TEST DOCUMENT

LibraryItem Class Test

Creates an LibraryItem object that called item and takes the name parameter as "test".

The expected case is assign the "test" as name to the object, set item availability true, give it an unique item id and keep the last item's id.

```
> LibraryItem item= new LibraryItem("test")
```

Calls the object method called getTitle. That method returns the item's name.

The expected output is the name that given to the constructor as "test".

```
> item.getTitle()
"test"
```

Calls the object method called getItemID. This method returns the item's id.

The expected output is the unique item id that assign as "0" in the constructor when the object is created.

```
> item.getItemID()
0
```

Call the object method called isAvailable. This method shows the availability of the item object.

The expected output is the availability value that assign as "true" in the constructor when the object is created.

```
> item.isAvailable()
true
```

Call the object method called borrowItem. This method changes the availability of the item object to "false".

Then calls isAvailable method again to indicate the object's availability.

The expected output is "false".

```
> item.borrowItem()

> item.isAvailable()

false
```

Call the object method called returnItem. This method changes the availability of the item object to "true".

Then calls isAvailable method again to indicate the object's availability.

The expected output is "true".

```
> item.returnItem()

> item.isAvailable()

true
```

Call the object method called toString. This method returns itemId, name and availability as text.

The expected output is "0 test available"

```
> item.toString()

"0 test available"
```

Book Class Test

Creates a Person object that called person. Person constructor takes name, surname, day, month, year parameters and assigns to fields as "sule", "gurbuz", "15", "8", "1978".

Creates a Book object that called book. Book constructor takes book's name, author and isbn parameters and assigns to fields as "kambur", person object and "12345". name parameter sends to the upper class with super method. Gives it an unique item id, assigns the availability "true" and keep the last item's id with upper class constructor.

```
> Person person= new Person("sule","gurbuz",15,8,1978)

> Book book= new Book("kambur", person, "12345")
```

Calls the object method called getAuthor. This method returns author as Person object.

The expected output is "sule gurbuz 15/18/1978".

```
> book.getAuthor()

sule gurbuz 15/8/1978
```

Calls the object method called getISBN. This method returns isbn value as text.

The expected output is “12345”.

```
> book.getISBN()

"12345"
```

Calls the object method called toString. This method calls upper class’es toString method with super method. Also returns author and isbn with super’s toString.

The expected output is “1 kambur available sule gurbuz 15/8/1978 12345”

```
> book.toString()

"1 kambur available sule gurbuz 15/8/1978 12345"
```

Calls the object method called borrowItem from upper class. This method changes the availability of the item object to “false”.

Then calls toString method again to indicate the object’s values.

The expected output is “1 kambur is not available sule gurbuz 15/8/12345”

```
> book.borrowItem()

> book.toString()

"1 kambur is not available sule gurbuz 15/8/1978 12345"
```

Calls the object method called returnItem from upper class. This method changes the availability of the item object to “true”.

Then calls toString method again to indicate the object’s values.

The expected output is “1 kambur available sule gurbuz 15/8/1978 12345”

```
> book.returnItem()

> book.toString()

"1 kambur available sule gurbuz 15/8/1978 12345"
```

Magazine Class Test

Creates a Date object called dateMagazine. Date constructor takes day, month and year parameters and assigns to field as “25”, “10” and “2023”.

Creates a Magazine object called magazine. Magazine constructor takes title, issue number and publication number parameters and assigns to fields as "popular science", "121" and Date object. title parameter sends to the upper class with super method. Gives it an unique item id, assigns the availability "true" and keep the last item's id with upper class constructor.

```
> Date dateMagazine= new Date(25,10,2023)

> Magazine magazine= new Magazine("popular science","121",dateMagazine)
```

Calls the object methods called getIssueNumber. This mehtod returns issue number as text.

The expected output is "121".

```
> magazine.getIssueNumber()

"121"
```

Calls the object methods called getPublicationDate. This method returns magazine publication date as Date object.

The expected output is "25/10/2023".

```
> magazine.getPublicationDate()

25/10/2023
```

Calls the object method called toString. This method calls upper class'es toString method with super method. Returns publication date and issue number with super's toString.

The expected output is "2 popular science available 25/10/2023 121".

```
> magazine.toString()

"2 popular science available 25/10/2023 121"
```

Calls the object method called borrowItem from upper class. This method changes the availability of the item object to "false".

Then calls toString method again to indicate the object's values.

The expected output is "2 populer science is not available 25/10/2023 121"

```
> magazine.borrowItem()
> magazine.toString()

"2 populer science is not available 25/10/2023 121"
```

Calls the object method called returnItem from upper class. This method changes the availability of the item object to “true”.

Then calls toString method again to indicate the object’s values.

The expected output is “2 populer science available 25/10/2023 121”

```
> magazine.returnItem()

> magazine.toString()

"2 populer science available 25/10/2023 121"
```

DVD Class Test

Creates a Person object called dvdPerson. Person constructor takes name, surname, day , month, year parameters and assigns to field as “cem”, “yilmaz” and Date object.

Creates a DVD object called dvd. DVD constructor takes title, director and run time number parameters and assigns to fields as “arog”, Person object and “125”. title parameter sends to the upper class with super method. Gives it an unique item id, assigns the availability “true” and keep the last item’s id with upper class constructor.

```
> Person dvdPerson= new Person("cem","yilmaz",14,10,1972)

> DVD dvd= new DVD("arog",dvdPerson,125)
```

Calls the object method called getDirector. This method returns the director information as Person object.

The expected output is “cem yilmaz 14/10/1972”.

```
> dvd.getDirector()

cem yilmaz 14/10/1972
```

Calls the object method called getRuntimeMInutes. This method returns the runtime as text.

The expected output is “125”.

```
> dvd.getRuntimeMinutes()

125
```

Calls the object method called toString. This method calls upper class’es toString method with super method. Returns publication director information and runtime minutes with super’s toString.

The expected output is “3 arog available cem yilmaz 14/10/1972 125”.

```
> dvd.toString()

"3 arog available cem yilmaz 14/10/1972 125"
```

Calls the object method called borrowItem from upper class. This method changes the availability of the item object to “false”.

Then calls toString method again to indicate the object’s values.

The expected output is “3 arog is not available cem yilmaz 14/10/1972 125”

```
> dvd.borrowItem()

> dvd.toString()

"3 arog is not available cem yilmaz 14/10/1972 125"
```

Calls the object method called returnItem from upper class. This method changes the availability of the item object to “true”.

Then calls toString method again to indicate the object’s values.

The expected output is “3 arog available cem yilmaz 14/10/1972 125”

```
> dvd.returnItem()

> dvd.toString()

"3 arog available cem yilmaz 14/10/1972 125"
```

Person Class Test

Creates a Date object called personDate. Date constructor takes day, month, year parameters and assigns to field as “3”, “5” and “1993”.

Creates a Person object called person. Person constructor takes name, surname and date of birth parameters and assigns to fields as “ali”, “yilmaz” and Date object.

```
> Date personDate=new Date(3,5,1993)

> Person person= new Person("ali","yilmaz",personDate)
```

Calls the object method called getFirstName. This method returns first name of the person as text.

The expected output is “ali”.

```
> person.getFirstName()

"ali"
```

Calls the object method called getLastName. This method returns person’s last name as text.

The expected output is ”yilmaz”.

```
> person.getLastName()

"yilmaz"
```

Calls the object method called getDateOfBirth. This method returns the person’s date of birth as text.

The expected output is “3/5/1993”.

```
> person.getDateOfBirth()

3/5/1993
```

Call the object method that called toString. This method returns first name, last name and date of birth as text.

The expected output is “ali yilmaz 3/5/1993”

```
> person.toString()

"ali yilmaz 3/5/1993"
```

Creates a Person object called person2. Person constructor takes name, surname, day, month, year parameters and assigns to fields as “hasan”, “duru” and Date object. Makes a method overloading for constructor for using different parameters.

```
> Person person2= new Person("hasan","duru",21,7,2005)
```

Calls the object method called getFirstName. This method returns first name of the person as text.

The expected output is “hasan”.

```
> person2.getFirstName()

"hasan"
```

Calls the object method called getLastName. This method returns last name of the person as text.

The expected output is “duru”.

```
> person2.getLastName()

"duru"
```

Calls the object method called getDateOfBirth. This method returns date of birth as text.

The expected output is “21/7/2005”.

```
> person2.getDateOfBirth()

21/7/2005
```

Call the object method that called toString. This method returns first name, last name and date of birth as text.

The expected output is “hasan duru 21/7/2005”

```
> person2.toString()

"hasan duru 21/7/2005"
```

Patron Class Test

Creates a Date object called patronDate. Date constructor takes day, month, year parameters and assigns to field as “11”, “11” and “1984”.

Creates a Patron object called patron. Patron constructor takes name, surname, date of birth and emailAdress parameters and assigns to fields as “buse”, “kara”, Date object and

“busekara@mail.com”. Gives it an unique library card number and keep the last library card number.

```
> Date patronDate= new Date(11,11,1984)

> Patron patron= new Patron("buse","kara",patronDate,"busekara@mail.com")
```

Calls the object method called getLibraryCardNumber. This method returns unique library card number as number.

The expected output is “0”.

```
> patron.getLibraryCardNumber()

0
```

Calls the object method called getEmailAdress. This method returns mail adress as text.

The expected output is “busekara@mail.com”.

```
> patron.getEmailAdress()

"busekara@mail.com"
```

Call the object method called toString. This method returns first name, last name, date of birth, library card number an mail adress as text.

The expected output is “buse kara 11/11/1984 0 busekara@mail.com”

```
> patron.toString()

"buse kara 11/11/1984 0 busekara@mail.com"
```

Librarian Class Test

Creates a Date object called librarianDate. Date constructor takes day , month, year parameters and assigns to field as “3”, “5” and “2002”.

Creates a Librarian object called librarian. Librarian constructor takes name, surname and date of birth parameters and assigns to fields as “zeynep”, “durmaz” and Date object. name, surname and date of birth parameter sends to the upper class with super method. Assigns the parameters to the fields with upper class constructor. Librarian constructor gives it an unique emloyee id, and keep the last employee’s id.

```
> Date librarianDate= new Date(3,5,2002)

> Librarian librarian= new Librarian("zeynep","durmaz",librarianDate)
```

Calls the object method called getEmployeeID. This method returns employee id as number.

The expected output is “0”.

```
> librarian.getEmployeeID()

0
```

Calls the object method called toString. This method calls upper class’es toString method with super method. Returns employee id with super’s toString.

The expected output is “zeynep durmaz 3/5/2002 0”

```
> librarian.toString()

"zeynep durmaz 3/5/2002 0"
```

Date Class Test

Creates a Date object called date. Date constructor takes day, month, year parameters and assigns to field as “12”, “12” and “2012”.

```
> Date date= new Date(12,12,2012)
```

Calls the object method called getDay. This method returns the day value as number.

The expected output is “12”.

```
> date.getDay()

12
```

Calls the object method called getMonth. This method returns the month value as number.

The expected output is “12”.

```
> date.getMonth()

12
```

Calls the object method called getYear. This method returns the year value as number.

The expected output is “2012”.

```
> date.getYear()

2012
```

Call the object method called toString. This method returns day, month and year value as text.

The expected output is “12/12/2012”.

```
> date.toString()

"12/12/2012"
```