for(Page page : pages){page.inc();

//替换算法

public void replace(int pageId){//寻找时间数最大的页面

int max=0,perCount,outPageId =-1,cur = 0;//cur为下标

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

perCount = pages[i].getCount();

if(max<perCount){

max = perCount:

outPageId=pages[i].getId();//换出去的页号

cur =i;

reSet(cur);pages[cur].setId(pageId);System.out.println("页号ID:"+pageId+”正在放入内存,页号ID:"+outPageId+"被替换出去");

//显示当前内存页public void display(){System.out.print("当前内存保密的页数是:");for(Page page :pages){System.out.print(page.getId()+" ");

System.out.println();

页面

class Pagefprivate int id = -1;private int count;public int getId() {return id;

public void setId(int id){this.id= id;

public int getCount()f

return count;

public void inc() {count ++;

public void setCount(int count){this.count = count;

public class FIFO {

/米米

\*先进先出算法

\*/

private static finalint PRO\_MEMORY =5;//系统分配的内存块数private static Page]pages =new Page[PRO\_MEMORY];private staticint countOldPoint;//纪录最久的页面下标private static int count;//纪录当前在使用的总页面数

private static int lackTime;//缺页次数

private List<Integer>usePageNumlist= new ArrayList<Integer>();//页面使用列表

public static void main(String[] args){

System.out.println("--------先进先出算法------------")

FIFO fifo = new FIFO();

fifo.initO;

fifo.input();

fifo.running();

//初始化public void init(){for(int i=0;i<pages.length;i++){pages[i]= new Page();

//接收用户输入列表public void input(){System.out.print("请输入页面使用列表,以空格分开:");Scanner sc=new Scanner(System.in);String input=sc.nextLine();String[] inputList = input.split("\\s");try{for(String in :inputList)fusePageNumList.add(Integer.value0f(in));catch(Exception e){System.out.println("输入的必须是数字,请重新开始!!!");System.exit(0);

子

//系统运行public void running(){Iterator<Integer>it= usePageNumList.iterator();//列表置入替换while(it.hasNext()){countOldPoint = countOldPoint % PRO\_MEMORY;int inPageId =it.next(;//查找内存中是否有该页面if(search(inPageId)){System.out.println("内存中有ID为"+inPageId+”这个页面,不进行替换"):else if(count<PRO\_MEMORY){//有空闲内存页pages[count].setId((Integer)inPageId);System.out.println("页号ID:"+pages[count].getId()+”正在放入内存");count ++;else{//替换replace(inPageId);lackTime ++;countOldPoint ++;display();System.out.println("缺页次数为:"+lackTime+",缺页率是:"+(float)lackTime/usePageNumList.size());

//查找内存中是否有该页面public boolean search(int pageId)ffor(int i=0;i<pages.length;i++){if(pages[i].getId()= pageId){return true,

return false,

//替换算法public void replace(int pageId)f//置换最久的页面int outPageId = -1;

outPageId =pages[count0ldPoint.getId();pages[count0ldPoint].setId(pageId);System.out.println("页号ID:"+pageId+”正在放入内存,页号ID:"+outPageId+"被替换出去")

//显示当前内存页public void display()fSystem.out.print("当前内存保留的页数是:");for(Page page : pages){System.out.print(page.getId()+" ");5

System.out.println();

public class OPT {

/米米

\*最佳置换算法

private static final int PRO\_MEMORY=5://系统分配的内存块数

private static Page2] pages = new Page2[PRO\_MEMORY;

private staticintcount;//纪录当前在使用的总页面数

private static int lackTime;//缺页次数

private List<Integer>usePageNumList= new ArrayList<Integer>();//页面使用列表

public static void main(String[] args){

System.out.println("

---最佳置换算法------------"):

OPT opt = new OPT();

opt.init();

opt.input();

opt.running();

//初始化public void init(){for(int i=0;i<pages.length;i++){pages[i]= new Page2();

//接收用户输入列表public void input(){System.out.print("请输入页面使用列表,以空格分开:");Scanner sc=new Scanner(System.in);String input = sc.nextLine();String[]inputList= input.split("\\s");try{for(String in :inputList)fusePageNumList.add(Integer.value0f(in));catch(Exception e)fSystem.out.println("输入的必须是数字,请重新开始!!!");System.exit(0);

//系统运行public void running()fIterator<Integer>it= usePageNumList .iterator();int currentPoint=0;//当前读取输入列表的下标//列表置入替换while(it.hasNext()){int inPageId= it.next();//查找内存中是否有该页面

if(search(inPageId)){System.out.println("内存中有ID为"+inPageId+”这个页面,不进行替换");

else if(count<PRO MEMORY)f//有空闲内存页

pages[count].setId( (Integer)inPageId):

System.out.println("页号ID:"+pages[count].getId()+”正在放入内存");

count ++:

timeRefresh();

elsef//替换replace(inPageId,currentPoint);//传入当前下标确定需要遍历的未来输入列表timeRefresh;lackTime ++;3currentPoint ++;display();

System,out.println("缺页次数为:"+lackTime+",缺页率是:"+(float)lackTime/usePageNumList.size());

//查找内存中是否有该页面public boolean search(int pageId){for(int i=0;i<pages.length;i++){if(pages[i].getId()== pageId){timeRefresh();return true,

return false;

//时间更新public void timeRefresh(){for(Page2 page :pages){page.inc();

//替换算法public void replace(int pageId,int currentPoint){//寻找最长时间不使用的页面,count最大的内存块

int max =0,perCount,outPageId= -1,cur =0,searchCounter=0;//cur为内存块下标,searchCounter纪录是否内存块搜索完

//循环爆出最长为使用的页面

for(int i=currentPoint+1,i<usePageNumlist.size();i++){

for(Page2 page : pages){

if(page.getId()== usePageNumList.get(i)){

、

elsef

searchCounter ++;

page.inc();//未找到则增长值

3

if(searchCounter == pages.length){

break;

子

//进行搜索,查找替换目标

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

perCount = pages[i].getCount();

if(max<perCount){

max= perCount;

cur =

i;

outPageId = pages[i].getId();

System.out.println("--------当前-页号count------->"+perCount);

pages[curl.setId(pageId);System.out.println("页号ID:"+pageId+”正在放入内存,页号ID:"+outPageId+"被替换出去");reSet();

//搜索完毕进行重置public void reSet(){

for(Page2 page : pages){page.setCount(0);

4

//显示当前内存页

public void display(){

System.out.print("当前内存保留的页数是:");

for(Page2 page : pages){

System.out.print(page.getId()+" ");

System.out.println();

\*未来最久未使用纪录

class Page2fprivate int id= -1;

private int count;

public int getId(){

return id:

public void setId(int id) {this.id = id;

public int getCount(){return count;

public void inc() {

count ++;

public void setCount(int count){this.count = count;

public class LRU {/米米

\*最近最久未使用算法水/

private static final int PRO\_MEMORY=5;//系统分配的内存块数private static Page[]pages = new Page[PRO\_MEMORY];private staticintcount;//纪录当前在使用的总页面数

private static int lackTime;//缺页次数

private List<Integer>usePageNumList =new ArrayList<Integer>();//页面使用列表

public static void main(String[ args)f

System.out.println(".

LRU lru = new LRU();

---最近最久未使用算法------------");

lru.init();

lru.input(;

lru.running();

//初始化public void init(){for(int i=0;i<pages.length;i++){pages[i]= new Page();

//接收用户输入列表public void input(){System.out.print("请输入页面使用列表,以空格分开:");Scanner sc=new Scanner(System.in);String input= sc.nextLine();String[]inputList=input.split("\\s");try{

for(String in : inputList){usePageNumList.add(Integer.value0f(in));

catch(Exception e){System.out.println("输入的必须是数字,请重新开始!!!");System.exit(0);3

//系统运行public void running(){Iterator<Integer>it=usePageNumList.iterator()://列表置入替换while(it .hasNext()){int inPageId = it.next();//查找内存中是否有该页面if(search(inPageId)){System.out.println("内存中有ID为"+inPageId+”这个页面,不进行替换");2else if(count<PRO\_MEMORY){//有空闲内存页pages[count].setId((Integer)inPageId);System.out.println("页号ID:"+pages[count].getId()+”正在放入内存")reSet(count);count ++;timeRefresh();else{//替换replace(inPageId);timeRefresh();lackTime ++;display();System.out.println("缺页次数为:"+lackTime+",缺页率是:"+(float)lackTime/usePageNumList.size());//查找内存中是否有该页面public boolean search(int pageId)ffor(int i=0;i<pages.length;i++){if(pages[i].getId()== pageId){timeRefresh();reSet(i);return true;return false;//访问后置0public void reSet(int cur){pages[cur].setCount(0);

//时间更新public void timeRefresh(){