

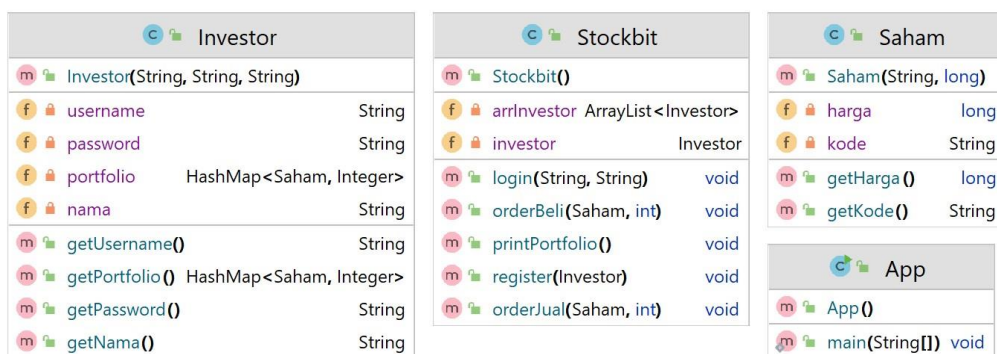
UNGUIDED 12 PRAKTIKUM RPL-BO

Java Collection

Waktu Pengerjaan: 2 jam

Soal (100 Poin)

Stockbit merupakan salah satu aplikasi yang dapat digunakan untuk melakukan transaksi jual beli saham. Pengguna dari aplikasi ini umumnya disebut dengan investor. Kumpulan dari saham yang diinvestasikan oleh investor disebut dengan portfolio. Kamu sebagai seorang programmer yang handal diminta oleh perusahaan untuk membuat aplikasi ini. Perhatikan gambar class diagram di bawah ini untuk membantu kamu dalam membangun aplikasi tersebut.



Buat class Investor, Stockbit, Saham, dan App sesuai dengan gambar class diagram di atas. Berikut adalah beberapa hal yang perlu kamu perhatikan pada saat membuat class tersebut:

Class Saham	
Nama Atribut	Keterangan
harga	Menyimpan nilai untuk harga saham
kode	Menyimpan nilai untuk kode saham
Constructor	Keterangan
Saham(String kode, long harga)	Menginisialisasi atribut harga dan kode sesuai dengan parameter
Method	Keterangan
getKode()	Getter untuk masing-masing atribut
getHarga()	

Class Investor	
Nama Atribut	Keterangan
nama	Menyimpan nilai untuk nama investor
username	Menyimpan nilai untuk username investor
password	Menyimpan nilai untuk password investor
HashMap<Saham,Integer> portfolio	Menyimpan object dari class Saham beserta dengan jumlah saham yang diinvestasikan dalam bentuk HashMap. Kamu diminta untuk <u>membuat HashMap</u> yang dapat menampung <u>object dari class Saham</u> sebagai <u>Key</u> dan <u>jumlah lot saham yang dipesan dengan tipe Integer</u> sebagai <u>Value</u> .
Constructor	Keterangan
Investor(String nama, String username, String password)	Menginisialisasi atribut nama, username, dan password sesuai dengan parameter
Method	Keterangan
getUsername()	Getter untuk masing-masing atribut
getPassword()	
getNama()	
getPortfolio()	

Class Stockbit	
Nama Atribut	Keterangan
arrInvestor	Menyimpan object dari class Investor dalam bentuk ArrayList. Kamu diminta untuk <u>membuat ArrayList</u> yang dapat <u>menampung object dari class Investor</u> .
investor	Menyimpan nilai untuk object dari class Investor
Method	Keterangan
register(Investor investor)	Menambahkan investor (client) ke dalam atribut arrInvestor bertipe <u>ArrayList</u>
login(String username, String password)	Menerima parameter username dan password untuk login ke dalam sistem. Perhatikan ketentuan berikut: <ul style="list-style-type: none"> • Jika username dan password tidak ada di dalam sistem (atribut arrInvestor bertipe <u>ArrayList</u>), maka tampilkan pesan username/password Anda salah.

	<ul style="list-style-type: none"> • Jika user berhasil login, maka atribut investor akan diisi dengan investor yang saat ini melakukan login dan tampilkan pesan keberhasilan
orderBeli(Saham saham, int lot)	<p>Menambahkan object dari class Saham beserta dengan jumlah lot ke dalam portfolio investor dan tampilkan pesan order beli sukses.</p> <p>Note: Atribut portfolio bertipe <u>HashMap</u> ada di class Investor</p>
orderJual(Saham saham, int lot)	<p>Mengurangi jumlah lot saham atau menghapus saham dari portfolio Investor. Perhatikan ketentuan berikut:</p> <ul style="list-style-type: none"> • Jika jumlah lot yang dijual sama dengan jumlah lot saham saat ini, maka hapus saham tersebut dari atribut portfolio • Jika jumlah lot yang dijual lebih dari jumlah lot saham saat ini, maka kurangi jumlah lot saham saat ini dengan jumlah lot terjual • Jika jumlah lot yang dijual lebih dari jumlah lot saham saat ini, maka tampilkan pesan order jual gagal. <p>Note: Atribut portfolio bertipe <u>HashMap</u> ada di class Investor</p>
printPortfolio()	<p>Menampilkan portfolio untuk investor (client). Perhatikan pedoman berikut ini untuk membantu kamu dalam menampilkan portfolio:</p> <ul style="list-style-type: none"> • Sub Total: harga saham * 100 * jumlah lot • Total Investasi: jumlah seluruh Sub Total <p><u>Gunakan isi dari atribut portfolio yang ada di class Investor untuk mendapatkan daftar saham milik investor.</u> Setiap investor melakukan transaksi saham, maka daftar saham dalam portfolionya akan terus bertambah.</p> <p>Note:</p> <p>- Satu lot saham = 100 lembar. Jika kamu membeli saham sebanyak 2 lot, maka kamu harus membayar sebanyak [2 lot * 100 lembar * harga saham]</p>

Apabila algoritma yang kamu bangun sudah tepat, maka masing-masing skenario akan menghasilkan **output** yang sesuai dengan gambar di bawah ini:

```

-----Skenario 1: Registrasi Investor-----
Registrasi akun atas nama Kevin Doyle sukses!
Registrasi akun atas nama ECOMMURZ sukses!
-----Skenario 2: Login Gagal-----
Username/password Anda salah!
-----Skenario 3: Login Berhasil (Username: kevin Doyle)-----
Login berhasil!
-----Skenario 4: Print Portfolio Client #1-----
Order beli LUPA sebanyak 2 lot sukses!
Order beli TOKEK sebanyak 3 lot sukses!
=====Portfolio Client=====
Client: Kevin Doyle

```

No.	Kode Saham	Jumlah Lot	Harga	Sub Total
1.	LUPA	2x	4330	866000
2.	TOKEK	3x	2530	759000

```

Total Investasi: Rp1625000
-----Skenario 5: Print Portfolio Client #2-----
Order jual gagal!
Order jual TOKEK sebanyak 1 lot sukses!
=====Portfolio Client=====
Client: Kevin Doyle

```

No.	Kode Saham	Jumlah Lot	Harga	Sub Total
1.	LUPA	2x	4330	866000
2.	TOKEK	2x	2530	506000

```

Total Investasi: Rp1372000

```

```

-----Skenario 6: Print Portfolio Client #3-----
Order jual LUPA sebanyak 2 lot sukses!
=====Portfolio Client=====
Client: Kevin Doyle
No. Kode Saham  Jumlah Lot      Harga      Sub Total
1.  TOKEK        2x          2530      506000
Total Investasi: Rp506000
-----Skenario 7: Login Berhasil (Username: murzfam)-----
Login berhasil!
-----Skenario 8: Print Portfolio Client #4-----
Order beli ILAP sebanyak 2 lot sukses!
Order beli TOKTOK sebanyak 1 lot sukses!
=====Portfolio Client=====
Client: ECOMMURZ
No. Kode Saham  Jumlah Lot      Harga      Sub Total
1.  TOKTOK        1x          7375      737500
2.  ILAP          2x          3150      630000
Total Investasi: Rp1367500

```

Poin Penilaian:

- Implementasi **Java Collection** pada class **Investor** dan class **Stockbit** tepat dan sesuai. (Poin 10)
- berhasil melakukan Maven Compile (Poin 20)
- semua skenario AppTest berjalan sukses: (Total Poin 70)
 - testRegister() (Poin 5)
 - testBerhasilLogin() (Poin 10)
 - testGagalLogin() (Poin 10)
 - testPrintNota() (Poin 25)
 - testBerhasilJual() (Poin 10)
 - testGagalJual() (poin 10)