Министерство науки и высшего образования РФ

Севастопольский государственный университет

Кафедра информационных систем

ОТЧЕТ

по лабораторной работе №2

ОСНОВЫ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ JAVA

по дисциплине «Платформа Java»

Выполнил:

Студент группы ИС/б 17-2-о

Черняев Н.Г.

Проверил:

Кузнецов С.А.

г. Севастополь 2019

**1.Цель работы**

В ходе выполнения данной лабораторной работы необходимо ознакомиться с особенностями объектно-ориентированного программирования (ООП) на языке Java, приобрести практические навыки программирования на языке Java с использованием основных принципов ООП.

**2.Постановка задачи**

**Вариант №10**

2.1. Описать абстрактный класс CBuffer, содержащий следующие поля:

* идентификатор буфера (int bufID) – уникальный идентификатор буфера;
* размер буфера (int bufSize) – максимальный размер буфера;
* количество созданных буферов (int BufCount).

Доступ к полям класса CBuffer должны иметь только методы этого класса и методы его потомков. Для организации доступа к этим полям из других классов необходимо реализовать общедоступные методы:

* int GetBufCount();
* int GetBufID().

Реализовать конструктор класса CBuffer(int count), выполняющий инициализацию идентификатора буфера(в качестве идентификатора использовать номер по порядку создаваемого буфера), размера буфера (значением count, передаваемым конструктору), увеличение количества созданных буферов.

В классе CBuffer описать абстрактный метод Generate()

2.2. Реализовать дочерний класс для создания буфера, хранящего значения заданного типа Double. Для хранения значений реализовать поле – массив значений типа Double. В конструкторе класса использовать вызов конструктора родительского класса CBuffer, и кроме того создать массив значений типа Double (с использованием оператора new) и проинициализировать его с иcпользованием метода Generate().

Pеализовать метод Generate(), заполняющий массив случайными числами.

2.3. Описать интерфейсы:

1) IBufferPrintable – описывающий методы вывода на экран:

* public void PrintInfo() – выводит на экран идентификатор, тип и размер буфера.
* public void Print() – выводит на экран содержимое буфера.
* public void PrintFirstN(int n) – выводит на экран первые n элементов буфера.
* public void PrintLastN(int n) – выводит на экран последние n элементов буфера

2) IBufferSortable – описывает метод для сортировки массива.

* public void Sort(); - сортирует массив методом выборки

3) IBufferComputable – описывает методы для вычисления статистики значений буфера.

* public void Min() – вычисляет минимальный элемент буфера;

4) IBufferStorable – описывает методы для выгрузки буфера в текстовый файл.;

* public void SaveSeparateLines (String filename) – сохраняет буфер в файл по одному элементу в строке

2.4. Создать произвольный класс, унаследованный от класса, разработанного при выполнении п. 2.2, и реализующий методы интерфейсов из п. 2.3.

2.5. Реализовать класс Lab2Java, в методе main которого в соответствии с вариантом задания реализовать работу с объектами класса.

* Создать 7 буферов заданного типа Double и размера 40;
* Вывести на экран информацию o буферах;
* Вывести на экран первые 10 элементов буферов;
* Вычислить функцию Мin для каждого буфера и вывести результат на экран;
* Выполнить сортировку буферов методом Выборки;
* Вывести на экран первые 10 элементов буферов
* Сохранить буферы в файл с использованием метода SaveSeparateLines.

**3.Текст программы**

Класс lab:

**package** lab2;

**import** java.util.\*;

**import** java.io.\*;

**abstract** **class** CBuffer{

**static** **int** *IDNumb* = 1;

**protected**

**int** BufCount;

**int** BufID;

**int** BufSize;

**public**

CBuffer(**int** count) {

BufSize = count;

BufCount = BufID = *IDNumb*;

*IDNumb*++;

}

**int** GetBufCount() {

**return** BufCount;

}

**int** GetBufID() {

**return** BufID;

}

**abstract** Double[] Generate(Double[] mas);

}

**class** BBuffer **extends** CBuffer{

**protected** Double[] BufElem;

BBuffer(**int** count){

**super**(count);

BufElem = **new** Double[count];

BufElem = Generate(BufElem);

}

@Override

Double[] Generate(Double[] mas) {

Random random = **new** Random();

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

mas[i] = random.nextDouble();

}

**return** mas;

}

}

**class** ABuffer **extends** BBuffer **implements** IBufferComputable,

IBufferPrintable,

IBufferSortable,

IBufferStorable

{

ABuffer(**int** count){

**super**(count);

}

**public** **void** PrintInfo() {

System.***out***.println("ID= " + BufID + "\n" +

"Size= " + BufSize + "\n"+

"Type= Double");

}

**public** **void** Print() {

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

System.***out***.println(BufElem[i] + " ");

**if** (i%10 == 0)

System.***out***.print("\n");

}

}

**public** **void** PrintFirstN(**int** n) {

**for** (**int** i = 0; i < n; i++) {

System.***out***.println(BufElem[i] + " ");

}

}

**public** **void** PrintLastN(**int** n) {

**for** (**int** i = n; i <= BufElem.length; i++) {

System.***out***.println(BufElem[i] + " ");

}

}

**public** **void** Sort() {

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

Double min = BufElem[i];

**int** min\_i = i;

**for** (**int** j = i+1; j < BufElem.length; j++) {

**if** (BufElem[j] < min) {

min = BufElem[j];

min\_i = j;

}

}

**if** (i != min\_i) {

Double tmp = BufElem[i];

BufElem[i] = BufElem[min\_i];

BufElem[min\_i] = tmp;

}

}

}

**public** **void** Min() {

Double min = BufElem[0];

**for**(**int** i = 0; i != BufElem.length; i ++){

**if**(BufElem[i] < min) {

min = BufElem[i];

}

}

System.***out***.println("Min = " + min);

}

**public** **void** SaveSeparateLines(String filename) **throws** IOException {

File fileout = **null**;

BufferedWriter bufferedWriter = **null**;

FileWriter fileWriter = **null**;

fileout = **new** File("E:\\" + filename);

**if** (!fileout.exists()) {

fileout.createNewFile();

}

fileWriter = **new** FileWriter(fileout,**true**);

bufferedWriter = **new** BufferedWriter(fileWriter);

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

bufferedWriter.newLine();

bufferedWriter.write(BufElem[i] + "");

}

bufferedWriter.write("\r\n");

bufferedWriter.flush();

bufferedWriter.close();

}

}

**public** **class** lab {

**public** **static** **void** main(String[] args) **throws** IOException {

**int** N = 7;

**int** L = 40;

ABuffer[] BufferMas = **new** ABuffer[N];

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

BufferMas[i] = **new** ABuffer(L);

}

**for**(ABuffer buffers : BufferMas) {

buffers.PrintInfo();

buffers.PrintFirstN(10);

buffers.Min();

buffers.Sort();

buffers.PrintFirstN(10);

buffers.SaveSeparateLines("1.txt");

}

}

}

Интерфейс IBufferComputable:

**package** lab2;

**public** **interface** IBufferComputable {

**public** **void** Min();

}

Интерфейс IBufferPrintable:

**package** lab2;

**public** **interface** IBufferPrintable {

**public** **void** PrintInfo();

**public** **void** Print();

**public** **void** PrintFirstN(**int** n);

**public** **void** PrintLastN(**int** n);

}

Интерфейс IBufferSortable:

**package** lab2;

**public** **interface** IBufferSortable {

**public** **void** Sort();

}

Интерфейс IBufferStorable:

**package** lab2;

**import** java.io.IOException;

**public** **interface** IBufferStorable {

**public** **void** SaveSeparateLines(String filename) **throws** IOException;

}

**4.Результаты**

Консоль:

ID= 1

Size= 40

Type= Double

0.8830422965807003

0.4276358378873245

0.8920304492917732

0.7210899791384584

0.1505159728957456

0.6067366789198768

0.4323575342000321

0.3048991734285723

0.5535165140482267

0.440166904780405

Min = 0.062360385326015066

0.062360385326015066

0.11591077286781704

0.11647142405319777

0.12028000543476935

0.12816361639180796

0.1353555901736162

0.1472588708320891

0.1505159728957456

0.15646685553039774

0.1884284412198144

ID= 2

Size= 40

Type= Double

0.3420253815060149

0.198234509919896

0.678883172866861

0.9661262134277641

0.45065892215949577

0.14425674676323086

0.7043829041019964

0.6784505119253506

0.9780682275859257

0.3722762288265484

Min = 0.007264456579268774

0.007264456579268774

0.008934818468601113

0.05598958209088867

0.09121012160199016

0.1164987856199361

0.14425674676323086

0.18348620672191374

0.1856545565898332

0.198234509919896

0.2061788072984998

ID= 3

Size= 40

Type= Double

0.6979043204647847

0.7662420888114305

0.5152655601236755

0.9891639451756673

0.3387052874650499

0.33335291572053416

0.6378308037053337

0.7608924667962006

0.8061777557409482

0.47703152833793083

Min = 0.058546068420351705

0.058546068420351705

0.09468117724331193

0.13987127578072667

0.19508813121951307

0.23221224367486226

0.25276877203585124

0.27648534236661415

0.3189510391914875

0.33335291572053416

0.3387052874650499

ID= 4

Size= 40

Type= Double

0.9163652921723565

0.6061961258925334

0.5178974532126691

0.2155510490481679

0.19805583023209705

0.7308737530930953

0.5765257601451937

0.23469423238940945

0.8715851016764175

0.5531171163229951

Min = 0.05836007342793381

0.05836007342793381

0.08382967414066966

0.0842448675319275

0.09734830765943492

0.13822825827681895

0.19805583023209705

0.2155510490481679

0.23379716974289866

0.23469423238940945

0.36331240428097

ID= 5

Size= 40

Type= Double

0.22828669853796657

0.4508788125717942

0.08892846965206247

0.7419224658200363

0.18857975372389812

0.1848109568622215

0.2849590346709473

0.9104100160609176

0.6861793299348451

0.5022361410585724

Min = 0.031102404337055467

0.031102404337055467

0.05945274654756327

0.061875209920476015

0.08871870877658461

0.08892846965206247

0.09534036839474802

0.10764261741383441

0.13554297060567366

0.1361728431135628

0.16581380472352514

ID= 6

Size= 40

Type= Double

0.1985910843247849

0.41447543210067006

0.6838575203063957

0.9454782225635847

0.06925940735730507

0.9020526308885874

0.28106657318071226

0.19775491736444195

0.7841629007259538

0.6490335550089399

Min = 0.02083700028745583

0.02083700028745583

0.03291080243768074

0.06925940735730507

0.0803758246404298

0.16469924552031634

0.18452363749321055

0.19775491736444195

0.1985910843247849

0.223549292200781

0.28106657318071226

ID= 7

Size= 40

Type= Double

0.13912083010266185

0.5396310160684779

0.9750034977981864

0.9727691251601908

0.7969021123826054

0.9350480702730759

0.9973301684274127

0.20727814012934132

0.9130513478895613

0.31660712911097244

Min = 0.13069540763644394

0.13069540763644394

0.13912083010266185

0.1465826608098313

0.18434271241036015

0.20121507573867647

0.20727814012934132

0.2125617660189142

0.2134846708249365

0.2345788099343491

0.2352064947849014

Файл 1.txt:

0.062360385326015066

0.11591077286781704

0.11647142405319777

0.12028000543476935

0.12816361639180796

0.1353555901736162

0.1472588708320891

0.1505159728957456

0.15646685553039774

0.1884284412198144

0.19565138067170884

0.27249678806090527

0.3048991734285723

0.3661562079152211

0.4038085870104431

0.41314966271280573

0.4276358378873245

0.4323575342000321

0.4363584607426477

0.440166904780405

0.4462882185950606

0.5392671649277294

0.5535165140482267

0.5647919649687642

0.5744340009109837

0.5841740452139452

0.5850980660036447

0.6067366789198768

0.6107472423587166

0.6186671693298649

0.6308025107278253

0.6423515358943886

0.6758793683165965

0.7210899791384584

0.7217174702468725

0.775557418901869

0.785546176057009

0.8830422965807003

0.8920304492917732

0.9176180717504298

0.007264456579268774

0.008934818468601113

0.05598958209088867

0.09121012160199016

0.1164987856199361

0.14425674676323086

0.18348620672191374

0.1856545565898332

0.198234509919896

0.2061788072984998

0.23157649153191018

0.23561034681923898

0.2696008285849234

0.32254146673726924

0.3241669561031305

0.3420253815060149

0.3722762288265484

0.3751903014115736

0.42225712010817507

0.42343035739041746

0.45065892215949577

0.5092509258398173

0.5431097258822843

0.5507920296364658

0.5536615120447389

0.5816363318374094

0.6124071417381703

0.6147068845050381

0.6784505119253506

0.678883172866861

0.7008170188776957

0.7043829041019964

0.7920754218616322

0.8110746334704501

0.8207931982670436

0.8579761216076629

0.8731500337469376

0.9661262134277641

0.9780682275859257

0.9940788818546521

0.058546068420351705

0.09468117724331193

0.13987127578072667

0.19508813121951307

0.23221224367486226

0.25276877203585124

0.27648534236661415

0.3189510391914875

0.33335291572053416

0.3387052874650499

0.34241856617676536

0.44701604557963226

0.4506314540122952

0.47703152833793083

0.483239439531137

0.4896050830089904

0.5152655601236755

0.5249269887665943

0.5684558146618341

0.6285226740510709

0.6303455821164821

0.6328110060685039

0.6378308037053337

0.6423445667237191

0.6979043204647847

0.7149986422873655

0.7334840994136334

0.7608924667962006

0.7662420888114305

0.8061777557409482

0.8378989927298532

0.8513484622945021

0.8925667179215074

0.9101650983793268

0.9348813672940621

0.9467006994096274

0.9542038542007488

0.9730190048737923

0.9805197314989206

0.9891639451756673

0.05836007342793381

0.08382967414066966

0.0842448675319275

0.09734830765943492

0.13822825827681895

0.19805583023209705

0.2155510490481679

0.23379716974289866

0.23469423238940945

0.36331240428097

0.36393688208984254

0.40370792474500483

0.4558797171038522

0.48045018562674635

0.4914267698890499

0.5022635550162172

0.5067019400561784

0.5178974532126691

0.5513805991203603

0.5531171163229951

0.5765257601451937

0.6061961258925334

0.7008977202788752

0.7172300076870562

0.7308737530930953

0.7639064066320613

0.7886285304559638

0.7887864106956449

0.7948699930695766

0.7954628346095595

0.8054965263464741

0.8133064181729628

0.8354913478379904

0.8606910173813327

0.8715851016764175

0.8748129130654027

0.9163652921723565

0.9197027592700019

0.9975468498603812

0.9986498925828897

0.031102404337055467

0.05945274654756327

0.061875209920476015

0.08871870877658461

0.08892846965206247

0.09534036839474802

0.10764261741383441

0.13554297060567366

0.1361728431135628

0.16581380472352514

0.1848109568622215

0.18703503947213562

0.18857975372389812

0.1900142354581945

0.22828669853796657

0.2849590346709473

0.286538631246982

0.368078925060795

0.44183692377928074

0.4508788125717942

0.47010226136544453

0.4889189975818933

0.4955992000543632

0.5022361410585724

0.5101438778805096

0.5633571082177534

0.5689691888354788

0.5738055799044431

0.583014385627014

0.6285332893529145

0.6363118413978106

0.646084618955083

0.6539533466840026

0.6802468660492491

0.6861793299348451

0.7419224658200363

0.7494536088542697

0.7727335746059731

0.8513696647829738

0.9104100160609176

0.02083700028745583

0.03291080243768074

0.06925940735730507

0.0803758246404298

0.16469924552031634

0.18452363749321055

0.19775491736444195

0.1985910843247849

0.223549292200781

0.28106657318071226

0.30126481433269536

0.3170723874111778

0.4053937962542983

0.41447543210067006

0.43627609080386864

0.543233029465469

0.6490335550089399

0.6692554476305634

0.6805954550842934

0.6838575203063957

0.6860146384331482

0.7053637623703761

0.7582130844966295

0.7588307123939023

0.7841629007259538

0.7863371000753351

0.7904711662518322

0.8230474576152974

0.8270088501533728

0.8527229019357275

0.8783612955674396

0.8799123100170224

0.8824558734142092

0.9020526308885874

0.9092529428120011

0.927113218215224

0.9280374461313007

0.9454782225635847

0.9819097313295616

0.9944978974941976

0.13069540763644394

0.13912083010266185

0.1465826608098313

0.18434271241036015

0.20121507573867647

0.20727814012934132

0.2125617660189142

0.2134846708249365

0.2345788099343491

0.2352064947849014

0.31027905644725773

0.31660712911097244

0.3483879023290486

0.38912138669910556

0.42656984670344833

0.4409107577035908

0.4585003387838942

0.505412757643088

0.5396310160684779

0.5938655882364796

0.6547775862177881

0.6881834138969452

0.6981831897646994

0.7291226514080306

0.7668975871466543

0.7969021123826054

0.827507367200726

0.8288210220792465

0.841182634031465

0.8764290956525272

0.8770301992057822

0.9032058561492439

0.9054516594458272

0.9130513478895613

0.9350480702730759

0.9544828709758254

0.9620918374090934

0.9727691251601908

0.9750034977981864

0.9973301684274127

**5.Вывод**

Вданной лабораторной работе были изучены особенности объектно-ориентированного программирования (ООП) на языке Java, приобретены практические навыки программирования на языке Java с использованием основных принципов ООП.