**🚴** Fahrradladen-Verwaltungssystem mit Fahrradtypen – OOP

**1. Anforderungen an das System**

Ein Fahrradladen soll Fahrräder **verkaufen und vermieten**. Es gibt verschiedene Fahrradtypen mit speziellen Eigenschaften:

* **Citybike**: Hat einen **Gepäckträger** und ist günstiger in der Miete.
* **Mountainbike**: Hat **zusätzliche Federung** und ist teurer in der Miete.
* **Rennrad**: Ist besonders **leicht**, aber hat keine Federung oder Gepäckträger.

Das System soll folgende Hauptfunktionen enthalten:

✅ **Fahrräder verwalten** (neue hinzufügen, verkaufen, vermieten)  
✅ **Kunden verwalten** (Daten speichern, Miet- und Kaufhistorie führen)  
✅ **Rückgabe mit Strafen für verspätete Rückgabe**  
✅ **Lagerbestand überwachen**  
✅ **Unterschiedliche Preise je nach Fahrradtyp**

✅ **Man soll auch mehrere Fahrräder nacheinander ausleihen oder kaufen können**

**2. Technische Umsetzung**

Das Projekt wird modular aufgebaut. Erstelle folgende **Python-Dateien**:

1. **fahrrad.py** – enthält die **Basisklasse Fahrrad**.
2. **citybike.py** – enthält die Unterklasse Citybike.
3. **mountainbike.py** – enthält die Unterklasse Mountainbike.
4. **rennrad.py** – enthält die Unterklasse Rennrad.
5. **kunde.py** – enthält die Klasse Kunde, die Kundendaten speichert.
6. **fahrradladen.py** – enthält die Klasse Fahrradladen, die den Laden verwaltet.
7. **main.py** – Hauptprogramm zum Testen der Funktionen.

**3. Detaillierte Klassendefinitionen**

**fahrrad.py (Basisklasse Fahrrad)**

* Attribute:
  + modell (z. B. Citybike, Mountainbike, Rennrad)
  + marke (z. B. Cube, Giant, Scott)
  + preis (Kaufpreis)
  + mietgebuehr (Mietkosten pro Tag)
  + verfuegbar (Ob das Fahrrad verfügbar ist oder nicht)
* Methoden:
  + fahrrad\_info() – Gibt die Details des Fahrrads aus.
  + berechne\_mietkosten(tage) – Berechnet die Mietkosten für eine bestimmte Anzahl von Tagen.

**citybike.py (Unterklasse Citybike)**

* Erbt von Fahrrad und hat zusätzlich:
  + gepaecktraeger (True/False)
  + mietgebuehr ist günstiger als bei anderen Fahrrädern

**mountainbike.py (Unterklasse Mountainbike)**

* Erbt von Fahrrad und hat zusätzlich:
  + federung (z. B. „Hardtail“ oder „Vollfederung“)
  + mietgebuehr ist teurer als beim Citybike

**rennrad.py (Unterklasse Rennrad)**

* Erbt von Fahrrad und hat zusätzlich:
  + gewicht (z. B. 8 kg)
  + mietgebuehr ist höher, da es für Profis gedacht ist

**4. Erweiterte Funktionen des Systems**

**kunde.py (Kunden-Klasse)**

* Attribute:
  + name
  + kunden\_id
  + gemietete\_fahrraeder (Liste der aktuell gemieteten Fahrräder)
  + gekaufte\_fahrraeder (Liste der gekauften Fahrräder)
* Methoden:
  + kaufe\_fahrrad(fahrrad) – Fügt ein gekauftes Fahrrad zur Liste hinzu.
  + miete\_fahrrad(fahrrad, tage) – Fügt ein gemietetes Fahrrad hinzu.
  + rueckgabe\_fahrrad(fahrrad, tage\_zu\_spaet) – Berechnet mögliche Strafgebühren.

**fahrradladen.py (Fahrradladen-Klasse)**

* Attribute:
  + bestand (Liste aller Fahrräder im Laden)
  + kunden (Liste aller registrierten Kunden)
* Methoden:
  + neues\_fahrrad\_hinzufuegen(fahrrad) – Fügt ein Fahrrad zum Bestand hinzu.
  + fahrrad\_verkaufen(kunde, fahrrad) – Verkauft ein Fahrrad an einen Kunden.
  + fahrrad\_vermieten(kunde, fahrrad, tage) – Vermietet ein Fahrrad an einen Kunden.
  + fahrrad\_zurueckgeben(kunde, fahrrad, tage\_zu\_spaet) – Bearbeitet die Rückgabe mit evtl. Strafgebühren.

**5. Interaktion im main.py**

* Erstellt **ein Citybike, ein Mountainbike und ein Rennrad**.
* Fragt den Nutzer:
  1. **Möchtest du ein Fahrrad kaufen oder mieten?**
  2. **Welches Fahrrad möchtest du auswählen?**
  3. **Wie viele Tage möchtest du es mieten?**
  4. **Beenden oder wieder zu 1.?**
* Berechnet und gibt die **Gesamtkosten** aus.

**6. Bonus-Aufgaben (optional)**

* **E-Bikes als neue Fahrradklasse hinzufügen** (höhere Mietkosten, Akku-Ladestand).
* **Fahrradwartung integrieren** (Ein Fahrrad kann nach X Vermietungen gewartet werden).
* **Kundenbewertungssystem** (Kunden können Fahrräder bewerten).