

Object Oriented Programming

10608881



04 August 2023

Practical Programming

A diagram of a student

Description automatically generated

UML EXPLAINATION

**Form1 class:**

This class represents the entry point of the GUI application. It is a Windows Forms class responsible for creating the main user interface of the application.

The class contains a private attribute dataGridView, which is a DataGridView control. This control is used to display data retrieved from the database.

The Form1 class will have methods to interact with the database and retrieve data to populate the dataGridView.

**Person class:**

This is an abstract class that serves as the base class for both students and lecturers. It contains common attributes and methods that are shared by both types of people.

The attributes of the Person class are Name, Address, County, Age, Phone, and Email, all of which are represented as private attributes.

The Person class has a parameterless constructor Person() and a constructor Person(...params) that takes parameters to initialize the attributes.

The class provides getter methods such as GetName(), GetAddress(), GetCounty(), GetAge(), GetPhone(), and GetEmail() to access the attribute values.

The class also provides setter methods such as SetName(name: string), SetAddress(address: string), SetCounty(county: string), SetAge(age: int), SetPhone(phone: string), and SetEmail(email: string) to modify the attribute values.

**Student class:**

This class represents a student and inherits from the Person class. It extends the Person class by adding a specific attribute StudentNumber, which is represented as a private attribute.

The Student class has a parameterless constructor Student() and a constructor Student(...params) that takes parameters to initialize the attributes of both the Person base class and the StudentNumber.

The class provides a getter method GetStudentNumber() and a setter method SetStudentNumber(number: int) to access and modify the StudentNumber attribute.

**Lecturer class:**

This class represents a lecturer and also inherits from the Person class. It extends the Person class by adding a specific attribute Pay, which is represented as a private attribute.

The Lecturer class has a parameterless constructor Lecturer() and a constructor Lecturer(...params) that takes parameters to initialize the attributes of both the Person base class and the Pay.

The class provides a getter method GetPay() and a setter method SetPay(pay: decimal) to access and modify the Pay attribute.

**Database class:**

This class represents the database used by the application. It is responsible for managing the database connection, executing queries, and interacting with the database tables.

The Database class contains a private attribute ConnectionString, which is a string representing the connection details for the database.

The class has methods like ExecuteQuery(query: string, params) and ExecuteNonQuery(query: string, params) to execute SQL queries and stored procedures on the database and return the result as a DataTable or an integer (for non-query operations).

Additionally, the Database class provides methods to add students and lecturers to the database (AddStudent(studentData: Student) and AddLecturer(lecturerData: Lecturer)).

It also includes methods to retrieve data for various queries, such as GetAllStudents(), GetAllLecturers(), GetMaleLecturers(), GetStudentsAboveAge(age: int), and two other custom queries (GetOtherQuery1() and GetOtherQuery2()).