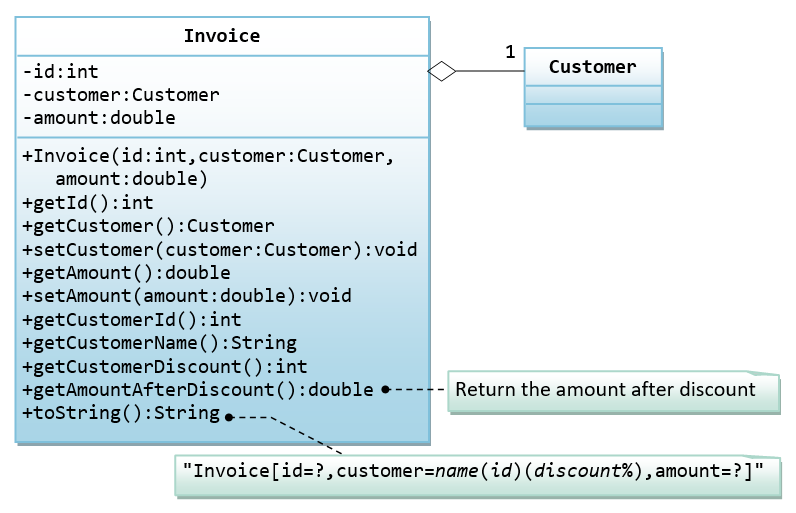
Privátní metody a datové složky mají -, veřejné +

#### The Customer and Invoice classes

A class called Customer, which models a customer in a transaction, is designed as shown in the class diagram. A class called Invoice, which models an invoice for a particular customer and composes an instance of Customer as its instance variable, is also shown. Write the Customer and Invoice classes.

# ClassCustomerJava.png



(poctivě ukradeno z <https://www3.ntu.edu.sg/> )

# Classroom

Vytvořte následující třídy: ClassRoom, Teacher a Student.   
Třída ClassRoom má následující datové složky:

-         teacher: Teacher

-         students: []Student

-         currentNumberOfStudents: int

Vytvořte k nim vlastnosti.

Třída ClassRoom má tyto veřejné metody:

+       ClassRoom(int)   
Konstruktor, který vytvoří novou instanci třídy ClassRoom. Parametrem konstruktoru je maximální počet studentů ve třídě. (k tomuhle nepotřebujeme další datovou složku – k čemu použijeme maximální počet studentů ve třídě?)

+       addStudent(Student): boolean      
metoda, která přidá studenta do třídy a vrátí boolean podle toho, jestli se to přidání zdařilo nebo ne (kdy se nezdaří vložení studenta do pole?)

Prozatím může metoda vkládat do pole na konec – která datová složka/vlastnost se k tomu dá použít?

+       removeStudent(int): boolean

+       toString(): String – vypíše třídního a studenty za použití metody ToString() příslušných tříd.

Třída Student má následující datové složky:

-         firstName: String

-         lastName: String

-         age: double

-         average: double

Třída Student má mít tyto veřejné metody:

+       Student(String, String, double, double)

+       toString(): String

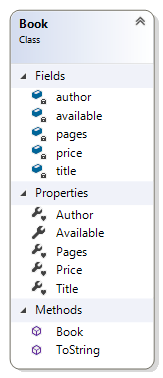
Třída Teacher nechť má datové složky a metody podle vaší fantazie. Například konstruktor třídy nevyplňuje učitele. Je tedy třeba učiteli třídu přiřadit metodou.

V metodě Main() ve třídě TestClass vytvořte několik žáků, zařaďte je do tříd, vypište je, zkuste si vypsat např. jméno učitele určité třídy, třetího žáka určité třídy.

V první fázi udělejte příklad tak, jak je – pak přidejte další metody podle vlastní fantazie. K takovým pokusům je ideální použít Git.

# Book

Vytvořte třídu Book, která bude popisovat knihu v knihovně.



Třída má datové složky:

* title:String
* author:Author
* pages:int
* price:double
* available:bool – určuje, jestli je kniha půjčená. Když se z knihovny vypůjčí, kniha tam fyzicky není, tedy má available nastavené na false.

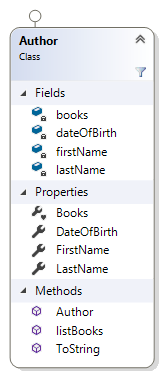
Vytvořte k nim vlastnosti.

Třída Book má veřejné metody:

+ Book (Author, title, pages, price)

+       toString():String

Třída Author, která popisuje autora:



Třída Author má mít tyto datové složky:

* firstName: String
* lastName:String
* books:List<Book>
* dateOfBirth:DateTime

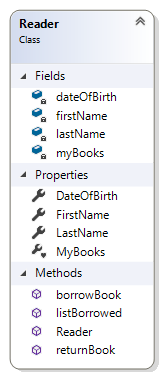
Třída Author má následující metody:

+       Author(String, String, DateTime)

+       listBooks():void – vypíše knihy autora

+       toString():java.lang.String

Třída Reader – popisuje čtenáře



Třída Reader má mít tyto datové složky:

* firstName: String
* lastName:String
* myBooks:List<Book> – kolekce vypůjčených knih
* dateOfBirth:DateTime

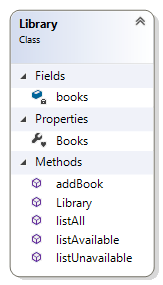
Metody:

+ borrowBook(): void – zkontroluje, jestli je kniha volná, pokud je, přesune ji do pole vypůjčených u čtenáře

+ returnBook():void - totéž opačně, pokud je kniha nedostupná, smaže ji z pole vypůjčených u daného čtenáře

+listBorrowed():void – vypíše knihy, které má čtenář aktuálně vypůjčené.

Třída Library



Datové složky:

* Books(List<Book>) – kolekce knih.

Nezapomeňte na vlastnost

Metody:

+ addBook(Book) – přidá knihu do kolekce - pokud nebude předem vytvořená instance knihy (čili bude instance anonymní, vytvořená v této metodě přes new Book(…) coby parametr metody), nepůjde pak půjčit, jedině přes Available[i].

+ listAvailable():void – vypíše aktuálně dostupné knihy

+ listUnavailable():void – vypíše aktuálně vypůjčené

+ listAll():void – vypíše všechny knihy. Využijte už vytvořenou vypisovací metodu třídy Book, ideálně ve tvaru: Karel Jaromir Erben – Kytice.

Vytvořte třídu TestLibrary, kde vytvoříte pár autorů a knih a knihovnu. Vyzkoušejte tvorbu knihy s new Author a new DateTime v konstruktoru (tedy bez instance autora a data).  Vytvořte totéž i s předem vytvořenými instancemi, zařaďte knihy do knihovny. Vypište všechny knihy, vypište aktuálně dostupné. Funguje vypsání aktuálně dostupných? Pokud ne a nevypíše se žádná kniha, nezapomněli jste něco někde nastavit? (na co se ve třídě inicializuje datová složka typu boolean?).

Pár knih vypůjčte, vypište, vraťte, vypište pro kontrolu. Znovu zkuste vypsat aktuálně dostupné, vypsat aktuálně vypůjčené.

To samé, co bylo řečeno výše: V první fázi udělejte příklad tak, jak je – pak přidejte další metody podle vlastní fantazie. Git je na tohle ideální kamarád.