

Der angesagteste Club Frankfurts benötigt für die Verwaltung seiner Gästeliste eine neue Software, die Sie programmieren sollen. Ihre Aufgabe ist es, die objektorientierten Datenstrukturen inklusive der passenden Implementation in C++ nach folgenden Vorgaben umzusetzen:

- Denken Sie an das Programmierparadigma der Datenkapselung.
- ~~Trennen Sie jeweils die Deklaration von der Implementation.~~
- Vermeiden Sie doppelten Code zur einfacheren Wartung Ihres Quelltextes.
- Fangen Sie in Ihrem Programm Fehler (wie bspw. NULL-Pointer oder Speicherlecks) ab.
- Programmieren Sie eine Klasse Person mit folgenden Eigenschaften:
  - Eine Person hat einen Namen vom Datentyp `char*`. Sollten Sie hiermit Schwierigkeiten haben, so können Sie alternativ (mit entsprechendem Punktabzug) auch `std::string` verwenden.
  - Der Name einer Person kann nur bei Instanziierung gesetzt werden.
  - Personen können nicht kopiert werden.
- Programmieren Sie eine Klasse Gast, welche alle Eigenschaften einer Person übernimmt mit folgenden zusätzlichen Eigenschaften:
  - Jeder Gast hat einen individuellen Eintrittspreis je nach Auftreten/Outfit zu zahlen. Der Türsteher entscheidet bei Instanziierung über die Höhe des Eintrittspreises.
  - Wird eine Instanz erzeugt, so wird der Text „XYZ hat für 123 Euro den Laden betreten.“ auf dem Bildschirm ausgegeben. XYZ wird dabei durch den entsprechenden Namen und 123 durch den entsprechenden Eintrittspreis der Instanz ersetzt.
  - Wird eine Instanz zerstört, so wird der Text „XYZ hat den Laden wieder verlassen.“ auf dem Bildschirm ausgegeben. XYZ wird dabei durch den entsprechenden Namen der Instanz ersetzt.
- Programmieren Sie eine Klasse Party mit folgenden Eigenschaften:
  - Eine Party besteht aus beliebig vielen Personen. Nutzen Sie hierzu die Datenstruktur `std::vector`.
  - Fügen Sie eine Funktion `hereinlassen()` hinzu, um der Party jeweils eine weitere Person hinzufügen zu können.
- Es kommt wie es kommen musste. Kurz vor Projektabgabe gibt es noch eine zusätzliche Anforderung. Sie sollen ebenfalls die Klasse VIP implementieren, welche im Gegensatz zur Klasse Gast keinen Eintrittspreis bezahlen muss, dafür aber die Kennzeichnung „VIP“ besitzt. Da Sie bereits vorab mit solchen Dingen rechneten, haben Sie Vorkehrungen getroffen, dass Sie diese Klasse ohne weitere Änderungen in Ihrem bisherigen Programm hinzufügen können (Polymorphismus). Sollten Sie hiermit Schwierigkeiten haben, so können Sie (mit entsprechendem Punktabzug) diese Klasse komplett weglassen.
- Überladen Sie den Streamingoperator so, dass bei Personen der Name, bei Gästen Name und Eintrittspreis und bei VIP – falls von Ihnen implementiert – Name und die Information, dass es sich um einen VIP handelt, ausgegeben werden. Bei Partys soll die Anzahl der Personen und die jeweiligen Informationen aller zugehörigen Personen ausgegeben werden.  
Falls Sie hiermit Schwierigkeiten haben, so können Sie (mit entsprechendem Punktabzug) auch nur eine Methode `print()` implementieren, welche anstelle des Streamingoperators aufgerufen werden kann.

Bitte wenden!

- Programmieren Sie nun ein Hauptprogramm zum Testen Ihrer Klassen. Ihr Hauptprogramm soll dabei folgenden Ablauf haben:
  - Erstellen Sie einen Gast „Paula Partygirl“ mit dem symbolischen Eintrittspreis von 1€,
  - Erstellen Sie einen Gast „Simon Supersimp“ mit dem Eintrittspreis von 100€.
  - Falls Sie auch die Klasse VIP implementiert haben, erstellen Sie einen VIP mit dem Namen „Ina Influencer“.
  - Erstellen Sie eine Party und fügen Sie die zwei bzw. drei oben erstellten Instanzen hinzu.
  - Geben Sie die Party per Streaming-Operator (oder alternativ, falls Sie den Streaming-Operator nicht programmiert haben per print()-Methode) auf dem Bildschirm aus.

Eine dazu passende Bildschirmausgabe könnte wie folgt aussehen:

**Paula Partygirl hat für 1 Euro den Laden betreten.**

**Simon Supersimp hat für 100 Euro den Laden betreten.**

**Auf der Party sind folgende 3 Personen:**

**Paula Partygirl (1 Euro)**

**Simon Supersimp (100 Euro)**

**Ina Influencer (VIP)**

**Paula Partygirl hat den Laden wieder verlassen.**

**Simon Supersimp hat den Laden wieder verlassen.**