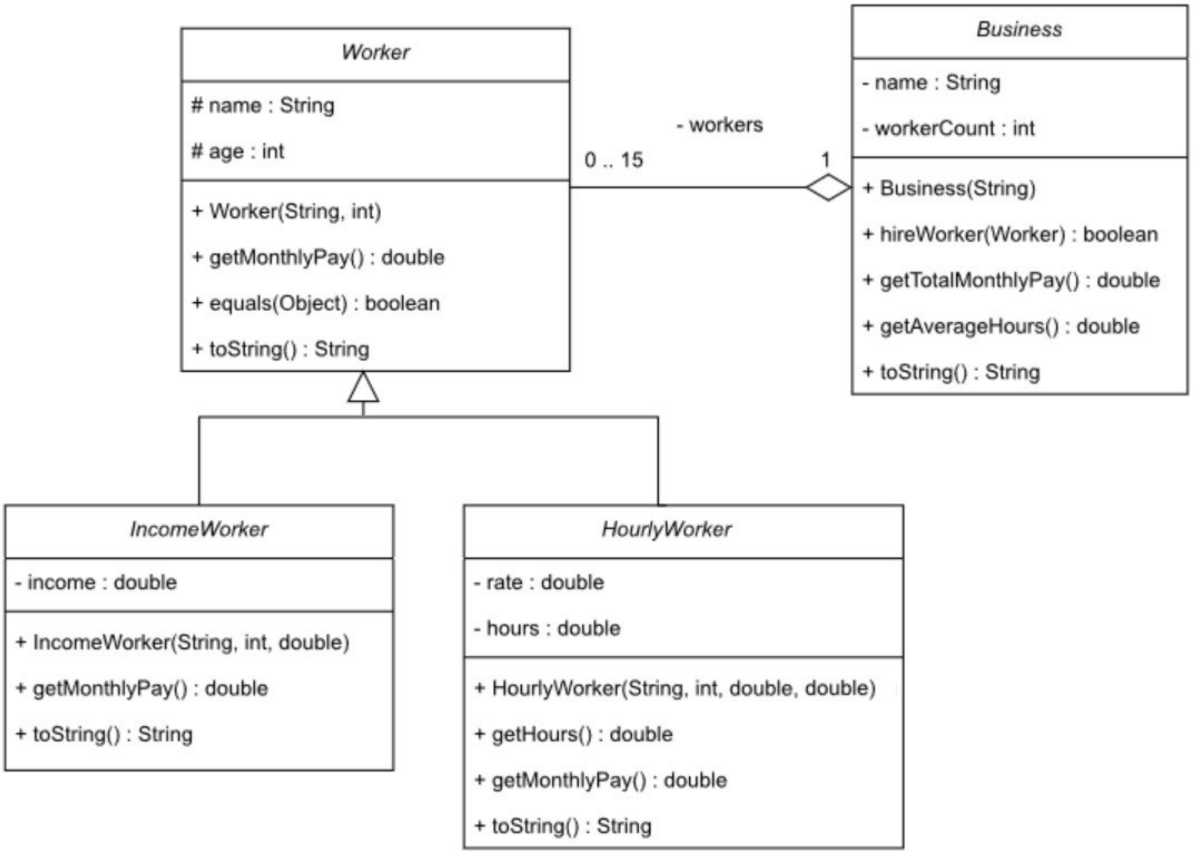
King Saud University
College of Computer and Information Sciences
Department of Computer Science
CSC113 – Computer Programming II – Abstract Class Lab

We want to write a program that manages the monthly worker payments for a business using the following UML diagram:



Class **Worker:**

- Instance Attributes:
  - ***name***: the name of the worker
  - ***age***: the age of the worker

- Methods:
  - ***Worker(name:String, age:int):*** constructor
  - ***equals(obj:Object):*** compares two workers based on their ***name*** and ***age*** and returns true if they are equivalent or false otherwise.

- ***toString():*** returns a string representation of the worker in the following format:
    - **Worker: *name*, age: *age***
- ***getMonthlyPay():*** returns the monthly pay for a worker:
    - IncomeWorker: the income paid in a month
    - HourlyWorker: the rate paid per hour of work in a month

Class **IncomeWorker:**

- Instance Attributes:
    - ***income***: yearly income

- Methods:
    - ***IncomeWorker(name:String, age:int, income:double):*** constructor
    - ***toString():*** returns a string representation of the worker in the following format:
        - **Worker: *name*, age: *age,* income*: income***

Class **HourlyWorker:**

- Instance Attributes:
    - ***rate***: the rate paid per hour
    - ***hours***: the number of hours worked in a month

- Methods:
    - ***HourlyWorker(name:String, age:int, rate:double, hours:double):*** constructor
    - ***getHours():*** returns ***hours***
    - ***toString():*** returns a string representation of the worker in the following format:
        - **Worker: *name*, age: *age,* rate*: rate,* hours*: hours***

Class **Business:**

- Instance Attributes:
    - ***name:*** name of the business
    - ***workers:*** array of Worker objects
    - ***workerCount:*** number of Worker objects in ***workers***

- Methods:
    - ***Business(name:String):*** constructor
    - ***hireWorker(w:Worker):*** adds ***w*** to the first available space of ***workers*** array if there's space and the worker isn't in the array already and returns true. Otherwise, returns false.
    - ***getTotalMonthlyPay***(): returns the total monthly pay for all workers in the business
    - ***getAverageHours***(): returns the average hours for all hourly workers in the business
    - ***toString():*** returns a string representation of the worker in the following format:
        - **Business *name* (*workerCount*):**
            - **Worker: *name*, age: *age,* income*: income***
            - **Worker: *name*, age: *age,* rate*: rate,* hours*: hours***
            - **…**

Exercise 1: Write classes **Worker, IncomeWorker, HourlyWorker & Business.**

Exercise 2: Write class **`BusinessTest`** that has a main method to test the functionalities of the classes:

- Create a business
- Hire 4 workers (see sample run)
- Print the business
- Print the total monthly pay for the business
- Print the average hours for hourly workers in the business

Sample run:

```
Business SICP (4):

Worker: Saleh, age: 23, income: 180000.0

Worker: Noor, age: 21, income: 120000.0

Worker: Jawaher, age: 25, rate: 80.0, hours: 160.0

Worker: Faisal, age: 22, rate: 60.0, hours: 200.0

Total monthly pay: 49800.0

Average hours: 180.0
```