**3-nd assignment, Makhmutov Eldar, Balkiyaeva Amira**

**(Please turn on viewer mode to web-site for better experience, right bottom corner in Word-App)**

**Github: https://github.com/Sintelaris/adv-assignment-3**

**1 task.**

**Account class (header):**

class Account: public I\_Printable{  
protected:  
 string name;  
 double balance, dollars, euros, tengeler;  
  
public:  
 Account();  
 Account(string name, double dollars, double euros, double tenge);  
  
 virtual void deposit(double add\_money) = 0;  
  
 virtual void withdraw(double withdraw\_number) = 0;  
  
 virtual void deposit(double withdraw\_number, int choice) = 0;  
  
 void recalculate\_balance();  
  
 void printBalance();  
  
 void setName(const string &name);  
  
 void setBalance(double balance);  
  
 void setDollars(double dollars);  
  
 void setEuros(double euros);  
  
 void setTengeler(double tengeler);  
};

**1) SavingsAccount class (header), which inherits from Account:**

class SavingsAccount: public Account {  
protected:  
 double interest\_rate\_dollar, interest\_rate\_euro, interest\_rate\_tenge;  
  
public:  
 SavingsAccount();  
 SavingsAccount(string name, double dollars, double euros, double tenge, double interest\_rate\_dollar, double interest\_rate\_euro, double interest\_rate\_tenge);  
 virtual void deposit(double add\_money) override;  
 virtual void deposit(double add\_money, int choice) override;  
 virtual void withdraw(double withdraw\_number) override;  
 void printBalance() override;  
  
 void setInterestRateDollar(double interestRateDollar);  
  
 void setInterestRateEuro(double interestRateEuro);  
  
 void setInterestRateTenge(double interestRateTenge);  
  
};

**2) CheckingAccount class (header) has been added to the hierarchy:**

class CheckingAccount: public Account {  
  
 protected:  
 double fee = 1.5;  
 double interest\_rate\_dollar, interest\_rate\_euro, interest\_rate\_tenge;  
 public:  
 CheckingAccount();  
 CheckingAccount(string name, double dollars, double euros, double tenge);  
  
 virtual void deposit(double add\_money) override;  
 virtual void withdraw(double withdraw\_number) override;  
 virtual void deposit(double add\_money, int choice) override;  
 void printBalance() override;  
};

**3) TrustAccount class (header) has been added to the hierarchy:**

class TrustAccount: public SavingsAccount{  
public:  
 TrustAccount();  
 TrustAccount(string name, double dollars, double euros, double tengeler, double interest\_rate\_dollar, double interest\_rate\_euro, double interest\_rate\_tenge);  
 virtual void deposit(double add\_money) override;  
 virtual void deposit(double withdraw\_number, int choice) override;  
 virtual void withdraw(double withdraw\_number) override;  
 void printBalance() override;  
};

**2 task.**

**1) Account class has been modified to absteact class with the help of deposit and withdraw pure virtual functions:**

virtual void withdraw(double withdraw\_number) = 0;  
  
virtual void deposit(double withdraw\_number, int choice) = 0;

**2) I\_Printable class interface (header) has been created and functionality has been provided so all accounts are printable:**

class I\_Printable {  
protected:  
 virtual void printBalance() = 0;  
};

**3 task.**

**1) Redistribution class with redistribute method has been created:**

class Redistribute{  
public:  
 void redistribute(vector <Account\*> &Accounts, double tenge){  
 int number\_of\_accounts = Accounts.size();  
 double tenge\_to\_each = tenge / number\_of\_accounts;  
 for (int i = 0; i < number\_of\_accounts; i++){  
 Accounts[i]->deposit(tenge\_to\_each, 3);  
 cout << "Distributed to each account: " << tenge\_to\_each << endl;  
 }  
 }  
  
};

**2) Code has been tested:**

