

Základy procedurálneho programovania 2



FIIT STU, Mlynská dolina
Aula Minor, pondelok 9:00

letný semester
2016/2017

Ako sa pracuje na tomto predmete?

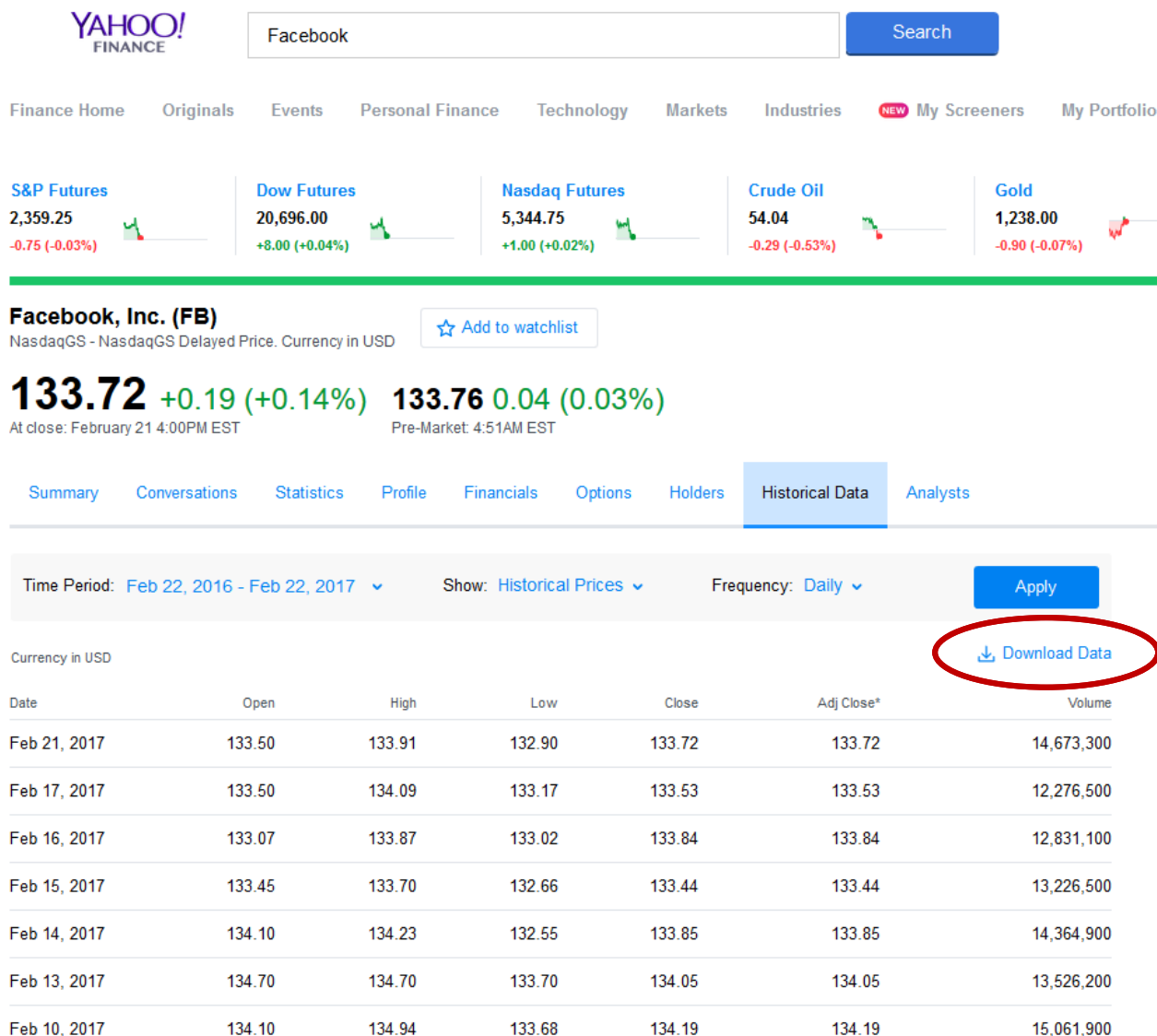
- **Prednášky** (pondelok 9:00):
Jozef Tvarožek
zapájať sa, pýtať sa, počúvať, ... písomné testy (20%)
- **Cvičenia** (streda 12:00, 14:00, 16:00, štvrtok 11.00):
Marián Potočný
riešenie malých úloh, projekty (40%)
www.turing.sk/fiit/zpp2
- Povinné minimum zo semestra 30 bodov
- **Skúška:**
písomný test (40%) (povinná účasť)



Algoritmické obchodovanie...



Historický vývoj cien akcií




Stiahneme údaje
vo formáte .CSV

Formát údajov

- Načítame najvyššiu (High) a najnižšiu (Low) cenu počas každého obchodného dňa

Date, Open, **High**, **Low**, Close, Volume, Adj Close



```
2017-02-17, 133.50, 134.089996, 133.169998, 133.529999, 12249900, 133.529999
2017-02-16, 133.070007, 133.869995, 133.020004, 133.839996, 12809800, 133.839996
2017-02-15, 133.449997, 133.699997, 132.660004, 133.440002, 13144500, 133.440002
2017-02-14, 134.100006, 134.229996, 132.550003, 133.850006, 14340100, 133.850006
2017-02-13, 134.699997, 134.699997, 133.699997, 134.050003, 13503300, 134.050003
2017-02-10, 134.100006, 134.940002, 133.679993, 134.190002, 14557400, 134.190002
2017-02-09, 134.490005, 134.50, 133.309998, 134.139999, 16441600, 134.139999
2017-02-08, 132.600006, 134.440002, 132.440002, 134.199997, 22329500, 134.199997
2017-02-07, 132.240005, 133.00, 131.660004, 131.839996, 14563800, 131.839996
2017-02-06, 130.979996, 132.059998, 130.300003, 132.059998, 16911200, 132.059998
```

- Usporiadané od najnovšieho dátumu (poradie načítaných údajov musíme obrátiť)

Načítanie historických cien (implementácia)

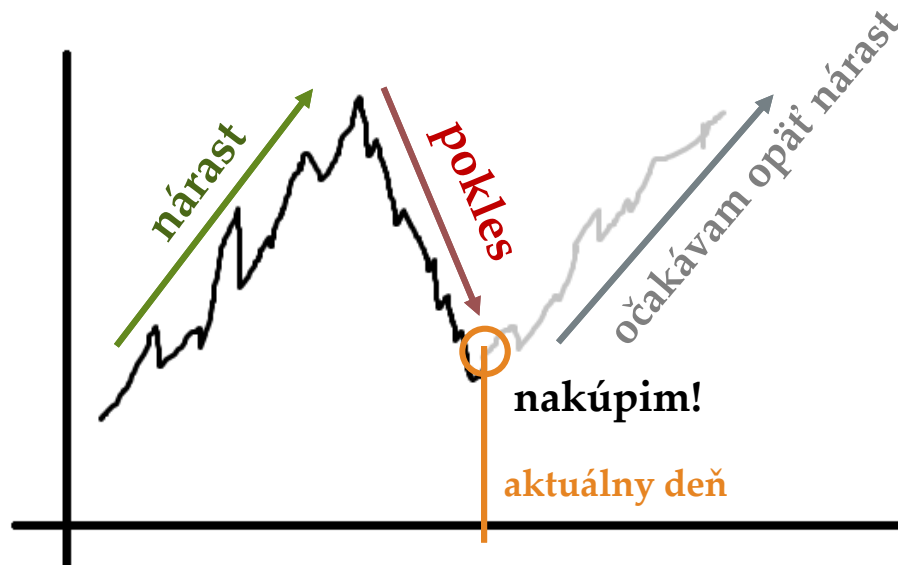
```
void read(FILE *f)
{
    char buf[1000], datum[100];
    double open;
    int i;
    n = -1;
    while (fscanf(f, "%s", buf) > 0)
    {
        if (n++ < 0)
            continue;
        for (i = 0; buf[i] > 0; i++)
            if (buf[i] == ',')
                buf[i] = ' '; // ciarky nahradim medzerami
        sscanf(buf, "%s %lf %lf %lf", datum, &open, &hi[n], &lo[n]);
    }
    n++;
    for (i = 0; i < n-i-1; i++) // obratim datумы
    {
        swap(&hi[i], &hi[n-i-1]);
        swap(&lo[i], &lo[n-i-1]);
    }
    n -= 2; // korekcia chyby vo vstupnom subore
}
```

Premyslíme investičnú stratégiu...

- Máme rozpočet 1000 USD
- Fee – poplatok za nákup/predaj (napr. 10 USD)
- Algoritmus: Prechádzame obchodné dni od najstaršieho po najnovší, a uvažujeme:
 - Či nakúpime (ak nemáme ešte nakúpené akcie)
 - Či predáme (ak máme nakúpené akcie)

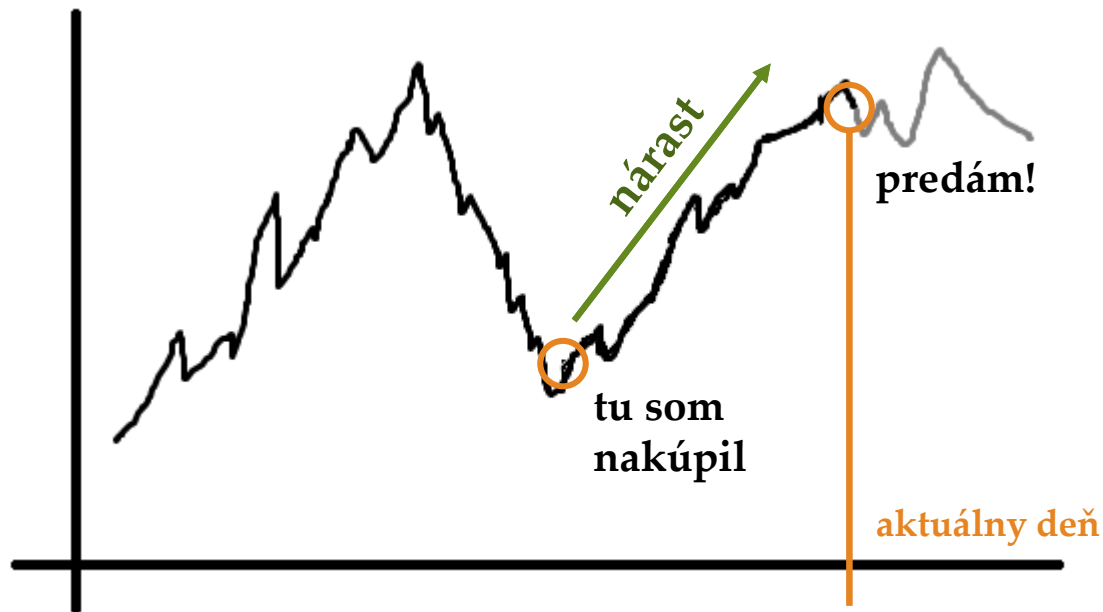
Kedy nakúpime?

- Hlavná myšlienka: očakávame, že nárast z minulosti sa zopakuje, a potom predáme s veľkým ziskom
- Kedy je teda vhodné nakúpiť?
Keď za posledných (napr. 45 dní) sme zaznamenali veľký nárast a teraz je veľký pokles ceny akcií ...



Kedy predáme?

- Cena akcií musí vzrásť aspoň o nejaké požadované percento (napr. 30%)
- Predáme keď cena akcií začala klesať (nebudeme predsa predávať ak cena každým dňom len rastie...)



Kedy nakúpime? (implementácia)

- Nakúpime ak bol v minulosti **nárast**, teraz je **pokles** a je to **prvé stúpanie** (nebudeme kupovať, keď cena každý deň len klesá...):

```
for (i = 1; i < n; i++)
{
    if (stocks == 0.0) // CHCEM NAKUPIT
    {
        // ak to v poslednej dobe (45dni) malo narast aspon support%
        int min, k = i-30;
        if (k < 0) k = 0;
        min = k; // najnizsia cena akcie za poslednych 45 dni
        for (k++; k < i; k++)
            if (lo[min] > lo[k])
                min = k;
        int max = min; // najvyssia cena akcie od datumu min do i.
        for (k++; k < i; k++)
            if (hi[max] < hi[k])
                max = k;
        double narast = (hi[max]-lo[min])/lo[min]; // >1.0
        double pokles = (hi[max]-lo[i])/hi[max]; // <1.0
        if (narast >= support && pokles >= fall && lo[i] > lo[i-1])
        {
            // NAKUPIM
            stock_price = hi[i];
            stocks += floor((budget - 2*fee)/hi[i]);
            budget -= stocks * hi[i];
        }
    }
}
```

Kedy predáme? (implementácia)

- Predáme pri **prvom poklese** ak sme dosiahli **aspoň požadované percento výnosu**:

```
for (i = 1; i < n; i++)
{
    if (stocks > 0.0) // CHCEM PREDAT
    {
        double vynos = (lo[i]-stock_price)/stock_price;
        if (vynos > rise && lo[i] < lo[i-1]) // prvý pokles: lo[i-1] > lo[i]
        {
            budget -= fee;
            budget += stocks * lo[i];
            stocks = 0;
        }
    }
}
```

Už len investujeme...

- Určíme vhodné parametre:
 - Aké % nárastu v minulosti požadujeme
 - Aké % poklesu v aktuálny deň požadujeme
 - Aké % nárastu oproti nákupnej cene požadujeme
 - Koľko dní dozadu budeme trh analyzovať
- **Trénovacia množina:** Parametre určíme na historických údajov okrem posledného roku
- **Testovacia množina:** Výnosnosť nášho algoritmu investovania overíme na poslednom roku vývoja cien
- Vyskúšame našu investičnú stratégiu na rozličných tituloch: facebook, Google, Intel, AMD, ...

Nabudúce...

dátum	prednáška	8:00	9:00	cvičenie	obsah
13.2.	1		Opakovanie	1	Projekt 1 Snehulienka
20.2.	2		Algoritmické obchodovanie	2	
27.2.	3	Test 1	Riešenie testu 1, Snehulienka	3	
6.3.	4		Snehulienka (pokr.)	4	Projekt 1: odovzdanie, konzultovanie
13.3.	5	Test 2	Riešenie testu 2, Smerníky	5	
20.3.	6		Čítanie kódu, Hľadanie chýb v kóde	6	