

Mind Teaser

On New Year's Day, the entire Smith family decided to get together for lunch. The family members present consisted of THREE grandchildren, FOUR children, TWO mothers, TWO fathers, ONE grandfather, ONE grandmother, TWO sisters, ONE brother, TWO daughters, TWO sons, ONE daughter-in-law, ONE mother-in-law and ONE father-in-law. How many people were present for the lunch that day?

How many classes are there?

Are you going to use a Class Diagram to depict?



Wk 8)

Study the following description of a Work Tracking Application (WTA):

- WTA is a system that maintains a centralized repository of all projects and tasks assigned to staff in an organisation. It contains the project and task details, including the tracking of issues. WTA allows the easy access of relevant information on the staff and projects as well as tracking a project's progress and staff's utilisation rate. A staff can play different roles in projects.
- A project has an identification number, a description, a project manager, the project awarded price and the customer. A project which is more than 1 million dollar will require also a project director. A project is broken down into multiple tasks which are assigned to the same or different staff. Each task will have a project leader, its team members, a task identification, task description and an expected date of completion. Project issues are tracked for every task in the project by assigning an identification number and providing details like title, detailed description, status, reported date and last updated date. A project is funded externally by a customer which the system will store the company name, company registration number, contact person and number

- WTA also stores the staff details like identification number, name, assigned projects and tasks, skillsets and man-hour rate. Staff will be assigned to ONE to TWO projects but multiple tasks within the same project depending on their skillsets. Skillsets like certification name, the level, awarded date and date of expiry are stored. Every month, each staff is required to clock his/her monthly utilisation details by recording the time spent on the assigned projects and tasks. Details such as the month and year, period of work (from date and to date) and type of work performed. The type of work can be administrative, project, on leave, training and mentoring.

You are tasked to identify the entity classes needed to build the application based on the description above.

Show your design in a Class Diagram. Your Class Diagram should show clearly the relationships between classes, enumeration, relevant attributes (at least TWO), logical multiplicities, meaningful role names, association names and constraint(s), if any. You need not show the class methods.

Steps:

1. Entity classes: Highlight the keywords of classes (Nouns)

- Those Nouns which have been repeated for a few times
- Those Nouns which have detailed description/definition

2. Relationship: Highlight the keywords of relationships (Verbs)

- The verbs indicate "is part of" relationship
- The verbs indicate "is a special" relationship
- The verbs indicate normal association relationship

3. Attribute: Those nouns in description/definition of Entity classes

4. Role name: Those nouns which are instances of some entity classes in description/definition of Entity classes

5. Multiplicity: Highlight the words related to quantity

- a
- One to two
- s. e.g., team members

What are the nouns, which have been repeated for a few times in following paragraph?

51 Response
Using Click

WTA is a system that maintains a centralized repository of all projects and tasks assigned to staff in an organisation. It contains the project and task details, including the tracking of issues. WTA allows the easy access of relevant information on the staff and projects as well as tracking a project's progress and staff's utilisation rate. A staff can play different roles in projects.

- A. System, repository, project, staff
- B. Project, staff
- C. Project, task, staff
- D. WTA, Project, task, staff

C

Se

What are the nouns, which have detailed description/definition in following paragraph, but not in the previous list?

31

Response

Writing Spec.

A project has an identification number, a description, a project manager, the project awarded price and the customer. A project which is more than 1 million dollar will require also a project director. A project is broken down into multiple tasks which are assigned to the same or different staff. Each task will have a project leader, its team members, a task identification, task description and an expected date of completion. Project issues are tracked for every task in the project by assigning an identification number and providing details like title, detailed description, status, reported date and last updated date. A project is funded externally by a customer which the system will store the company name, company registration number, contact person and number.

- A. ProjectIssue, customer, description
- B. ProjectIssue, customer
- C. ProjectIssue, customer, company
- D. Project, ProjectIssue, customer

b

Se

What are the nouns, which have detailed description/definition in following paragraph, but not in the previous lists?

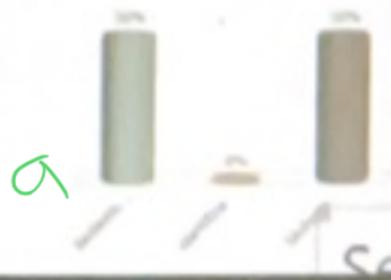
3

Response
12

Using Word

WTA also stores the staff details like identification number, name, assigned projects and tasks, skillsets and man-hour rate. Staff will be assigned to ONE to TWO projects but multiple tasks within the same project depending on their skillsets. Skillsets like certification name, the level, awarded date and date of expiry are stored. Every month, each staff is required to clock his/her monthly utilisation details by recording the time spent on the assigned projects and tasks. Details such as the month and year, period of work (from date and to date) and type of work performed. The type of work can be administrative, project, on leave, training and mentoring.

- A. Skillsets, monthlyutilisation, typeofwork
- B. identificationNumber, typeofwork
- C. Skillsets, monthlyutilisation



3. (a) Study the following description of a Work Tracking Application (WTA):

2 WTA is a system that maintains a centralized repository of all projects and tasks assigned to staff in an organisation. It contains the project and task details, including the tracking of issues. WTA allows the easy access of relevant information on the staff and projects as well as tracking a project's progress and staff's utilisation rate. A staff can play different roles in project.

role

1 A project has an identification number, a description, a project manager, the project awarded price and the customer. A project which is more than 1 million dollar will require also a project director. A project is broken down into multiple tasks which are assigned to the same or different staff. Each task will have a project leader, its team members, a task identification, task description and an expected date of completion. Project issues are tracked for every task in the project by assigning an identification number and providing details like title, detailed description, status, reported date and last updated date. A project is funded externally by a customer, which the system will store the company name, company registration number, contact person and number.



3

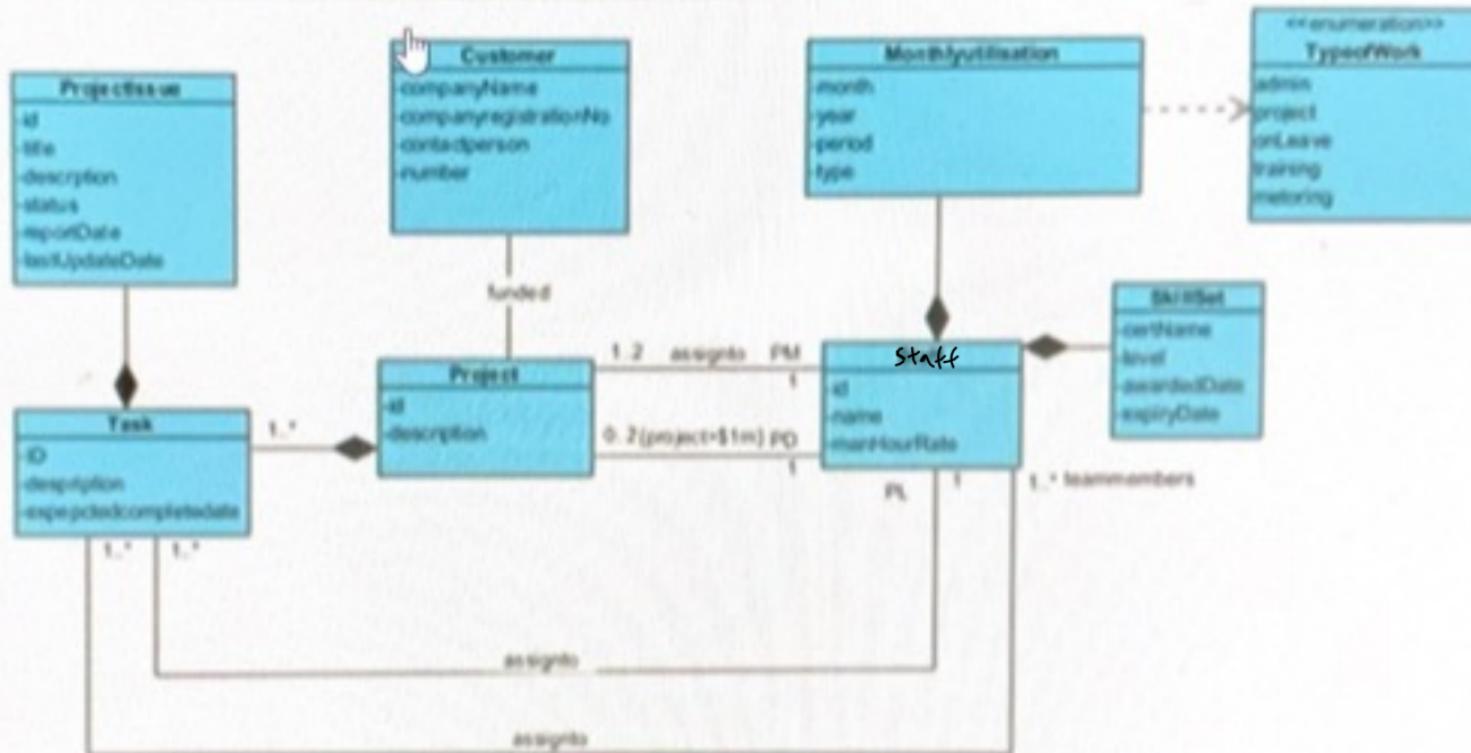
WTA also stores the staff details like identification number, name, assigned projects and tasks, skillsets and man-hour rate. Staff will be assigned to ONE to TWO projects but multiple tasks within the same project depending on their skillsets. Skillsets like certification name, the level, awarded date and date of expiry are stored. Every month, each staff is required to clock his/her monthly utilisation details by recording the time spent on the assigned projects and tasks. Details such as the month and year, period of work (from date and to date) and type of work performed.

8

The type of work can be administrative, project, on leave, training and mentoring.

enumeration

See



Correct classes : 5 marks (e.g. Staff, Project, Task, Issue, Utilisation)

WorkType as enum 1 mark

Constraints for Project cost > 1 need to have Proj Description 1 mark

Correct use of association, Composition 2 marks

Attributes: 1 marks (no private :1)

Correct logic: Multiplicity 2 marks [0..* / 1..*] - show 0..2

Associations name : 1 mark

exam:
method
not,
require

```
public class Project {  
  
    Staff PD;  
    Staff PM;  
    private int id;  
    private int description;  
  
}
```

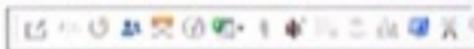
```
public class Staff {  
  
    private int id;  
    private int name;  
    private int manHourRate;  
  
}
```

```
import java.util.*;  
  
public class Task {  
  
    Staff PL;  
    Collection<Staff> teammembers;  
    private int ID;  
    private int description;  
    private int expectedcompletedate;  
  
}
```

```
public class ProjectIssue {  
  
    private int id;  
    private int title;  
    private int description;  
    private int status;  
    private int reportDate;  
    private int lastUpdateDate;  
  
}
```

```
class Book {  
    String name;  
    String authorName;  
    int year;  
    int price;  
    String isbn;  
  
    public Book(String name, String authorName, int year, int price, String isbn) {  
        this.name = name;  
        this.authorName = authorName;  
        this.year = year;  
        this.price = price;  
        this.isbn = isbn;  
    }  
}
```

```
public class Invoice {  
  
    private Book book;  
    private int quantity;  
    private double discountRate;  
    private double taxRate;  
    private double total;  
  
    public Invoice(Book book, int quantity, double discountRate, double taxRate) {  
        this.book = book;  
        this.quantity = quantity;  
        this.discountRate = discountRate;  
        this.taxRate = taxRate;  
        this.total = this.calculateTotal();  
    }  
  
    public double calculateTotal() {  
        double price = ((book.price - book.price * discountRate) * this.quantity);  
  
        double priceWithTaxes = price * (1 + taxRate);  
  
        return priceWithTaxes;  
    }  
  
    public void printInvoice() {  
        System.out.println(quantity + " x " + book.name + " " + book.price + "$");  
        System.out.println("Discount Rate: " + discountRate);  
        System.out.println("Tax Rate: " + taxRate);  
        System.out.println("Total: " + total);  
    }  
  
    public void saveToFile(String filename) {  
        // Creates a file with given name and writes the invoice  
    }  
}
```



- A. The printing format changes
- B. The invoice will be saved into database instead of a file
- C. The formula to calculate discount changes
- D. None of above
- E. All of above

Violates
SRP
Single responsibility
principle



```
public class InvoicePrinter {
    private Invoice invoice;

    public InvoicePrinter(Invoice invoice) {
        this.invoice = invoice;
    }

    public void print() {
        System.out.println(invoice.quantity + "x " +
invoice.book.name + " " + invoice.book.price + " $");
        System.out.println("Discount Rate: " +
invoice.discountRate);
        System.out.println("Tax Rate: " + invoice.taxRate);
        System.out.println("Total: " + invoice.total + " $");
    }
}
```

Session ID cx2002

```
public interface CalculatorOperation {}
```

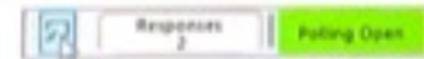
```
public class Addition implements  
CalculatorOperation {  
    private double left;  
    private double right;  
    private double result = 0.0;  
  
    public Addition(double left, double right) {  
        this.left = left;  
        this.right = right;  
    }  
  
    // getters and setters  
}
```

```
public class Subtraction implements  
CalculatorOperation {  
    private double left;  
    private double right;  
    private double result = 0.0;  
  
    public Subtraction(double left, double right) {  
        this.left = left;  
        this.right = right;  
    }  
  
    // getters and setters  
}
```

Session ID cx2002

```
public class Calculator {  
    public void calculate(CalculatorOperation operation) {  
        if (operation == null) {  
            throw new InvalidParameterException("Can not perform operation");  
        }  
  
        if (operation instanceof Addition) {  
            Addition addition = (Addition) operation;  
            addition.setResult(addition.getLeft() + addition.getRight());  
        } else if (operation instanceof Subtraction) {  
            Subtraction subtraction = (Subtraction) operation;  
            subtraction.setResult(subtraction.getLeft() - subtraction.getRight());  
        }  
    }  
}
```

The above classes



- A. look fine.
- B. violates Single Responsibility Principle (SRP)
- C. violates Open-Closed Principle (OCP)

✓
If we need
to add multiplication
or division then need
modify calculator

```
public interface CalculatorOperation {  
    void perform();  
}
```

```
public class Addition implements  
CalculatorOperation {  
    private double left;  
    private double right;  
    private double result;  
  
    // constructor, getters and setters  
  
    @Override  
    public void perform() {  
        result = left + right;  
    }  
}
```

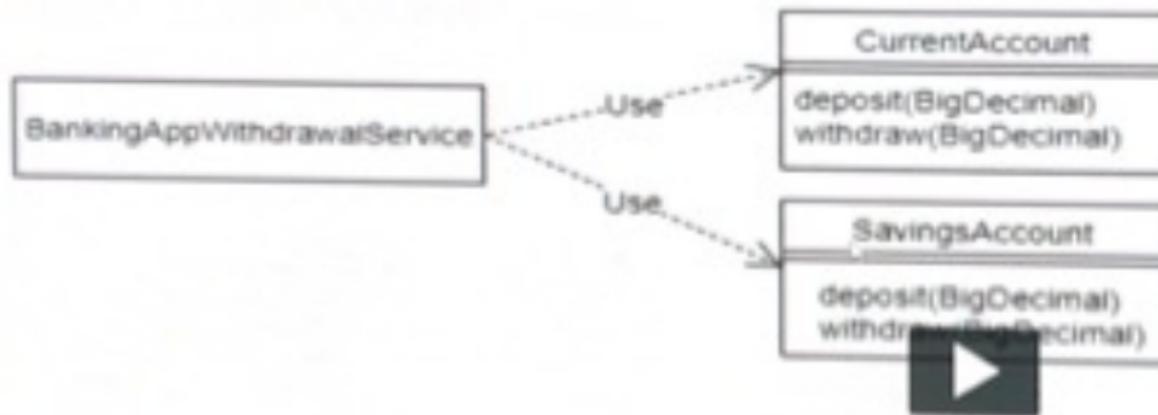
```
public class Division implements CalculatorOperation {  
    private double left;  
    private double right;  
    private double result;  
  
    // constructor, getters and setters  
    @Override  
    public void perform() {  
        if (right != 0) {  
            result = left / right;  
        }  
    }  
}
```

```
public class Calculator {  
  
    public void calculate(CalculatorOperation operation) {  
        if (operation == null) {  
            throw new InvalidParameterException("Cannot perform  
operation");  
        }  
        operation.perform();  
    }  
}
```

Polymorphism

Our Calculator class doesn't need to implement new logic as we introduce new operators. That way the class is closed for modification but open for an ext

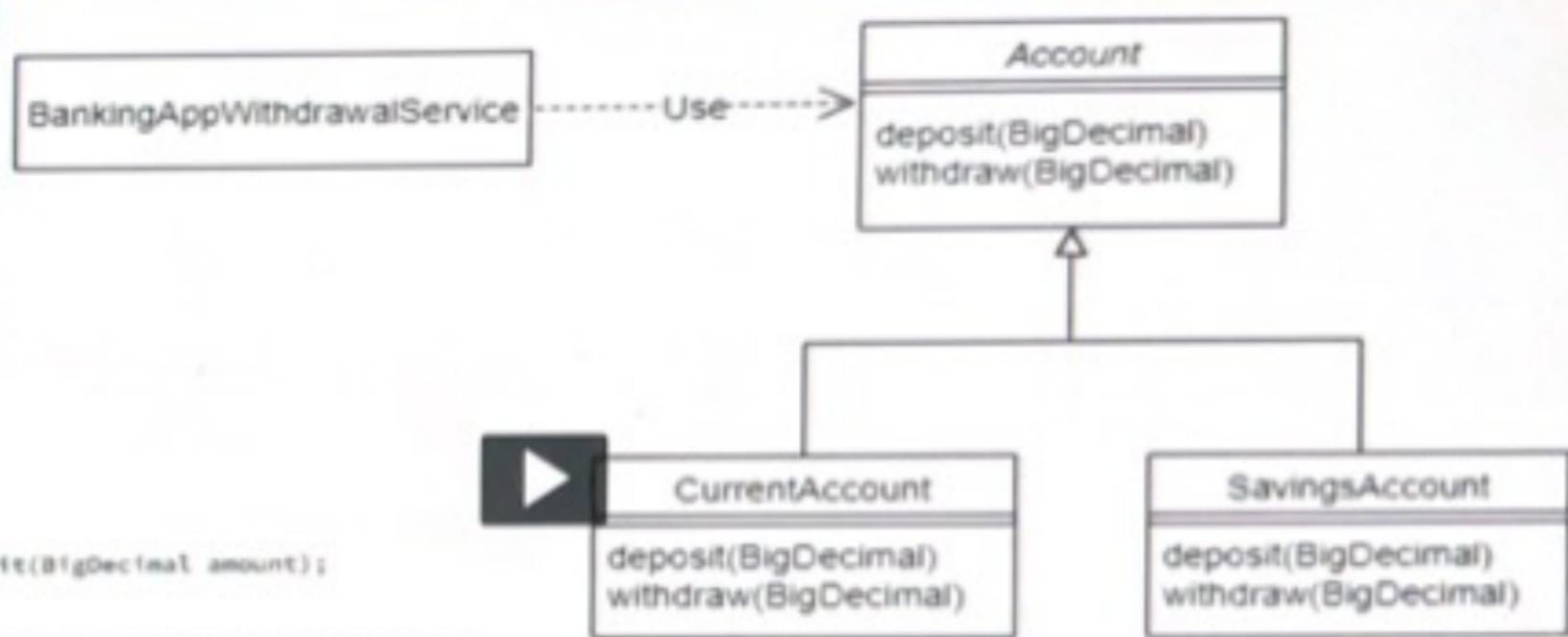
The following design



- A. looks fine.
- B. violates Single Responsibility Principle (SRP)
- C. violates Open-Closed Principle (OCP)
- D. violates Liskov Substitution Principle (LSP)

if add new type of acc^t ✓

```
public abstract class Account {  
    protected abstract void deposit(BigDecimal amount);  
  
    /**  
     * Reduces the balance of the account by the specified amount  
     * provided given amount > 0 and account meets minimum available  
     * balance criteria.  
     *  
     * @param amount  
     */  
    protected abstract void withdraw(BigDecimal amount);  
}
```



The above design

if we add a type of account FixedTermDepositAccount that cannot withdraw but can deposit

- A. look fine.
- B. violates Single Responsibility Principle (SRP)
- C. violates Open-Closed Principle (OCP)
- D. violates Liskov Substitution Principle (LSP)



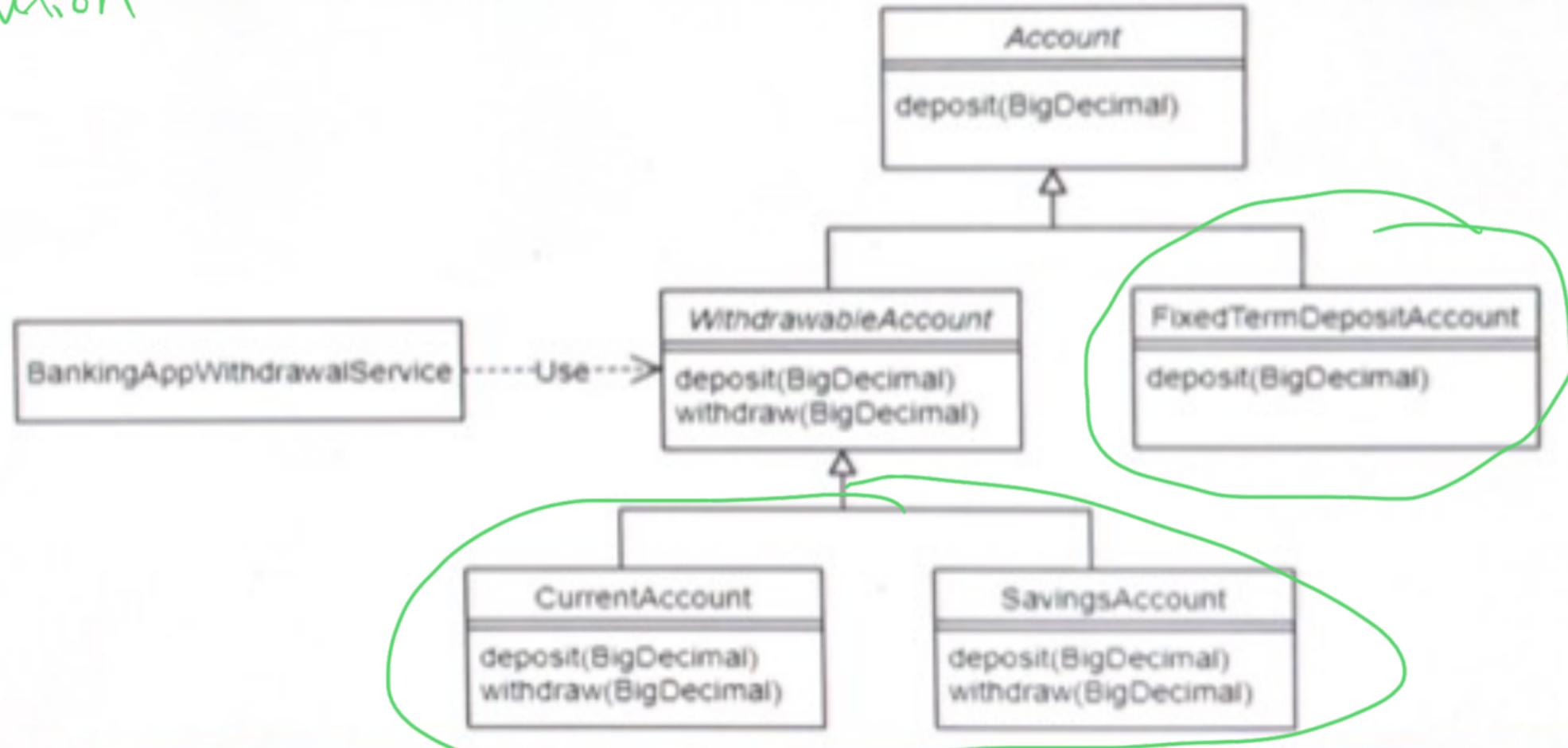
```
public class FixedTermDepositAccount extends Account {  
    @Override  
    protected void deposit(BigDecimal amount) {  
        // Deposit into this account  
    }  
  
    @Override  
    protected void withdraw(BigDecimal amount) {  
        throw new UnsupportedOperationException("Withdrawals are not supported by FixedTermDepositAccount!");  
    }  
}
```

```
Account myFixedTermDepositAccount = new  
FixedTermDepositAccount();  
myFixedTermDepositAccount.deposit(new BigDecimal(1000.00));  
BankingAppWithdrawalService withdrawalService = new  
BankingAppWithdrawalService(myFixedTermDepositAccount);  
withdrawalService.withdraw(new BigDecimal(100.00));
```

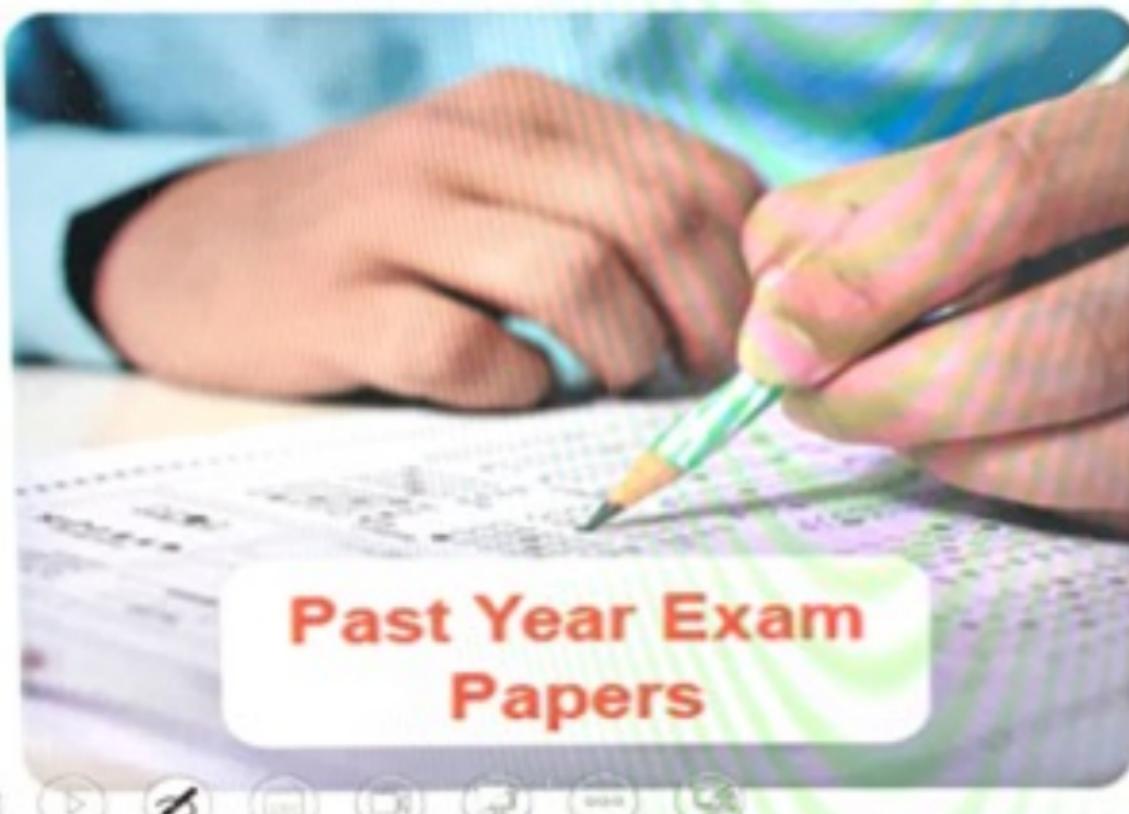


the banking application crashes with the
error:Withdrawals are not supported by

Solution



PYP revision



2017-18 S1

*diagram
below*

4(b) The UML Class Diagram in Appendix E shows the relationship between FOUR classes and snippet code of some of the methods implementation. Both SavingAccount and InvestmentAccount classes have a reference to the DataPresenter object in order to know the type of format (XML or HTML). DataPresenter class also has a reference each to the SavingAccount and InvestmentAccount objects in order to read their content and present the format accordingly.

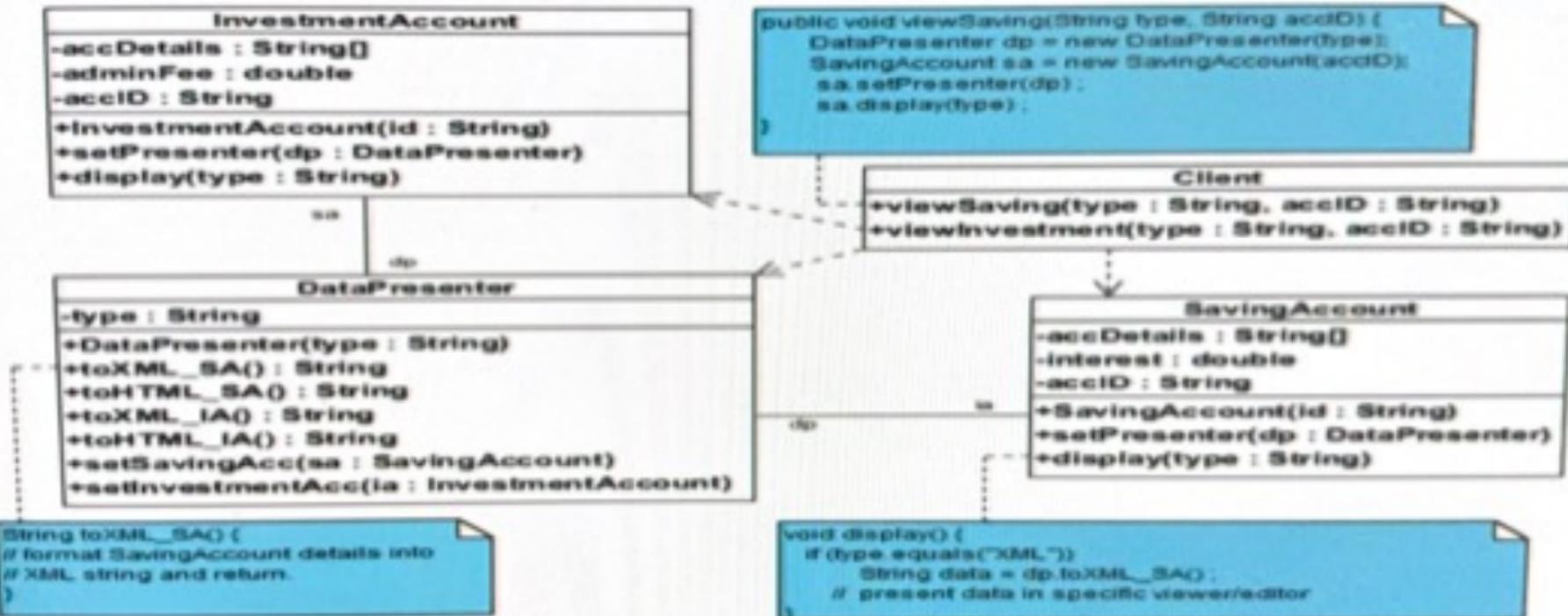
It is suggested that a new RetirementAccount class will be added and the data should be presented also in JSON format.

- (i) State and explain ONE design issue of the current design.

(3 marks)

- (ii) Suggest and explain, with a Class Diagram, how you can improve the current design to cater to the new requirements in your design with reusability, extensibility and maintainability in mind. State the design principle(s) you have applied. **You should show the class/interface method(s) to illustrate your idea.**

(7 marks)



- Violates SFP, Cluster like investment in saving have too much function leading to a lot of modification needed
- Violates OCP, cannot extend easily
-

Exam : 10 Marks

4bi Q: State and explain ONE design issue of the current design.

(3 marks)

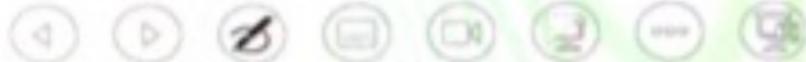
Ans: 4b (i) (3 marks)

Any of Bad encapsulation, Tight coupling and Low Cohesion

OCP: cannot extend easily. Adding new account type has big impact of change

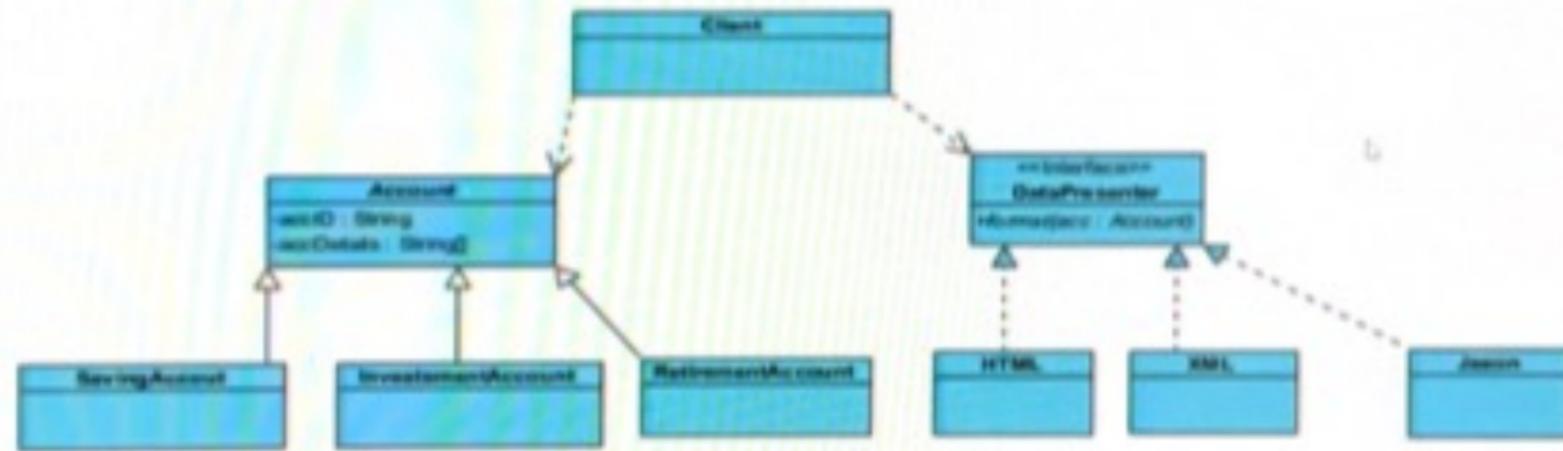
SRP: Dataprestenter has multiple responsibilities

DIP: XML , HTML, JSON might have different version, so should depend on Interface instead of Dataprestenter concrete class.



4bii Q: Suggest and explain, with a Class Diagram, how you can improve the current design to cater to the new requirements in your design with reusability, extensibility and maintainability in mind. State the design principle(s) you have applied. **You should show the class/interface method(s) to illustrate your idea.**

(7 marks)



4 marks

- have an Account abstract/concrete class
- each saving, investment and retirement account are subclasses
- have a DataPresenter interface with a `format(Account)`/`display(..)` method
- each XML, HTML, JSON classes will override `format(Account)`/`display(..)` method
- explore the benefit of client class