JAVA DAY -8

TASK 2

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

Input -

package task.two;

public class Person {

public String name;

public int age;

public Person(String name){

this.name = name;

this.age = 18;

}

public Person(String name,int age){

this.name=name;

this.age=age;

}

public void display(){

System.*out*.println("Name is:"+ name);

System.*out*.println("Age is :" + age);

}

public static void main(String[] args) {

Person p1 = new Person("Nivi");

Person p2 = new Person("John", 58);

p1.display();

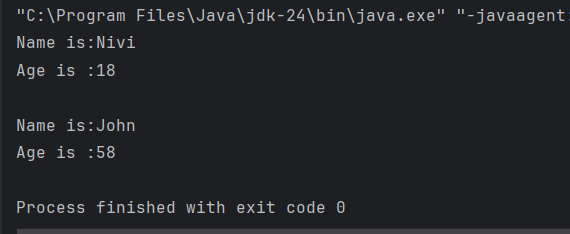
System.*out*.println(" ");

p2.display();

}

}

Output -



1. 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

Input :

Product.java

package task.two.two;

public class Product {

public int pid;

public double price;

public int quantity;

public Product(int pid, double price, int quantity) {

this.pid = pid;

this.price = price;

this.quantity = quantity;

}

}

ProductMain.java

package task.two.two;

import java.util.Scanner;

public class ProductMain {

public static void main(String[] args) {

Scanner s1 = new Scanner(System.*in*);

Product[] products = new Product[5];

for (int i = 0; i < 5; i++) {

System.*out*.println("Enter the Product\_id,Price and Quantity of the product "+(i+1)+" =");

int pid = s1.nextInt();

double price = s1.nextDouble();

int quantity = s1.nextInt();

products[i] = new Product(pid,price,quantity);

}

double maxPrice = products[0].price;

double maxPid = products[0].pid;

for (int i = 0; i <products.length ; i++) {

if (products[i].price > maxPrice) {

maxPrice = products[i].price;

maxPid = products[i].pid;

}

}

System.*out*.println("Pid of the product with the highest price: " + maxPid);

double totalCostAmount = *totalAmount*(products);

System.*out*.println("Total amount spent on all products :"+totalCostAmount);

}

public static double totalAmount(Product[] products){

double total = 0;

for (Product p : products) {

total += p.price \* p.quantity;

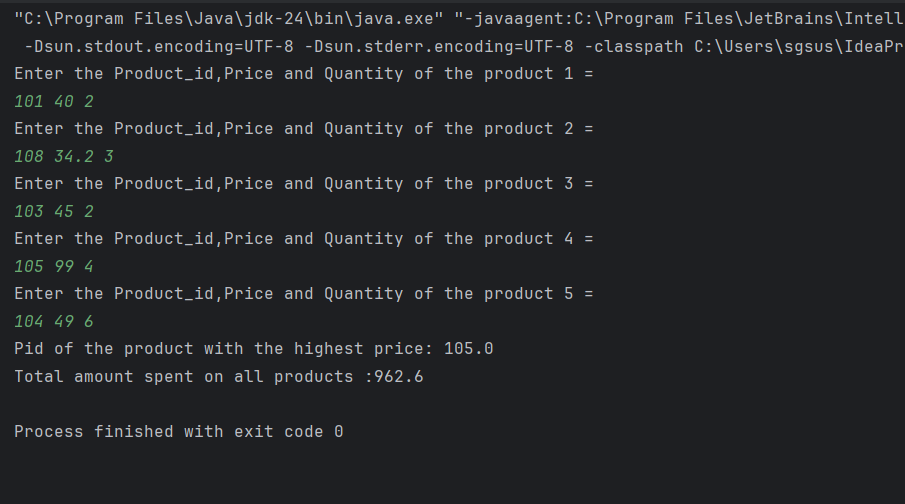
}

return total;

}

}

Output :



1. 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

Input:

package task.two.three;

public class Account {

private double balance;

public Account(){

balance = 0.0;

}

public Account(double initialBalance) {

balance = initialBalance;

}

public void deposit(double amount) {

if (amount > 0) {

balance = balance + amount;

System.*out*.println("Deposited Amount: " + amount);

} else {

System.*out*.println("Invalid Amount");

}

}

public void withdraw(double amount) {

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

balance = balance - amount;

System.*out*.println("Withdraw Amount: " + amount);

} else {

System.*out*.println("Invalid amount");

}

}

public void displayBalance(){

System.*out*.println("The Balance : "+ balance);

}

public static void main(String[] args) {

Account acc1 = new Account();

Account acc2 = new Account(200.00);

System.*out*.println("Account-1:");

acc1.displayBalance();

acc1.deposit(700);

acc1.withdraw(480);

acc1.displayBalance();

System.*out*.println("Account-2:");

acc2.displayBalance();

acc2.deposit(5000);

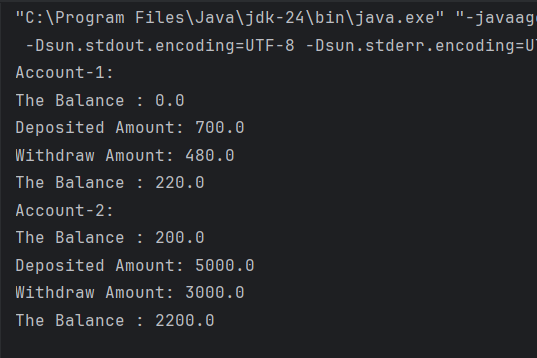
acc2.withdraw(3000);

acc2.displayBalance();

}

}

Output:



1. 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

Input:

Person.java

package task.two.four;

public class Person {

public String name;

public int age;

public Person(String name , int age){

this.name = name;

this.age = age;

}

public void displayPerson(){

System.*out*.println("Name: " + name);

System.*out*.println("Age: " + age);

}

}

Employee.java

package task.two.four;

public class Employee extends Person{

public int employeeID;

public double salary;

public Employee(String name, int age, int employeeID, double salary) {

super(name, age);

this.employeeID = employeeID;

this.salary = salary;

}

public void displayEmployee() {

super.displayPerson();

System.*out*.println("Employee ID: " + employeeID);

System.*out*.println("Salary: " + salary);

}

public static void main(String[] args) {

Employee emp = new Employee("John", 57, 1105, 290000.0);

emp.displayEmployee();

}

}

Output :

