Санкт-Петербургский Политехнический Университет

Институт компьютерных наук и технологий

«Высшая школа программной инженерии»

**Отчёт по лабораторной работе**

по дисциплине: «Системный анализ и принятие решений»

Выполнил:

студент гр. в3530904/80322 Мирошниченко В.А.

Руководитель: Амосов В.В.

Санкт-Петербург

2020

**Постановка задачи**

Себестоимость W-Polo – 15000$

Цена в 1 год – 25000$

Цена в 2 год – 10000$

Цена в 3 год – 1000$

Вероятность спроса в 1 и 2 годы представлены в таблице:

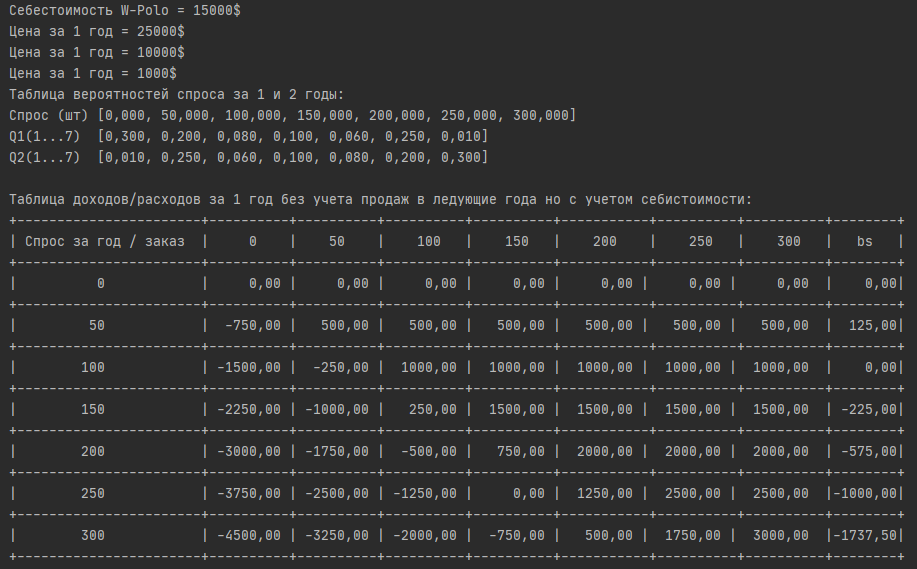
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Спрос (шт)** | **0** | **50** | **100** | **150** | **200** | **250** | **300** |
| **Q1(1…7)** | 0,3 | 0,2 | 0,08 | 0,10 | 0,06 | 0,25 | 0,01 |
| **Q2(1…7)** | 0,01 | 0,25 | 0,06 | 0,10 | 0,08 | 0,2 | 0,3 |

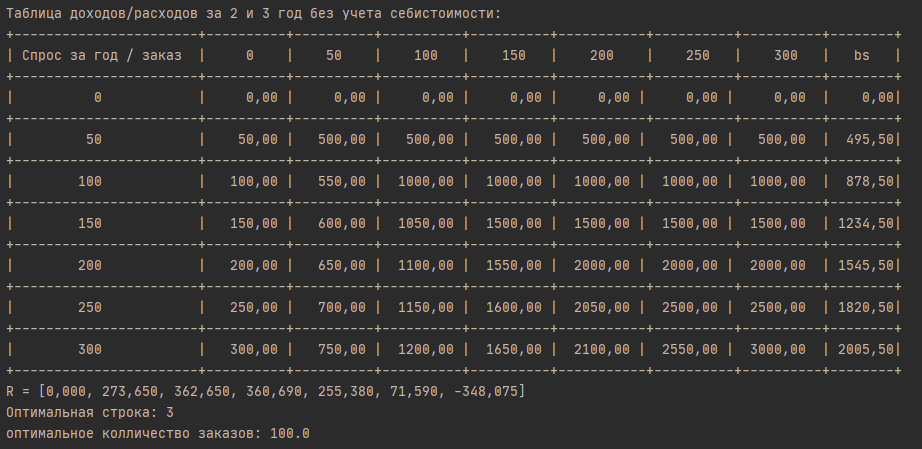
Если автомобиль марки «Volkswagen Polo» продан в первый год, он может быть реализован за 25000$. Если автомобиль продаётся во второй год, то его стоимость составляет 10000$. Иначе автомобиль продаётся гарантированно в третий год за 1000$.

Необходимо определить, вывеси таблицу Таблица доходов/расходов за 1 год без учета продаж в следующие годы, но с учетом себестоимости. Таблицу доходов/расходов за 2 и 3 годы без учета себестоимость. Используя метод Байеса-Лапласа, посчитать математическое ожидание.

Посчитать оптимальное количество заказов по формуле:

**Результаты работы программы**





Выводы: по полученным данным в ходе выполнения лабораторной работы, можно сделать вывод, что оптимальным количество заказов будет 100 автомобилей. Максимальная прибыль при этом составит 362,650 тыс. долларов.

**Код программы**

Исходный код программы, написанный на языке программирования Java:

package com.company;  
  
import java.util.Arrays;  
  
public class Main {  
  
 public static void main(String[] args) {  
 System.*out*.println("Себестоимость W-Polo = 15000$");  
 System.*out*.println("Цена за 1 год = 25000$");  
 System.*out*.println("Цена за 1 год = 10000$");  
 System.*out*.println("Цена за 1 год = 1000$");  
  
 double[][] probability = new double[][] {{0,50,100,150,200,250,300},{0.3, 0.2, 0.08, 0.10, 0.06, 0.25, 0.01},{0.01, 0.25, 0.06, 0.10, 0.08, 0.2, 0.3}};  
  
 System.*out*.println("Таблица вероятностей спроса за 1 и 2 годы:");  
  
 System.*out*.print("Спрос (шт) "); *printVector*(probability[0]);  
 System.*out*.print("Q1(1...7) "); *printVector*(probability[1]);  
 System.*out*.print("Q2(1...7) "); *printVector*(probability[2]);  
  
 //таблица вероятности спроса за 1 и 2 год  
  
 double[][] mat = *getResultMatrix*(probability[0], true);  
 System.*out*.println();  
 double[] s1 = *bayesLaplaceMethod*(mat, probability[1]);  
 System.*out*.println("Таблица доходов/расходов за 1 год без учета продаж в ледующие года но с учетом себистоимости:");  
 *printMatrix*(mat, probability, s1);  
  
 double[][] mat2 = *getResultMatrix*(probability[0], false);  
 double[] s2 = *bayesLaplaceMethod*(mat2, probability[2]);  
 System.*out*.println("Таблица доходов/расходов за 2 и 3 год без учета себистоимости:");  
 *printMatrix*(mat2, probability, s2);  
  
 double[] r = *getR*(probability, s1, s2);  
  
 System.*out*.print("R = ");  
 *printVector*(r);  
 double max = Arrays.*stream*(r).max().getAsDouble();  
 int maxInd = *findIndexInArray*(r, max);  
 System.*out*.printf("Оптимальная строка: %1d%n", maxInd);  
 System.*out*.println("оптимальное колличество заказов: "+probability[0][--maxInd]);  
  
 }  
  
 private static void printMatrix(double[][] matrix, double[][] probability, double[] bs) {  
 System.*out*.format("+-----------------------+----------+----------+----------+----------+-----------+----------+----------+--------+%n");  
 System.*out*.format("| Спрос за год / заказ | 0 | 50 | 100 | 150 | 200 | 250 | 300 | bs |%n");  
 System.*out*.format("+-----------------------+----------+----------+----------+----------+-----------+----------+----------+--------+%n");  
  
 for (int i=0; i<matrix.length; i++) {  
 System.*out*.format("|%11s%12s", String.*format*("%.0f",probability[0][i])," ");  
 for (int j=0; j<matrix[0].length; j++){  
 System.*out*.format("| %-8s ", String.*format*("%8.2f",matrix[i][j]));  
 }  
 System.*out*.format(" |%-8s|%n",String.*format*("%8.2f",bs[i]));  
 System.*out*.format("+-----------------------+----------+----------+----------+----------+-----------+----------+----------+--------+%n");  
 }  
 }  
  
 //процедура вывода вектора в консоль  
 private static void printVector(double[] vector) {  
 System.*out*.print("[");  
 for (int i = 0; i < vector.length; i++) {  
 if(i != vector.length-1) {  
 System.*out*.printf("%.3f, ", vector[i]);  
 } else {  
 System.*out*.printf("%.3f", vector[i]);  
 }  
 }  
 System.*out*.print("]");  
 System.*out*.println();  
 }  
 //Функция поиска индекса  
 private static int findIndexInArray(double[] array, double element) {  
 int index = -1;  
 for (int i = 0; i < array.length; i++) {  
 if(array[i] == element) {  
 index = i;  
 break;  
 }  
 }  
  
 return ++index;  
 }  
 //Заполнение таблицы решений (спроса и предложений)  
 private static double[][] getResultMatrix(double[] array, boolean isCostPrise){  
 double[][] answerMat = new double[array.length][array.length];  
  
  
 for (int i = 0; i < answerMat.length; i++) {  
 for (int j = 0; j < answerMat.length; j++) {  
 if(isCostPrise) {  
 if (i == j || i < j) {  
 answerMat[i][j] = 25 \* array[i] - 15 \* array[i];  
 } else {  
 answerMat[i][j] = 25 \* array[j] - 15 \* array[i];  
 }  
 } else {  
 if (i == j || i < j) {  
 answerMat[i][j] = 10 \* array[i];  
 } else {  
 answerMat[i][j] = 10 \* array[j] + (array[i]-array[j]);  
 }  
 }  
 }  
 }  
  
 return answerMat;  
 }  
 //Реализация критерия Байеса-Лапласа  
 private static double[] bayesLaplaceMethod(double[][] resultMatrix, double[] probabilityVector) {  
 double[] e = new double[probabilityVector.length];  
  
 for (int i = 0; i < resultMatrix.length; i++) {  
 for (int j = 0; j < resultMatrix[0].length; j++){  
 e[i] += resultMatrix[i][j] \* probabilityVector[j];  
 }  
 }  
  
 return e;  
 }  
  
 private static double[] getR(double[][] probability,double[] s1, double[] s2) {  
 double[] r = new double[probability[0].length];  
  
 for(int i=0; i<7;i++) {  
 double s = 0;  
 for(int j=0; j<i;j++){  
 s += probability[1][j]\*s2[i-j];  
 }  
 r[i] = s1[i]+s;  
 }  
  
 return r;  
 }  
  
}