(double balance){
        this.balance = balance;
    }
    1 usage
    public void deposit(double amount){
        balance += amount;
    }
    2 usages
    public void withdraw(double amount){
        if(amount <= balance){
            balance -= amount;
        }
        else{
            System.out.println("insufficient balance");
        }
    }
    1 usage
    public void displayBalance(){
        System.out.println(balance);
```

tions2 > src > Account > Account

Main.java | Account.java | AccountBalance.java

```
public class AccountBalance {
    public static void main(String[] args) {
        Account acc = new Account( balance: 1000.0);
        acc.deposit( amount: 500);
        acc.withdraw( amount: 200);
        acc.displayBalance();
        acc.withdraw( amount: 1500);
    }
}
```

The output is:

```
"C:\Program Files\Java\jdk-17.0.12\bin\java.exe" "-javaagent:C:\Users\
1300.0
insufficient balance

Process finished with exit code 0
```

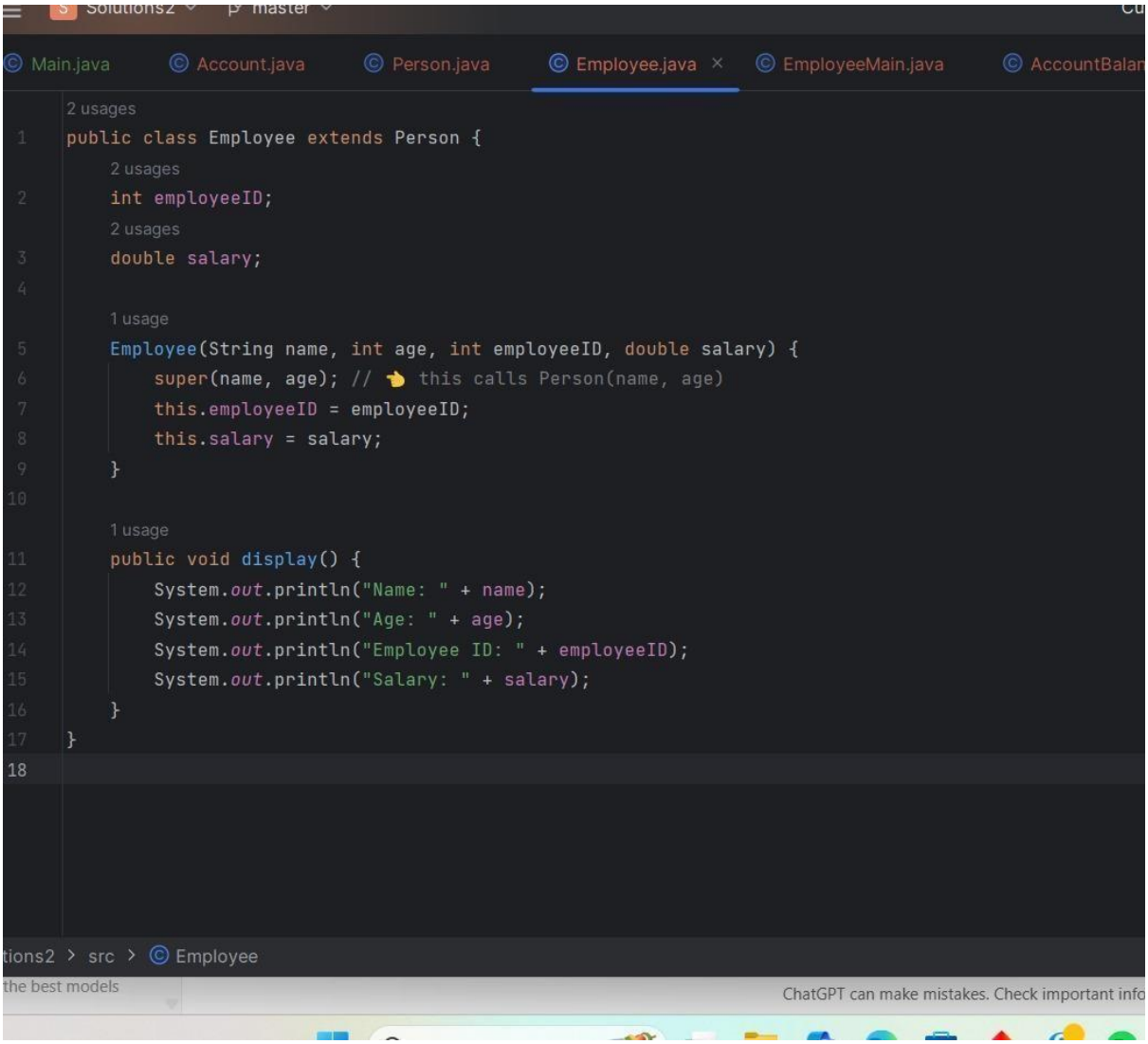
1.4) Define a base class Person with attributes name and age.

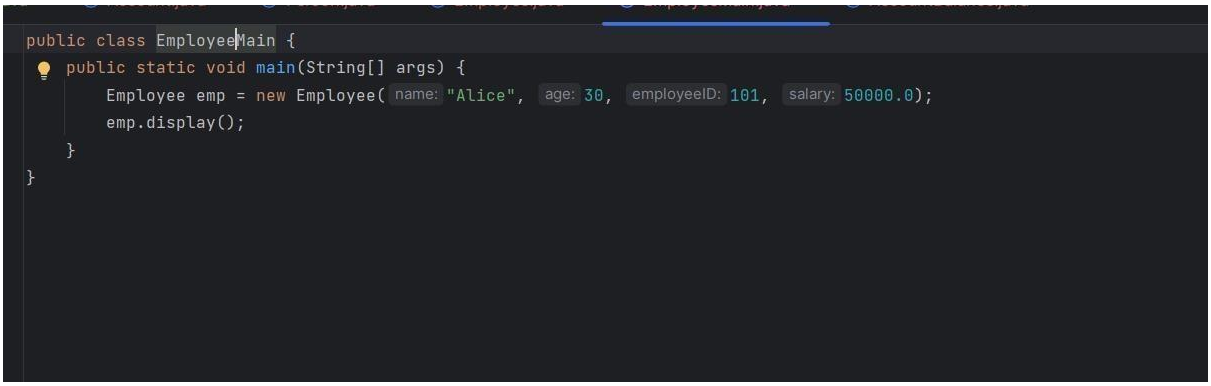
Create a subclass Employee that inherits from Person and adds attributes like employeeID and salary.

Use the super keyword to initialize the Person attributes in the Employee constructor.

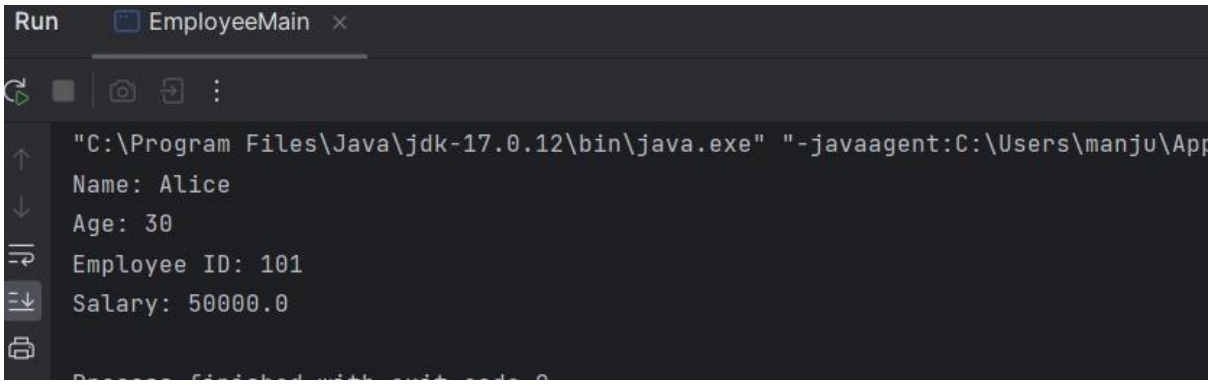
```
1 usage  1 inheritor
public class Person {
    2 usages
    String name;
    2 usages
    int age;
    1 usage
    Person(String name,int age){
        this.name = name;
        this.age = age;
    }

}
```





The output is:


Run
EmployeeMain
"C:\Program Files\Java\jdk-17.0.12\bin\java.exe" "-javaagent:C:\Users\manju\App
Name: Alice
Age: 30
Employee ID: 101
Salary: 50000.0