# Лаб: Дефиниране на класове

## [Дефиниране на клас](https://judge.softuni.bg/Contests/Practice/Index/228#0) Person

Създайте клас Person. Класът трябва да има:

* name: String - поле
* age: int - поле
* Name: String - свойство
* Age: int - свойство

Използвайте класа в Main по аналогичен начин:

Graphical user interface, text

Description automatically generated

### **Решение**

Разгледайте решението на задачата в презентацията.

Тествайте решението: <https://judge.softuni.bg/Contests/Practice/Index/228#0>

## [Дефиниране на клас](https://judge.softuni.bg/Contests/Practice/Index/674#0) BankAccount (банкова сметка)

Създайте клас BankAccount.

Класът трябва да има:

* id: int
* balance: double
* ID: int
* Balance: double

Трябва да може да използвате класа по аналогичен начин:

Graphical user interface, text, application, chat or text message

Description automatically generated

### **Решение**

Решението на тази задача е аналогично на решението на предната.

Тествайте решението: <https://judge.softuni.bg/Contests/Practice/Index/674#0>

## [Методи](https://judge.softuni.bg/Contests/Practice/Index/674#1)

Надградете класа BankAccount. Този клас трябва да има полета за:

* id: int
* balance: double

Класът трябва да има свойства за:

* ID: int
* Balance: double

Създайте методите:

* Deposit(Double amount): void – който да вкарва пари в сметката
* Withdraw(Double amount): void – който да изтегля пари от сметката

Заменете метода ToString()

Трябва да можете да използвате класа по аналогичен начин:

Text

Description automatically generated

### **Решение**

Създайте метод Deposit(double amount)

A picture containing diagram

Description automatically generated

Създайте метод Withdraw(double amount)

Text

Description automatically generated with medium confidence

Заменете метода ToString()

A picture containing logo

Description automatically generated

Тествайте решението: <https://judge.softuni.bg/Contests/Practice/Index/674#1>

## [Тестов Клиент](https://judge.softuni.bg/Contests/Practice/Index/674#2)

Създайте тестов клиент, който използва BankAccount.

Трябва да поддържате следните операции:

* Create {Id}
* Deposit {Id} {Amount}
* Withdraw {Id} {Amount}
* Print {Id}
* End

Ако се опитате да създадете сметка със съществуващо Id, изведете "Account already exists".

Ако се опитате да извършите операция върху несъществуваща сметка, изведете "Account does not exist".

Ако се опитате да изтеглите сума, която е по-голяма от баланса, изведете "Insufficient balance".

Print командата, трябва да изведе "Account ID{id}, balance {balance}". Закръглете баланса до втория знак след запетаята.

### **Примери**

|  |  |
| --- | --- |
| Вход | Изход |
| Create 1  Create 2  Deposit 1 20  Withdraw 1 30  Withdraw 1 10  Print 1  End | Account already exists  Insufficient balance  Account ID1, balance 10.00 |
| Create 1  Deposit 2 20  Withdraw 2 30  Print 2  End | Account does not exist  Account does not exist  Account does not exist |

### **Решение**

Използвайте Dictionary<int, BankAccount> за да пазите сметките

Направете си цикъла за приемане на команда

Text

Description automatically generated with low confidence

Създайте методи към Program.cs, за всяка от командите.

Create – проверявате дали в речника има ключ с такова id – ако няма, създавате сметката.

Graphical user interface, text, application

Description automatically generated

Имплементирайте останалите команди работейки с подобна логика.

Тествайте решението: <https://judge.softuni.bg/Contests/Practice/Index/674#2>

## [Човекът и неговите пари](https://judge.softuni.bg/Contests/Practice/Index/674#3)

Създайте клас Person.

Той трябва да има полета за:

* Name: string
* Age: int
* Accounts: List<BankAccount>

Класът трябва да има метод, който изчислява всички пари, които притежава човека от сметките си:

* GetBalance(): double

### **Решение**

Използвайте по-горния клас и му добавете възможност за пазене на списък от банкови сметки

Graphical user interface, text, application

Description automatically generated

Създайте метод GetBalance()

A screenshot of a computer

Description automatically generated with low confidence

Тествайте решението: <https://judge.softuni.bg/Contests/Practice/Index/674#3>