Problem 2 – Nightlife Entertainment

You are given a system for selling cinema tickets. However, it is quite incomplete. Your task is to extend the system's functionality.

The system consists of **venues**, **performances** and **tickets**.

- A **venue** is a place where **performances** happen. **Venues** have **name**, **location**, **number of seats**, and **allowed performance types**. The allowed types specify what kinds of performances can happen at a given venue.
- A **performance** happens at a **venue**. It has a **name**, **performance type**, **base price**, **start time** and a set of **tickets**. The base price specifies the price of a single ticket which has no discounts or any other price corrections applied.

  There are certain **restrictions** regarding a performance: it must not have more tickets than the number of seats in a venue, and the venue type must be **compatible** (i.e. the performance must be allowed in that venue).

- A **ticket** contains information about the **performance** it belongs to, the **price** (which is related to the base price of the performance, but there may be some corrections applied), the **seat number**, and the **type**. There are currently three possible types of tickets: **regular**, **student**, and **VIP**. A ticket also has a **status** – sold or unsold.

When a **performance is created**, **tickets may be added** (supplied) to the performance and **sold**.

The system currently supports the following classes:

- **`Cinema`** – a venue where movies and concerts may be shown.
- **`Movie`** – a performance which may be shown in venues that support **movies** or **theatres**.
- **`RegularTicket`** – a ticket whose price is the same as the performance's base price.

You are given an engine which can execute commands. The commands this engine currently supports are:

- **`insert venue cinema <name> <location> <number_of_seats>`** – Adds a new cinema with the given name, location, and number of available seats to the database.
- **`insert performance movie <name> <base_price> <venue_name> <date> <time>`** – Adds a new performance with the given name, base price, venue, and starting time to the database. In case of invalid venue, the engine returns an error message.
- **`supply_tickets regular <venue_name> <performance> <count>`** – Adds the specified number of tickets to the performance. In case of invalid venue or performance, or if all seats in the venue have been sold, the engine returns an error message.
- **`sell_ticket <performance> <type>`** – Sells the first available ticket for the given performance. It also changes the ticket status to be **sold**. In case there is no such performance, or there are no tickets left of this particular type, the engine returns an error message. As a result, the system prints the ticket so that it can be sold to the user.
- **`report <performance>`** – Due to a lack of time and abilities, the developers have skipped this command and it does not currently work.
- **`find <word> <start_date> <start_time>`** – Our developers have skipped this too so guess whose task it will be to implement it :D.

The engine skips all invalid commands. It also handles invalid data passed as command parameters.

Your task is to study how the system currently works and extend it using the **best practices** of **Object-Oriented Design (OOD)** and **Object-Oriented Programming (OOP)**. You have to properly apply the principles of OOP in order to **avoid code repetition** and **achieve code reuse**.

You are NOT allowed to change any of the provided classes or interfaces. You may edit **only one line** in the **Main()** method.

Venues

- Implement an **opera hall**:
  - An opera hall is a type of venue where **operas** and **theatre plays** can be displayed.
  - Command for inserting an opera hall:

```
insert venue opera <name> <location> <number_of_seats>
```

- Implement a **sports hall**:
  - A sports hall is a type of venue where people can watch **sports** and **concerts**.
  - Command for inserting a sports hall:

```
insert venue sports_hall <name> <location> <number_of_seats>
```

- Implement a **concert hall**:
  - A concert hall is a type of venue where people can watch **operas**, **theatres** and **concerts**.
  - Command for inserting a concert hall:

```
insert venue concert_hall <name> <location> <number_of_seats>
```

Performances

- Implement a **theatre**:
  - A theatre is a type of performance which can happen in any venue that supports showing **theatres**.
  - Command for inserting a theatre:

```
insert performance theatre <name> <base_price> <venue_name> <date> <time>
```

- Implement a **concert**:
  - A concert is a type of performance which can be held anywhere, as long as the place supports showing **concerts** or **operas**.
  - Command for inserting a concert:

```
insert performance concert <name> <base_price> <venue_name> <date> <time>
```

Tickets

- Implement a **student ticket**:
  - The student ticket should try to get its **performance**. In case there is a valid performance for this ticket, its price should be **80%** of the regular price of the performance.
  - A student ticket is identified by the type **student** in the engine
- Implement a **VIP ticket**:
  - The VIP ticket should try to get its **performance**. In case there is a valid performance for this ticket, its price should be **1.5 times** the regular price of the performance.
  - A VIP ticket is identified by the type **vip** in the engine

Note that you should implement **supplying** and **selling** tickets of the new types.

Reporting

Implement the **report** command. The command returns a report of how many tickets have been sold for the performance, and their total price (**rounded to two digits** after the decimal separator). A report has the following form:

```
<name>: <tickets_sold> ticket(s), total: $<total_price>
Venue: <venue_name> (<venue_location>)
Start time: <start_time>
```

Refer to the sample input and output to get a better understanding about the report structure and data.

Searching

Implement the **find** command. The command performs a **case-insensitive** search for **venues** and **performances** whose name matches the search phrase **at least partially** (for example, "**wolfof**" will match "The**WolfOf**WallStreet").

First display all found **performances** that start **at or after the specified time**, sorted by their **start time**, then by **price** in descending order, and finally by **name**.

After that display all found **venues** which match the search phrase, sorted by **name**.

If there are any events for a given venue that start **at or after the specified time**, list them, sorted by **start time**, then by **price** in descending order, and finally by **name**.

A search result has the following form:

```
Search for "<phrase>"
Performances:
-<performance_name>
-<performance_name>
...
Venues:
-<venue_name>
--<performance_name>
--<performance_name>
-<venue_name>
-<venue_name>
--<performance_name>
--<performance_name>
...
```

If there are no performances or no venues, print "**no results**" under Performances:/Venues:.

Input

```
insert venue cinema Arena Sofia-Mladost 200
insert performance movie Dracula 12.00 Arena 12.10.2014 20:00:00
supply_tickets regular Arena Dracula 50
sell_ticket Dracula regular
sell_ticket Dracula regular
sell_ticket Dracula regular
sell_ticket Dracula regular
insert venue opera LaScala Milan 800
insert venue sports_hall ArenaArmenets Sofia 210
insert venue concert_hall Metropolitan NewYork 400
insert performance theatre MuchAdoAboutNothing 10.00 Metropolitan 5.11.2014 18:30:00
insert performance concert TarjaTurunen 55.00 Metropolitan 10.11.2014 20:00:00
supply_tickets regular Metropolitan TarjaTurunen 100
supply_tickets student Metropolitan TarjaTurunen 50
supply_tickets vip Metropolitan TarjaTurunen 50
sell_ticket TarjaTurunen regular
sell_ticket TarjaTurunen student
sell_ticket TarjaTurunen vip
report Dracula
report MuchAdoAboutNothing
report TarjaTurunen
insert venue concert_hall ConcertHallNDK Sofia 150
insert performance concert RadoShisharkataConcert 25.55 Metropolitan 8.12.2014
18:30:00
insert performance concert SlaviTrifonovConcert 12.00 ConcertHallNDK 7.12.2014
19:00:00
insert performance concert TarjaTurunen 55.00 ConcertHallNDK 10.12.2014 20:00:00
insert venue concert_hall OtherConcertHall Sydney 300
find conCERt 1.08.2014 14:30:00
find conCErt 1.08.2015 22:30:00
end
```

Output

```
===== Dracula =====
At Arena (Sofia-Mladost)
On 12/10/2014 20:00:00
Seat: 1
Price: $12.00
===============
===== Dracula =====
At Arena (Sofia-Mladost)
On 12/10/2014 20:00:00
Seat: 2
Price: $12.00
===============
===== Dracula =====
At Arena (Sofia-Mladost)
On 12/10/2014 20:00:00
Seat: 3
Price: $12.00
===============
===== Dracula =====
```

```
At Arena (Sofia-Mladost)
On 12/10/2014 20:00:00
Seat: 4
Price: $12.00
===============
===== TarjaTurunen =====
At Metropolitan (NewYork)
On 10/11/2014 20:00:00
Seat: 1
Price: $55.00
===============
===== TarjaTurunen =====
At Metropolitan (NewYork)
On 10/11/2014 20:00:00
Seat: 101
Price: $44.00
===============
===== TarjaTurunen =====
At Metropolitan (NewYork)
On 10/11/2014 20:00:00
Seat: 151
Price: $82.50
===============
Dracula: 4 ticket(s), total: $48.00
Venue: Arena (Sofia-Mladost)
Start time: 12/10/2014 20:00:00
MuchAdoAboutNothing: 0 ticket(s), total: $0.00
Venue: Metropolitan (NewYork)
Start time: 05/11/2014 18:30:00
TarjaTurunen: 3 ticket(s), total: $181.50
Venue: Metropolitan (NewYork)
Start time: 10/11/2014 20:00:00
Search for "conCERt"
Performances:
-SlaviTrifonovConcert
-RadoShisharkataConcert
Venues:
-ConcertHallNDK
--SlaviTrifonovConcert
--TarjaTurunen
-OtherConcertHall
Search for "conCErt"
Performances:
no results
Venues:
-ConcertHallNDK
-OtherConcertHall
```