TravelBuddy

Franziska Schmidt Frieder Ullmann Marcel van der Heide Tim Vogel

Gliederung

- 1. Projektvorstellung
- 2. Programmstruktur
- > 3. Code
- ▶ 4. Soll vs. Ist



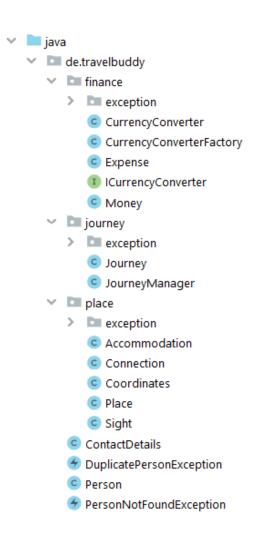
- > 5. Lessons Learned
- ► 6. Ausblick

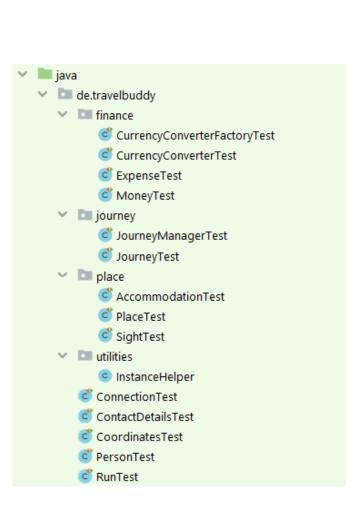
1. Projektvorstellung

TravelBuddy

- Reisebegleiter
- Inhalt: Reiseplanung
 - Unterkunft mit Kontaktdaten
 - Ausflugsziele mit Kontaktdaten
 - Verbindung zwischen Reise- oder Ausflugszielen
 - Berechnung der Entfernung in Zeit
 - Berechnung der Ausgaben
 - Umrechnung in andere Währungen

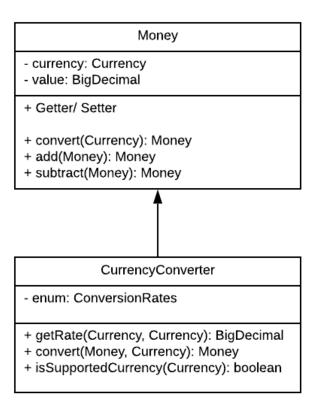
2. Programmstruktur





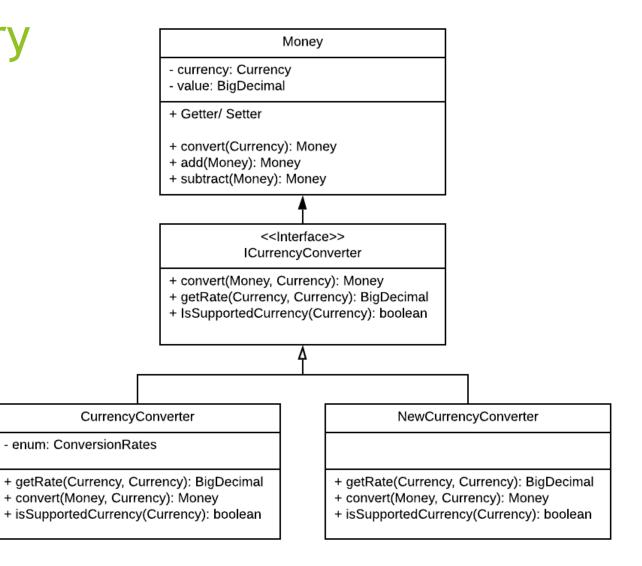
JourneyManager getJourneyNames: Map<String,Journey> + getJourneyNames():List<Strring> + journeyCount():int + addJourney (Journey): void 2. Programm Struktur removeJourney(Journey): Journey getJourney(String): Journey + totalCost(Currency):Money + totalCostOfPerson (Currency,Person):Money title: String places: List<Places> Person persons: List<Persons> + Getter/ Setter - name: String birthdate: LocalDate + addPlace(Place): void contactDetails: ContactDetails + removePlace(Place): void + addPerson(Person): void + Getter/ Setter + remove Person(Person): void + totalCost(Currency): Money + toalCostOfPerson(Person, Currency): Money ContactDetails + searchPerson(String):List<Person> <<enumeration>> + searchPlace(String):List<Place> AccommodationType phone: String - email: String + searchPlace(String,Class<T>):List<Place> - town: String HOTEL HOSTEL - street: String CAMPING streetnumber: int - ZIP: String COUCHSURF AIRBNB country: String + Getter/ Setter name: String coordinates: Coordinates contactDetails: ContactDetails arrive: LocalDateTime Accommodation Expense departure: LocalDateTime type:AccommodationType(Enum) expenses: Map<String, Expense> Money - title: String connectionsToNextPlace: List<Connection> - description: String involvedPersons: List<Person> + Getter/ Setter currency: Currency - price: Money value: BigDecimal - involvedPersons: List<Person> + Getter/ Setter - status(planned): enum -----+ Getter/ Setter - perPerson: boolean addPerson(Person): void + convert(Currency): Money + removePerson(Person): void + Getter/ Setter + add(Money): Money addExpense(Expense): void + subtract(Money): Money + removeExpense(Expense): void + getMoneyPerPerson(): Money + addConnection(Connection): void indoor: boolean + addPerson(Person): void + removeConnection(Connection): void + removePerson(Person): void + isIndoor(): boolean + isInvolved(Person): boolean + totalCost(Currency): Money + setIndoor(boolean): void + totalCostOfPerson(Currency): Money CurrencyConverterFactory <<Interface>> + findConnection(Place, Place): List<Connection> ICurrencyConverter + findExpense(String): List<Expense> create(): ICurrencyConverter + findPersons(String): List<Persons> Connection + convert(Money, Currency): Money + findPossibleConnectionsTo(Place, + getRate(Currency, Currency): BigDecimal - title: String LocalDateTime, int): List<Connection> + IsSupportedCurrency(Currency): boolean arrive: LocalDateTime departure: LocalDateTime start: Place end: Place expense: Expense Coordinates - used: boolean <<enumeration>> CurrencyConverter ConversionRate · longitude: double + Getter/ Setter enum: ConversionRates EUR, CHF, CNY, CZK, PLN, RUB, USD + Getter/ Setter + getDuration(): Duration getRate(Currency, Currency): BigDecimal conversionRate:BigDecimal + checkLatitude(latitude): boolean + convert(Money, Currency): Money + isSupportedCurrency(Currency): boolean + checkLongitude(longitude): boolean + getConversionRate():BigDecimal

3. Factory



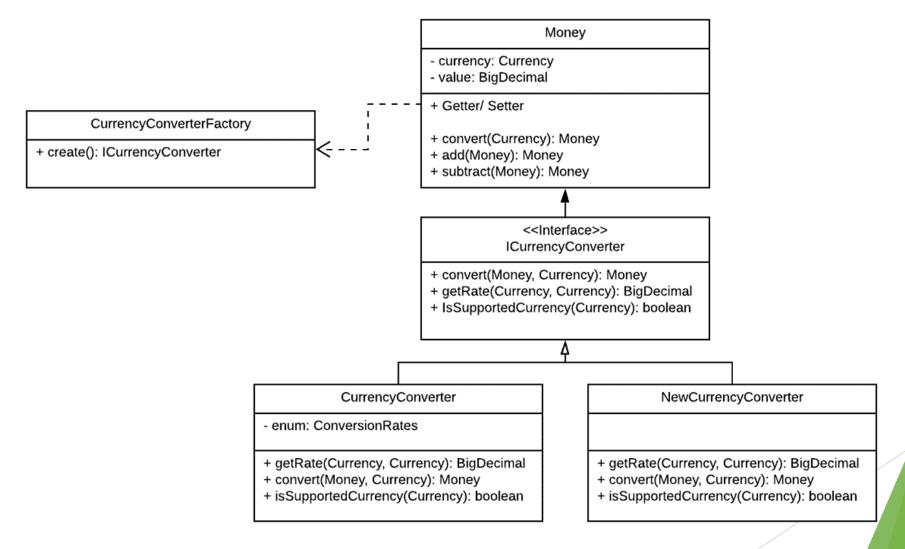
Problem: Umstellen auf eine neue, verbesserte CurrencyConverter-Klasse

3. Factory



Problem: Umstellung von "new CurrencyConverter()" auf "new NewCurrencyConverter"

3. Factory



3. Code

Live zeigen...



4. Soll VS. Ist

SOLL

- •Fahrten
- Personen
- Orte

- Reise
- •Reisedatum

IST

- ✓ Fahrten
- ✓ Personen
- **√**Orte
 - ✓ Sehenswürdigkeiten
 - ✓ Unterkünfte
- ✓ Reise
- ✓ Reisedatum
- ✓ Kosten (Währungsrechner, Ausgaben p.P.)

5. Ausblick

- Nicht nur local
- Multiuser
- Leute einladen / Reisen zuweisen
- ▶ Api einbinden, DB App, Google Maps, Wetter (in-/outdoor)...
- Kalender exportieren
- Graphische Darstellung der Reise (evtl. Google Maps)

6. Lessons Learned

- Regelmäßiger treffen
- GitLab:
 - öfter pushen
 - mehr Issues

Vielen Dank für ihre Aufmerksamkeit!