**Every Task must be created as a new project.**

Task 1

* Create a package: com.itacademy.java.oop.basics
* Create a class: **Employee** which has fields: **name**, **surname**, **contract** **type**(**Full-Time, Part-Time**), **contract start date**, **salary** and **position.**
* Create a class **EmployeeManager** which has main method.
* Create an **Employee** object in **EmployeeManager**. Fill out all fields with data.
* Print all employee information.

Requirements:

* Employee should have **constructors** (with-args, without-args).
* Use encapsulation.
* Use enum where it‘s needed (enum constants must have display values and a method to create enum from display value).
* To print out all information create method in Employee class and call it from EmployeeManager.

Task 2

* Create a package, name: com.itacademy.java.oop.basics
* Create three classes: **Customer**, **Loan** and **LoansApplication**.
* Customer has fields: **name**, **surname**, **age**, **personal number, loans**.
* Loan has fields: **id**, **amount**, **loan type** (**leasing, consumer**) and **termination date**.
* LoansApplication has: main method.
* Create Customer object in LoansApplication and fill out all fields with data.
* Create two Loans (leasing, consumer loan) objects in LoansApplication and fill out all fields with data.
* Add two created loans to customer. Calculate both loans amount and print out information about customer and his loans including both loans amount.

Requirements:

* Use constructors (with-args, without-args).
* Use encapsulation.
* Use array for customer loans.
* Use enum where it‘s needed.
* To print out all information create method

Task 3

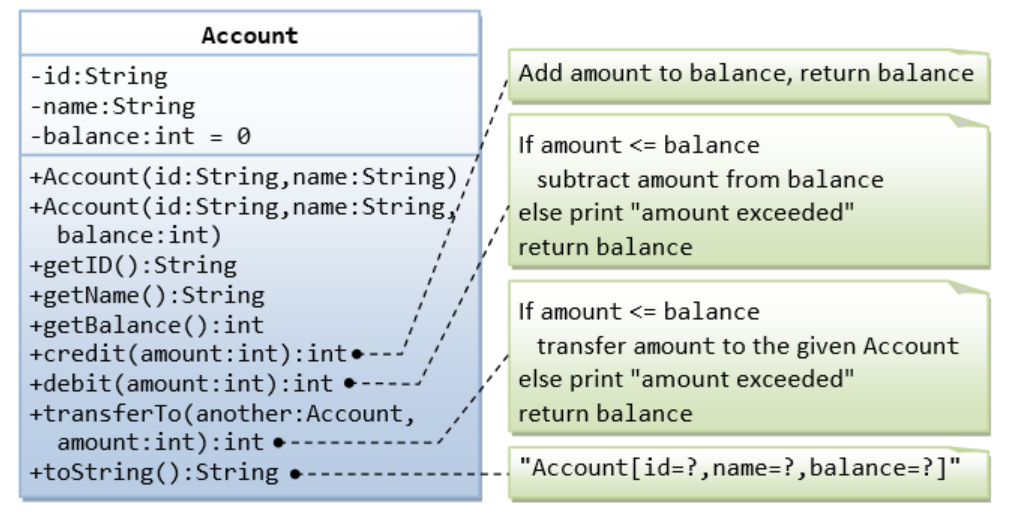
* Create a package, name: com.itacademy.java.oop.basics
* Create two classes: **Square** and **ShapeApplication**.
* Square has fields: **length** and **width** of Double type
* ShapeApplication has: main method.
* Create Square objects (more than one) in ShapeApplication and fill out all fields with data. Store shapes in an array.
* Iterate through the array of the shapes and check if shape is valid (values are not negative) if any of the value is negative print them out. For valid shapes print them out as well and calculate area and perimeter and print it out too.

Requirements:

* Use constructors (with-args, without-args).
* Use encapsulation.
* Perimeter and area calculation must be done using methods.
* To print out all information create the method

Task 4

* Create a package, name: com.itacademy.java.oop.basics
* Create a class named **AccountApplication** which has main method.
* Create class from this class diagram:



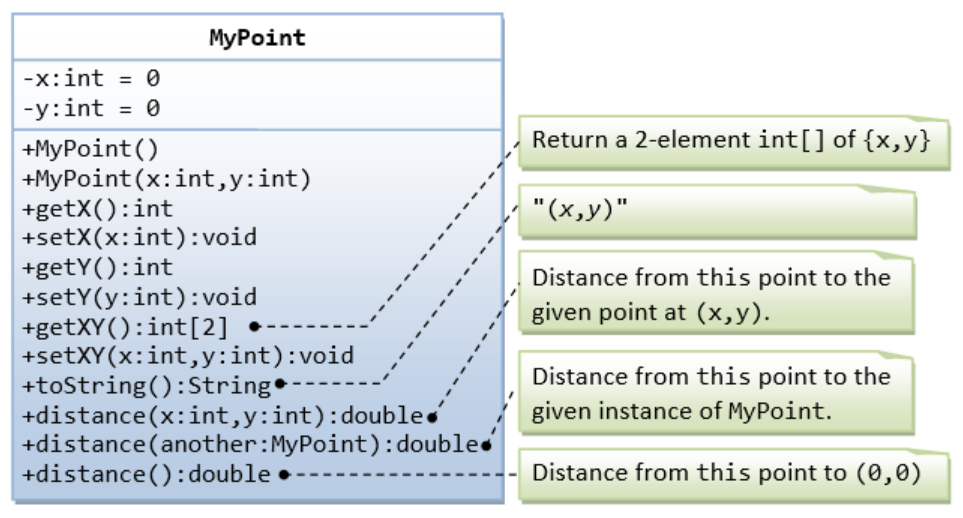
* In AccountManagerApplication create two accounts (starting balance for first is 100 for the second 50).

Requirements:

* Read the class diagram to identify functional requirements.
* You must execute actions in this order:
  + You must credit first account twice (20, 10). Print out the new balance.
  + You must credit second account once (10). Print out the new balance.
  + You must debit first account with 500.
  + You must transferTo from second account to first account 70.
  + You must debit first account 30. Print out the new balance.
  + You must transferTo from first account to second account 10. Print out both accounts balances.
  + Print out both accounts final information after actions using the **format provided by the class diagram**.

Task 5

* Create a package, name: com.itacademy.java.oop.basics
* Create a class named **PointApplication** which has main method.
* Create MyPoint class from this class diagram:



* Tip: to calculate distance use Math library.

Requirements:

* You must create two points.
* Read the class diagram to identify functional requirements.
* You must use each method at least once and print out the response if there if anything to be printed.
* You must use a loop to print response from getXY method for first point.
* You must print second point getXY data x and y coordinates without using loop.

Task 6

* Create a package, name: com.itacademy.java.oop.basics
* Create class **TravelApplication** with a main method.
* Create class **Person** with fields **name**, **surname**, **gender**, **age**.
* Create class **Vehicle** with fields **name, brand, fuel (**currentvehicle fuel in liters)**, consumption** (how much liters of fuel per 100km)
* Create class **TravelDestination** with fields **name**, **city**, **distance (**in km).
* Create class **Family** with fields **array with Person instances, Vehicle instance, TravelDestination instance.**
* Create static class **TravelManager** with methods changeDestination(Family family, TravelDestination destination): Family – should change families destination to a new destination. If current family’s destination is the same as new nothing should be change for family. travel(Family family): void – should print out if family was able to travel to their destination (to determinate if they were able to, you need to calculate how far their vehicle can travel with the current setup and how far is the travel destination, you will probably need helper methods in TravelManager). If family was not able to travel to their destination print out how much fuel they need in their vehicle to be able to reach their destination. printFamilyMembers(Family family):void you must print all of the family members.

Requirements

* You must use immutability.
* You must determine which methods should be public which should be private.
* You must create two families with at least two family members in them. One family must be able to travel to their destination and the other should not be able to.
* You must use all TravelManager methods at least once.
* Use toString() method for printing objects information.