



VAAAL UNIVERSITY
OF TECHNOLOGY

Inspiring thought. Shaping talent.

FACULTY: APPLIED AND COMPUTER SCIENCE
ASDSX3A – Development Software 3.1
DUE DATE: 24 March 2023
EXAMINER: MRS S. MOYO

DEPARTMENT: ICT
Assignment-Inheritance
TOTAL: 100 Marks
MODERATOR: MR S. SIYABONGA

The hospital at your locality requires a system to record details about patients and to calculate the amount payable by each patient or their medical aid. You are required to do a prototype of a possible system that could later be developed into a full patient management system.

The hospital has two types of patients: out-patients and in-patients

Out-Patients: These visit the hospital, receive medicine, go home and come back for a follow up visit.

In-Patients: These patients are admitted to the hospital, stay for a number of days either in the intensive care ward, or the normal ward and is then discharged.

Among other details, the hospital wants to store the following information about patients: id number, surname, telephone number/cell, patient number, address (street, town, postal code), number of days spent at hospital, ward patient spent at (intensive care/normal ward), medicine amount, patient type.

- The suggestion is that you should use these 4 classes for the application: **Patient , Address, OutPatient, InPatient** with the following relationships:
 - **A Patient has an Address**
 - **An OutPatient is a Patient**
 - **An InPatient is a Patient**

Address Class

Attributes:

Street, Town, Postal code

Constructor:

- Code the parameterized constructor only

Methods:

- Override the toString method to return all the attributes of the class separated by a semi-colon (;)

Patient Class

Properties:

- PatientNr, Surname, ID, Telephone, Address
- The following properties have rules that you must implement:
- PatientNr must be assigned in class, starting with 1001, and the application (GUI) must not be able to change the value property
- Surname is not allowed to be empty
- ID is not allowed to be empty

Constructors

- Default constructor
- Parameterized constructor with five parameters for Surname, ID, Telephone, Address.

Methods

- Declare an abstract method calculateAmountDue which will return the amount payable by the patient
- Override the toString method to return all the properties separated by a semi-colon (;).

(PatientNr, Surname, ID, Telephone, street, Town, postalCode)

OutPatient Class

Attributes:

The PatientNr, Surname, ID, Telephone, Address must be inherited from the Patient class and the following added:

- MedicineAmount: This attribute will contain the amount payable by the patient for medicine provided. Throw an IllegalArgumentException with an appropriate message if the amount is negative.

Constructors:

- Default constructor with no parameters
- Parametrized constructor with five parameters for Surname, ID, Telephone, Address and medicine amount.

Methods:

- Override the calculateAmountDue method inherited from the patient class .The amount due by out-patients calculated by adding a consultation fee of r120 to the medicine amount.
- Override the toString() method to return the string: Out Patient, followed by all the properties separated by a semi-colon (;). (Out Patient; PatientNr, Surname, ID, Telephone, street, Town, postalCode;

InPatient Class

Attributes:

The PatientNr, Surname, ID, Telephone, Address must be inherited from the Patient class and the following added:

- NumberofDays: This property must contain the number of days that the patient has spent in the hospital. Throw an IllegalArgumentException with an appropriate message if the number of days is 0 or less.
- IntensiveCare: The property must be set to true if the patient spent the days in the intensive care ward and false if the days were spent in a normal ward.

Constructors:

- Default constructor with no parameters
- Parametrized constructor with 6 parameters for Surname, ID, Telephone, Address number of days and intensive care.

Methods:

- Override the **calculateAmountDue** method inherited from the patient class. The amount due by in-patients is calculated by multiplying the number of days R1500 if the patient spent the days in intensive care, and multiplying by the number of days by R1000 if the patient spent the days in a normal ward.
- Override the toString() method to return the string: "In Patient", followed by all the properties separated by a semi-colon (;). (In Patient; PatientNr, Surname, ID, Telephone, street, Town, postalCode; numberofDays; IntensiveCare)

Part 2: GUI

The GUI for the application consists of one screen as shown in **Figure 1**:

- Panels have been used to group associated controls. **Combo boxes** have been used for town and medicine amount input entry.

The GUI consists of the following elements:

- General** panel:
 - Surname:
 - ID:
 - Telephone:
- Address** panel:
 - Street:
 - Town:
 - Postal code:
- Type of Patient** panel:
 - ☐ Out Patient
 - ☐ In Patient
- Out Patient** panel:
 - Medicine amount:
- In Patient** panel:
 - ☒ Intensive Care
 - Days spent:
- A large empty rectangular box for additional information or notes.
- A **Create Patient and Display** button at the bottom.

Figure 1

- When the form is first displayed, the **Out Patient** and **In Patient** panels are hidden/not visible (**Figure 2**).

Patient Management App (Vosloorus Hospital)

General

Surname

ID

Telephone

Address

Street

Town

Postal code

Type of Patient

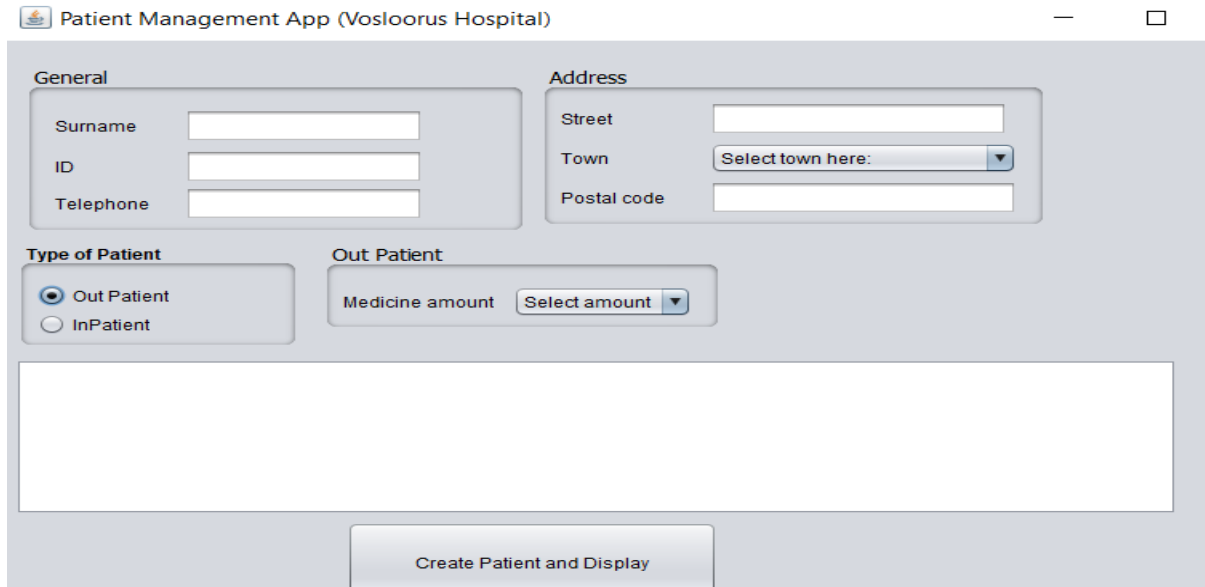
☐ Out Patient

☐ In Patient

Create Patient and Display

Figure 2

- When the **Out Patient radio** is selected the out Patient panel is displayed (**Figure 3**) and when **the Inpatient radio** button is selected, the In Patient panel is displayed (**Figure 4**) respectively.



The screenshot displays the 'Patient Management App (Vosloorus Hospital)' window. The interface is divided into several sections: 'General' with input fields for Surname, ID, and Telephone; 'Address' with input fields for Street, Town (a dropdown menu showing 'Select town here:'), and Postal code; 'Type of Patient' with two radio buttons, 'Out Patient' (which is selected) and 'InPatient'; and an 'Out Patient' section with a 'Medicine amount' dropdown menu showing 'Select amount'. A large empty text area is located below these sections. At the bottom center, there is a button labeled 'Create Patient and Display'.

Figure 3

Patient Management App (Vosloorus Hospital)

General

Surname

ID

Telephone

Address

Street

Town

Postal code

Type of Patient

☐ Out Patient

☒ InPatient

In Patient

☒ Intensive Care

Days spent

Create Patient and Display

Figure 4

- The **Create Patient** button creates a patient object and displays it in the text area (see **Figure 5**)

Patient Management App (Vosloorus Hospital)

General

Surname:

ID:

Telephone:

Address

Street:

Town:

Postal code:

Type of Patient

☐ Out Patient

☒ InPatient

In Patient

☐ Intensive Care

Days spent:

Out Patient; 1001;Nox;9810271234567;0781027123;12 Vilakazi st;Devon;1888;;\$350.00
In Patient; 1002;Nelisiwe;9610271234567;0681027123;45 Shaka st;Heidelberg;1491;;5,true;\$7,500.00
In Patient; 1003;Lerato;9910271234567;0881027123;4 Sibiya st;Duduza;1490;;8,false;\$8,000.00

Create Patient and Display

*****The End *****