Bahria University,

Karachi Campus



COURSE: SEL-221 SOFTWARE DESIGN & ARCHITECTURE

TERM: SPRING 2022, CLASS: BSE- 4 (B)

Submitted By:

**M Muaz Shahzad 02-131202-081**

(Name) (Reg. No.)

Submitted To:

Engr. Majid Kaleem/ Engr. Saniya Sarim

Signed Remarks: Score:

INDEX

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SNO | DATE | LAB NO | LAB OBJECTIVE | SIGN |
| 01 | 9-3-22 | 01 | Intro to Rational Rose |  |
| 02 | 16-3-22 | 02 | Forward & Reverse Engineering |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Bahria University,

Karachi Campus



LAB EXPERIMENT NO.

**01**

LIST OF TASKS

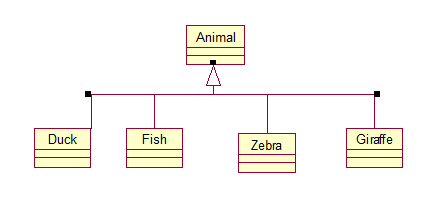
|  |  |
| --- | --- |
| TASK NO | OBJECTIVE |
| **1** | **Create class diagram with 5 classes and apply forward engineering approaches to generate code.** |
| **2** | **From forward engineering update your Generated java code and add attributes and operations in your classes.** |
| **3** | **Reverse Engineer your edited code in task2 and display updated UML diagram.** |
|  |  |

Submitted On:

29-march-2022

**Task 1: Create class diagram with 5 classes and apply forward engineering approaches to generate code.**

**Solution:**



**Output:**

//Source file: C:\\JP1\\Animal.java

public class Animals

{

/\*\*

@roseuid 5AA037D9004E

\*/

public Animals() {

}

/\*\*

@roseuid 5AA0379F008D

\*/}

private class Duck extends Animals {

/\*\*

@roseuid 5AA0379F009D

\*/

public Duck() {

}}

private class Fish extends Animals

{ /\*\*

@roseuid 5AA0379F00AC

\*/

public Fish() {

}}

private class Giraffe extends Animals

{ /\*\*

@roseuid 5AA0379F00BC

\*/

public Giraffe() {

}}

private class Zebra extends Animals {

/\*\*

@roseuid 5AA0379F00CB

\*/

public Zebra() {

}}

**Task 2: From forward engineering update your Generated java code and add attributes and operations in your classes.**

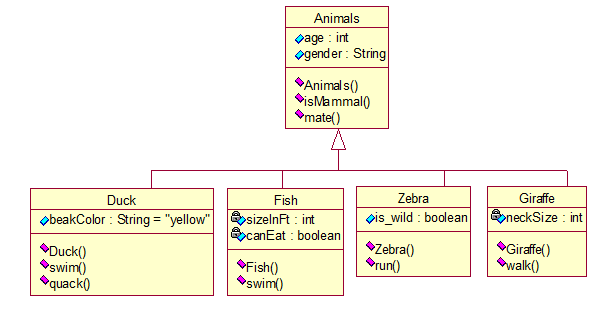
**Solution:**

//Source file: C:\\JP1\\Animal.java  
public class Animals {  
 public int age;  
 public String gender;  
 public void isMammal(){}  
 public void mate(){}  
 /\*\*  
 @roseuid 5AA037D9004E  
 \*/  
 public Animals()   
 { } /\*\*  
@roseuid 5AA0379F008D  
 \*/}  
private class Duck extends Animals   
{ public String beakColor = "yellow";  
 public void swim(){}  
 public void quack(){}  
 /\*\*  
 @roseuid 5AA0379F009D  
 \*/  
 public Duck()   
 { }}  
private class Fish extends Animals   
{  
 private int sizeInFt;  
 private boolean canEat;  
 public void swim(){}  
 /\*\*  
 @roseuid 5AA0379F00AC  
 \*/  
 public Fish() {  
}}  
private class Giraffe extends Animals   
{  
 private int neckSize;   
 public void walk(){}  
 /\*\*  
 @roseuid 5AA0379F00BC  
 \*/

public Giraffe()   
 { }  
}  
private class Zebra extends Animals   
{  
 public boolean is\_wild;  
 public void run(){}  
 /\*\*  
 @roseuid 5AA0379F00CB  
 \*/  
 public Zebra() {  
}}

**Task 3: Reverse Engineer your edited code in task2 and display updated UML diagram.**

**Output:**



Bahria University,

Karachi Campus

A picture containing text, room

Description automatically generated

LAB EXPERIMENT NO.

**02**

LIST OF TASKS

|  |  |
| --- | --- |
| TASK NO | OBJECTIVE |
| **1** | **Create class diagram of University Registration and School (shown in next slide) using that diagram apply concept of forward engineering in Rational Rose.** |
| **2** | **Apply concept of reverse engineering to create the class diagram using java code in Rational Rose for both diagrams given in next slide** |
|  |  |

Submitted On:

29-March-2022

**Task 1: Create class diagram of University Registration and School (shown in next slide) using that diagram apply concept of forward engineering in Rational Rose.**

**Solution:**

Diagram

Description automatically generated**University Registration:**

Diagram

Description automatically generated **School:**

**Task 2: Apply concept of reverse engineering to create the class diagram using java code in Rational Rose for both diagrams given in next slide**

**University Registration:**

//Source file: C:\\Moasfar\\Registration.java

private class Account {

public Registration theRegistration;

public Student theStudent;

private String StudentID;

void CreateAccount(){}

void Account(){} }

private class Course {

private String CourseCode;

private String Date;

private String Instructor;

void CreateCourse(){}

void DropCourse(){}

private void Course(){}}

private class Professor {

private int StaffID;

private String Department;

void CreateProfessor(){}

void getProfessorInfo(){}

private void Professor(){}}

public class Registration {

public Course theCourse;

public Professor theProfessor;

private String AccountID;

private String CoursCode;

private String ExamID;

private String grade;

void getRegistrationNo(){}

/\*\*

@roseuid 5AA04582001F

\*/

public Registration()

{

}

}

private class Student

{

private String StudentID;

private String Name;

private String DateOfBirth;

private String Email;

private String ContactNo;

void CreateStudent(){}

void getStudentInfo(){}

private void Student(){}

}

**School:**

//Source file: C:\\Moasfar\\School.java

private class Course {

private String name;

private int courseID;

/\*\*

@roseuid 5AA039B1032C

\*/

public Course()

{

}}

private class Department extends School {

public Instructor theInstructor;

public Course theCourse;

private String name;

void addInstructor(){}

void removeInsruction(){}

void getInstructor(){}

void getAllInstructors(){}

/\*\*

@roseuid 5AA039B1034C

\*/

public Department()

{

}}

public class School {

private String name;

private String address;

private int phone;

void addStudent(){}

void removeStudent(){}

void getStudent(){}

void addDepartment(){}

void removeDepartment(){}

void getDepartment(){}

void getAllDepartments(){}

/\*\*

@roseuid 5AA039B1036B

\*/

public School()

{

}}

private class Instructor extends Department{

public Course theCourse;

private String name;}

private class Student extends School {

public Course theCourse;

private String name;

private int studentID;

/\*\*

@roseuid 5AA039B1037B

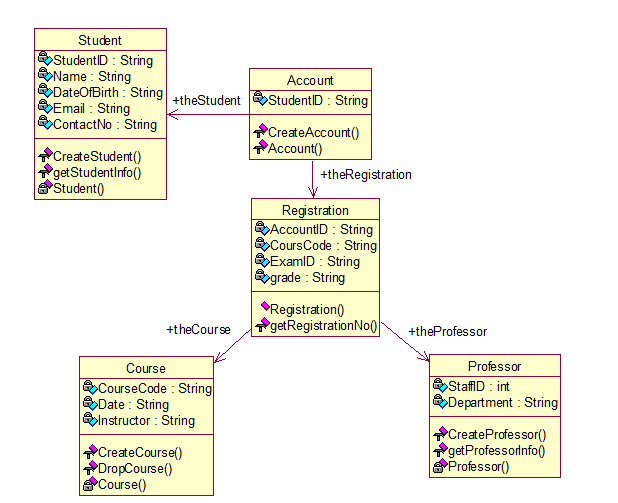
\*/

public Student()

{ }}

**Reverse Engineering:**

**University Registration:**



**School:**

Diagram

Description automatically generated