using System;

using System.Collections.Generic;

namespace Homework12.\_10.\_23

{

//4

struct Book

{

string title;

string author;

int year;

int price;

public Book(string title,string author,int year,int price)

{

this.title = title;

this.author = author;

this.year = year;

this.price = price;

}

public string Title

{

get=> title;

set => title = value;

}

public string Author

{

get => author;

set => author = value;

}

public int Year

{

get => year; set => year = value;

}

public int Price

{

get => price; set => price = value;

}

public void Print()

{

Console.WriteLine("Title: "+title+" Author: "+author+" Year: "+year+" Price: "+price);

}

public static bool operator ==(Book a, Book b)

{

if(a.price==b.price)

return true;

else return false;

}

public static bool operator !=(Book a, Book b)

{

if (a.price == b.price)

return false;

else return true;

}

public override string ToString()

{

return String.Format("Title:{0} Author: {1} Year: {2}- Price:{3}",title,author,year,price);

}

}

class QueueBooks

{

Queue<Book> books;

public QueueBooks()

{

books = new Queue<Book>();

}

public QueueBooks(Book[]book)

{

books = new Queue<Book>();

for (int i = 0; i < book.Length; i++)

{

books.Enqueue(book[i]);

}

}

public void AddBook(Book book) {

books.Enqueue(book);

}

public void AddBookAtPos(Book book, uint pos)

{

Queue<Book> currbooks = new Queue<Book>();

int size = books.Count;

for (uint i = 0; i <size+1; i++)

{

if (i != pos)

currbooks.Enqueue(books.Dequeue());

else

currbooks.Enqueue(book);

}

books = currbooks;

}

public void PrintBooks()

{

Queue<Book> currbooks = new Queue<Book>();

int size = books.Count;

for (int i = 0; i < size; i++)

{

currbooks.Enqueue(books.Peek());

Console.WriteLine(books.Dequeue());

}

books =currbooks;

}

public Book PopBook()

{

Book book = books.Dequeue();

return book;

}

public Book PopBookAtPos(uint pos)

{

Queue<Book> currbooks = new Queue<Book>();

Book book = new Book();

int size = books.Count;

for (uint i = 0; i < size; i++)

{

if(pos!=i)

currbooks.Enqueue(books.Dequeue());

else

book = books.Dequeue();

}

books = currbooks;

return book;

}

}

internal class Program

{

static void Main(string[] args)

{

//4

/\*Book[]BA = new Book[4]

{

new Book("ddddd","ssssss",2015,300),

new Book("aaaaaa","ddddd",2011,500),

new Book("ggggg","xxxxx",1990,1000),

new Book("xxxxx","xxxxx",1870,100000),

};\*/

QueueBooks queueBooks = new QueueBooks(new Book[4]

{

new Book("ddddd","ssssss",2015,300),

new Book("aaaaaa","ddddd",2011,500),

new Book("ggggg","xxxxx",1990,1000),

new Book("xxxxx","xxxxx",1870,100000),

});

queueBooks.PrintBooks();

Console.WriteLine("\_\_\_\_\_\_\_\_\_\_\_\_");

queueBooks.AddBook(new Book("fffffff", "fffffff", 2015, 300));

queueBooks.PrintBooks();

Console.WriteLine("------------");

queueBooks.PopBookAtPos(1);

queueBooks.AddBookAtPos(new Book("fffffff", "fffffff", 2015, 300), 0);

queueBooks.PrintBooks();

}

}

}