Představení úkolu: Burza

V tomto domácím úkolu si naprogramujeme zjednodušenou reprezentaci akciové burzy. Skutečné burzy typicky umožňují zadávat různé druhy pokynů, my se však omezíme jen na jednoduché nákupní a prodejní pokyny. Každý pokyn vždy obsahuje jméno (nebo jiný identifikátor) obchodníka, který jej podal, množství akcií, které chce nakoupit nebo prodat, a jednotkovou cenu za akcii. Podle typu pokynu je jednotková cena buď maximální cena, za kterou chce obchodník nakupovat, nebo minimální cena, za kterou chce prodávat.

Nenajde-li se k podanému pokynu odpovídající pokyn opačný, stane se z něj pokyn čekající. Čekající pokyny se řadí podle jednotkové ceny – přednost mají nákupní pokyny s vyšší cenou a prodejní pokyny s nižší cenou (tedy ty, které jsou blíže středu). Mezi pokyny se stejnou cenou mají přednost ty, které přišly na burzu dříve. Přijde-li na burzu nový pokyn, pokusí se vypořádat (i částečně) s existujícími opačnými pokyny. Pokud se celý nevypořádá, jeho zbytek se zařadí mezi pokyny čekající.

Vypořádání mezi pokyny je možné, pokud je jednotková cena nákupního pokynu větší nebo rovna jednotkové ceně prodejního pokynu. Při vypořádání se mezi obchodníky převede tolik akcií, kolik je minimum z počtů v obou pokynech. Cena vypořádání se v realitě počítá různě. My budeme v tomto zadání používat vypořádání zvýhodňující kupce, tj. cena obchodu bude vždy dána cenou prodejního pokynu.

Ukážeme si to na příkladu: Mějme akciovou společnost ACME, která na začátku obchodování nemá žádné čekající příkazy. Postupně budou přicházet tyto příkazy:

1. Strýček Skrblík chce prodat 50 akcií za cenu $120.
2. Rampa McKvák chce nakoupit 100 akcií za cenu $90.
3. Hamoun Držgrešle chce prodat 70 akcií za cenu $110.
4. Kačer Donald chce prodat 20 akcií za cenu $120.

Žádné z těchto pokynů se zatím nevypořádají a situaci čekajících pokynů si můžeme znázornit takto:

```
     cena  počet  obchodník
-------------------------------------------------------------
      120     20 (Kačer Donald)
      120     50 (Strýček Skrblík)
      110     70 (Hamoun Držgrešle)
-------------------------------------------------------------
       90    100 (Rampa McKvák)
-------------------------------------------------------------
```

(Nákupní pokyny píšeme dolů, prodejní pokyny nahoru, přednost mají ty pokyny, které jsou blíže středové čáře.)

Dále přijde pokyn Paní Čvachtové, která chce koupit 90 akcií za $110. Tento pokyn se částečně vypořádá s pokynem Hamouna Držgrešle a paní Čvachtová obdrží jeho 70 akcií za cenu $110. Protože její pokyn nebyl zcela vypořádán a další prodejní pokyny, se kterými by se mohl vypořádat, už nejsou, zařadí se mezi čekající nákupní pokyny. Situace tedy vypadá takto:

```
     cena  počet  obchodník
-------------------------------------------------------------
      120     20 (Kačer Donald)
      120     50 (Strýček Skrblík)
-------------------------------------------------------------
      110     20 (Paní Čvachtová)
       90    100 (Rampa McKvák)
-------------------------------------------------------------
```

Nakonec přijde nákupní pokyn Magiky von Čáry, která chce koupit 60 akcií za cenu $130. Tento pokyn se nejprve vypořádá s pokynem Strýčka Skrblíka (má stejnou cenu jako pokyn Kačera Donalda, ale přišel na burzu dříve), a to za $120. Protože tím není pokyn zcela vypořádán, zbylých 10 akcií se za stejnou cenu zobchoduje s Kačerem Donaldem.

Výsledná situace tedy vypadá takto:

```
     cena  počet  obchodník
-------------------------------------------------------------
      120     10 (Kačer Donald)
-------------------------------------------------------------
      110     20 (Paní Čvachtová)
       90    100 (Rampa McKvák)
-------------------------------------------------------------
```

Pro rekapitulaci, postupně proběhly následující obchody:

```
Hamoun Držgrešle -> Paní Čvachtová: 70 akcií za $110
Strýček Skrblík -> Magika von Čáry: 50 akcií za $120
Kačer Donald -> Magika von Čáry: 10 akcií za $120
```

Část první (1 bod): Datové struktury

Třída `Transaction` bude reprezentovat jednotlivou burzovní transakci. Má mít tyto atributy:

- `buyer_id` – řetězec označující jméno kupujícího;
- `seller_id` – řetězec označující jméno prodávajícího;
- `amount` – kladné celé číslo označující počet zobchodovaných akcií;

- `price` – kladné celé číslo označující jednotkovou cenu, za kterou obchod proběhl.

  Vytvořte ve třídě `Transaction` metodu `__init__` tak, aby se její objekty daly vytvářet pomocí `Transaction(buyer_id, seller_id, amount, price)`.

Třída `Order` bude reprezentovat obchodní pokyn (nákup nebo prodej). Má mít tyto atributy:

- `trader_id` – řetězec označující obchodníka, který vydal pokyn;
- `amount` – kladné celé číslo označující požadovaný počet akcií;
- `price` – kladné celé číslo označující požadovanou cenu.

Vytvořte ve třídě `Order` metodu `__init__` tak, aby se její objekty daly vytvářet pomocí `Order(trader_id, amount, price)`.

Třída `Stock` bude obsahovat všechny informace týkající se jedné akciové společnosti. Má mít tyto atributy:

- `history` – seznam všech transakcí (objektů typu `Transaction`), které se s akciemi této společnosti prováděly;
- `buyers` – seznam všech čekajících (dosud nevypořádaných) nákupních pokynů (objektů typu `Order`);
- `sellers` – seznam všech čekajících (dosud nevypořádaných) prodejních pokynů (objektů typu `Order`).

Seznam `buyers` budeme chtít udržovat vždy seřazený tak, aby pokyny s vyšší cenou byly blíže ke konci seznamu; pokud je více pokynů se stejnou cenou, pak bude blíže ke konci seznamu pokyn, který přišel dříve (pomocí procedury `place_buy_order` níže).

Seznam `sellers` budeme chtít udržovat vždy seřazený tak, aby pokyny s nižší cenou byly blíže ke konci seznamu; pokud je více pokynů se stejnou cenou, pak bude blíže ke konci seznamu pokyn, který přišel dříve (pomocí procedury `place_sell_order` níže).

Poznámka: Toto uspořádání zaručí, že pokyny na konci seznamu budou ty, které se mají vypořádat jako první, čehož můžete s výhodou využít při implementaci procedur `place_*_order`.

Vytvořte ve třídě `Stock` metodu `__init__` tak, aby se její objekty daly vytvářet pomocí `Stock()`, přičemž vytvořený objekt bude ve výše uvedených atributech obsahovat prázdné seznamy.

Kromě požadovaných atributů a metody `__init__` si můžete do tříd přidat libovolné pomocné atributy nebo metody.

Pro reprezentaci celé burzy budeme používat typový alias `StockExchange`, který zastupuje slovník, jehož klíči jsou zkrácená burzovní jména akciových společností (tzv. *ticker symbol*) a hodnotami objekty typu `Stock`. Tento alias je již připravený v kostře; neměňte jej.

Dále implementujte proceduru

```
add_new_stock(stock_exchange: StockExchange, ticker_symbol: str) -> bool
```

která se do zadané burzy pokusí přidat novou akciovou společnost (tedy nový objekt typu `Stock`) se zadaným ticker symbolem. Pokud už akciová společnost s tímto symbolem na burze existuje, nic nezmění a vrátí `False`. V opačném případě vrátí `True`.

V kostře řešení je připravena procedura `print_stock`, která pro zadaný ticker symbol textově vykreslí tabulku čekajících pokynů a seznam transakcí. Tuto proceduru můžete využít pro ladění.

Část druhá (2 body): Zadávání pokynů

Implementujte procedury:

```
place_buy_order(stock_exchange: StockExchange, ticker_symbol: str,
                trader_id: str, amount: int, price: int) -> None

place_sell_order(stock_exchange: StockExchange, ticker_symbol: str,
                 trader_id: str, amount: int, price: int) -> None
```

které na burzu vloží nový nákupní, resp. prodejní pokyn a pokusí se jej co nejvíce vypořádat (viz úvod). Vypořádaný pokyn (nebo jeho části) vloží na konec seznamu `history`, nevypořádaný pokyn (nebo jeho část) zařadí do odpovídajícího seznamu pokynů.

Vstupní podmínkou obou procedur je, že

- `ticker_symbol` je validní ticker symbol, který je klíčem ve slovníku burzy;
- `amount` a `price` jsou kladná celá čísla.

Procedury implementuje tak, aby udržovaly
seznamy `buyers` a `sellers` seřazené, jak je popsáno v první části.

Část třetí (1,5 bodu): Dotazy

Implementuje tyto čisté funkce:

```
stock_owned(stock_exchange: StockExchange, trader_id: str) -> Dict[str,
int]

all_traders(stock_exchange: StockExchange) -> Set[str]

transactions_by_amount(stock_exchange: StockExchange,
                       ticker_symbol: str) -> List[Transaction]
```

Funkce `stock owned` vrátí slovník všech akcií, které podle informací burzy vlastní zadaný obchodník. Klíči slovníku jsou ticker symboly akcií, hodnotami pak jejich počet, který musí odpovídat proběhlým obchodům. Akcie, jejichž počet by měl být nulový, nebudou ve slovníku zastoupeny.

Poznámka: Všimněte si, že počet akcií může být i záporný, pokud například zadaný obchodník pouze prodával, ale nic nenakoupil. Můžete si představovat, že to reprezentuje situace, kdy obchodník akcie získal někde mimo burzu nebo si je půjčil (pro účely tzv. krátkého prodeje). Rovněž se může stát, že nějaký obchodník prodá akcie sám sobě. Takový obchod samozřejmě počet jeho akcií nijak nemění.

Funkce `all_traders` vrátí množinu všech (jmen) obchodníků, kteří na burze kdy podali nějaký pokyn.

Funkce `transactions_by_amount` vrátí seznam všech transakcí, které se prováděly s akciemi dané společnosti, seřazený podle množství akcií sestupně. Pořadí transakcí se stejným množstvím akcií by mělo zůstat zachováno.

Část čtvrtá (1,5 bodu): Dávkové zpracování

Napište proceduru

```
process_batch_commands(stock_exchange: StockExchange,
                       commands: List[str]) -> Optional[int]
```

která dostane na vstupu seznam příkazů a provede je se zadanou burzou v pořadí daném tímto seznamem. Seznam příkazů přitom nijak nemodifikuje.

Příkazy jsou řetězce, přičemž smíte předpokládat, že se v těchto řetězcích objevují pouze tisknutelné znaky US-ASCII (s ordinálními čísly 32 až 126).

Povolené příkazy jsou tyto:

- `ADD ticker_symbol`, kde `ticker_symbol` je libovolná neprázdná posloupnost znaků **bez mezer**. Vykonání tohoto příkazu přidá do slovníku burzy novou akciovou společnost. Pokud už akciová společnost se zadaným ticker symbolem na burze existuje, je tento příkaz neplatný.
- `name: BUY amount ticker_symbol AT price`, kde `name` je libovolná neprázdná posloupnost znaků **neobsahující znak dvojtečky**, `amount` a `price` jsou desítkově zapsaná kladná celá čísla a `ticker_symbol` je libovolná neprázdná posloupnost znaků **bez mezer**. Pokud zadaný ticker symbol na burze neexistuje, je tento příkaz neplatný. V opačném případě se tento příkaz vykoná a přidá na burzu nákupní pokyn (pomocí procedury `place_buy_order`) se zadaným jménem obchodníka, množstvím, cenou a ticker symbolem akciové společnosti.
- `name: SELL amount ticker_symbol AT price` je podobný jako předchozí, jen místo nákupního pokynu se jedná o pokyn prodejní.

Jakýkoli jiný řetězec je jako příkaz neplatný.

Pokud seznam řetězců obsahuje nějaký neplatný příkaz, provedou se pouze příkazy před prvním neplatným příkazem a procedura vrátí index prvního neplatného příkazu. V opačném případě se provedou všechny příkazy a procedura vrátí `None`.