

Benutzerhandbuch

Allgemein

Das Projekt beinhaltet die Klassen **“Customer”**, **“Invoice”**, **“Performance”**, **“Play”**, **“StatementPrinter”**, **“TragedyPlay”**, **“ComedyPlay”**, **TestStatement**, **“HtmlStatement”**, **“PlayTest”**, **“StatePrinterTest”**, **“CustomerTest”**

Klassen

Customer

Die Klasse **“Customer”** repräsentiert einen Kunden. Die Klasse Customer hat einen Konstruktor, der dafür sorgt, dass dem jeweiligen Kunden einen Namen zugewiesen werden kann. Zudem kann der Name über eine getter Methode zurückgegeben oder über die überschriebene toString Methode ausgegeben werden.

Invoice

Die Klasse **“Invoice”** repräsentiert eine Rechnung. Eine Rechnung bezieht sich auf einen Kunden und besteht aus mehreren Aufführungen, die in der Liste **“performances”** gespeichert sind.

Die Klasse Invoice hat einen Konstruktor, der eine neue Rechnung mit dem übergebenen Kunden und der Liste an Aufführungen erstellt.

Die Methode **“calculateAmount”** berechnet die Summe der Kosten aller Aufführungen.

Die Methode **“calculateCredits”** berechnet die Summe an Rabatt-Credits für die gesamte Rechnung.

Performance

Die Klasse **“Performance”** repräsentiert eine konkrete Aufführung eines Stückes. Der Konstruktor erstellt eine neue Aufführung mit dem übergebenen Stück und der Anzahl an Zuschauern.

Play

Die abstrakte Klasse **“Play”** repräsentiert ein Theaterstück. Ein Stück hat einen Namen, der im Konstruktor zugewiesen wird. Die abstrakten Methoden **“calculateAmount(int audience)”** und **“calculateCredits(int audience)”** werden in den jeweiligen Unterklassen von **“Play”** implementiert, um den Preis der jeweiligen Aufführung mithilfe des Parameters **“audience”**, die Anzahl der Zuschauer, zu berechnen.

Die Methode "baseCredits(int audience)" gibt den Basispreis einer beliebigen Aufführung an. Der Basispreis ergibt sich aus Audienz-30, ist aber mindestens 0 Geldeinheiten.

TragedyPlay

Die Klasse "TragedyPlay" ist eine konkrete Implementierung eines Stückes. Die Klasse stellt dabei eine Tragödie dar.

Die Methode "calculateAmount" gibt den Gesamtbetrag der Rechnung aller Tragödien zurück.

Die Methode "calculateCredits" gibt dabei die Credits zurück, die der Kunde für die jeweilige Vorstellung bekommen hat.

ComedyPlay

Auch die Klasse "ComedyPlay" ist eine konkrete Implementierung eines Stückes. Die Klasse stellt dabei allerdings eine Komödie dar.

Die Methode "calculateAmount" gibt den Gesamtbetrag der Rechnung aller Komödien zurück.

Die Methode "calculateCredits" gibt dabei die Credits zurück, die der Kunden für die jeweilige Vorstellung bekommen hat.

PastoralPlay

Auch die Klasse "PastoralPlay" ist eine konkrete Implementierung eines Stückes.

Die Methode "calculateAmount" gibt den Gesamtbetrag der Rechnung vom Typ "pastoral" zurück.

Die Methode "calculateCredits" gibt dabei die Credits zurück, die der Kunden für die jeweilige Vorstellung bekommen hat.

HistoryPlay

Auch die Klasse "HistoryPlay" ist eine konkrete Implementierung eines Stückes. Die Klasse stellt dabei etwas Geschichtliches dar.

Die Methode "calculateAmount" gibt den Gesamtbetrag der Rechnung vom Typ "history" zurück.

Die Methode "calculateCredits" gibt dabei die Credits zurück, die der Kunden für die jeweilige Vorstellung bekommen hat.

StatementPrinter

“StatementPrinter” ist die Oberklasse zu “HTMLStatement” und “TextStatement” und gibt deren Methoden vor.

Die Klasse hat die Methode “print”, die das komplette Statement eines Invoice Objektes zurückliefert. Die Methode bekommt dazu eine Instanz auf das jeweilige Invoice Objekt.

Die abstrakte Methode “header” erzeugt den Header einer Ausgabe mit den Details zu einem Kunden.

Die abstrakte Methode “forEachPerformance” erzeugt die Ausgabe für eine Aufführung.

Die abstrakte Methode “footer” erzeugt die Ausgabe mit Details zur Rechnung.

TextStatementPrinter

Die Klasse “TextStatement” erbt von der abstrakten Klasse StatementPrinter und implementiert die abstrakte Methode “headerString” sowie “eachRentalString” und “footerString”.

HtmlStatementPrinter

Die Klasse “HtmlStatement” erbt auch von der abstrakten Klasse “StatementPrinter”. Innerhalb der Klasse wird implementiert, wie das Statement in HTML ausgegeben wird.

CustomerTest

Die Klasse “CustomerTest” testet die Klasse “Customer”. Sie implementiert die Methode “testCustomer”, die die Erzeugung eines Kunden testet.

StatementPrinterTest

Die Klasse “StatementPrinterTest” testet die Klasse “StatementPrinter”. Sie implementiert die Methode “testStatementPrinter”, die die Erzeugung eines StatementPrinters testet.

PlayTest

Die Klasse "PlayTest" testet die Klasse "Play". Sie implementiert die Methoden "testComedyPlay", "testHistoryPlay", "testTragedyPlay" und "testPastoralPlay", die die Erzeugung eines Stückes testen.