Packages
   - It is a container
   - It is used to group logical related types together

eattendance system	package util
class Student{	class Date{
}	}
class Employee{	class Time{
}	}
package entity	

Dmart ->
Biscuit
Snacks
Grocery

Demo02

```
bin                          src
  p1                           Time.java
      Time.class                   - package p1;
  p2                           Program.java
      Program.class                - package p2;

                             javac -d ..\bin Time.java

                             javac -d ..\bin Program.java
```
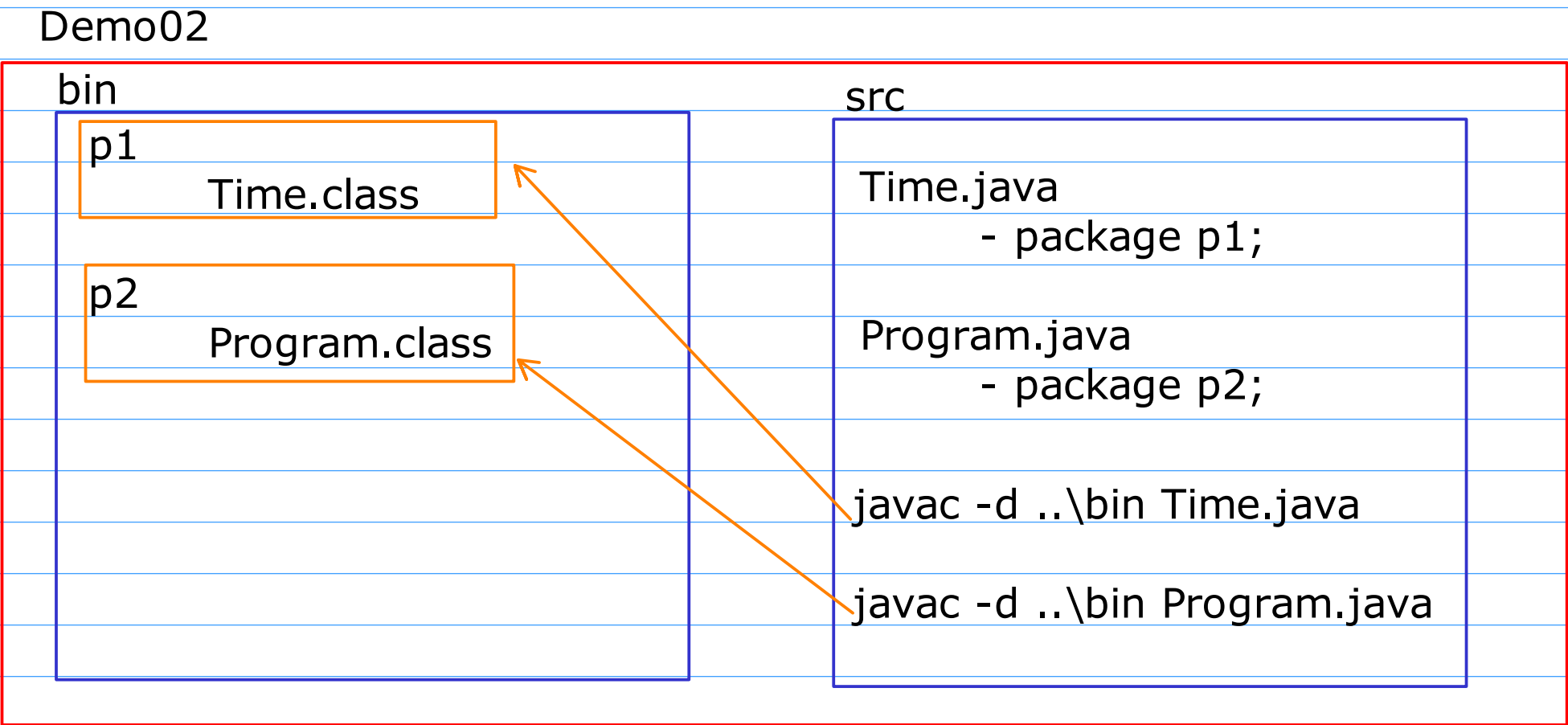
Editor	IDE - Integrated Development Environment
Notepad	- created for specific Tech or lang for building projects
gedit	- automate the build process
Notepad++	
vscode	- eclipse(Java SE)
	- intellij (Java SE)
.txt , .c, .cpp, .java, .js, .py,	- STS (Java SE and Java EE) Spring Framework
manual efforts	
	- pycharm
	- Android Studio
	- Visual Studio

package
- the package name should be kept in small case
- the name of the package should be on the company domain name in reverse order

domain name - sunbeam.com
package name - com.sunbeam

com.sunbeam.eattendancesystem.enity
com.sunbeam.eattendancesystem.utils
com.sunbeam.eattendancesystem.tester

privtate
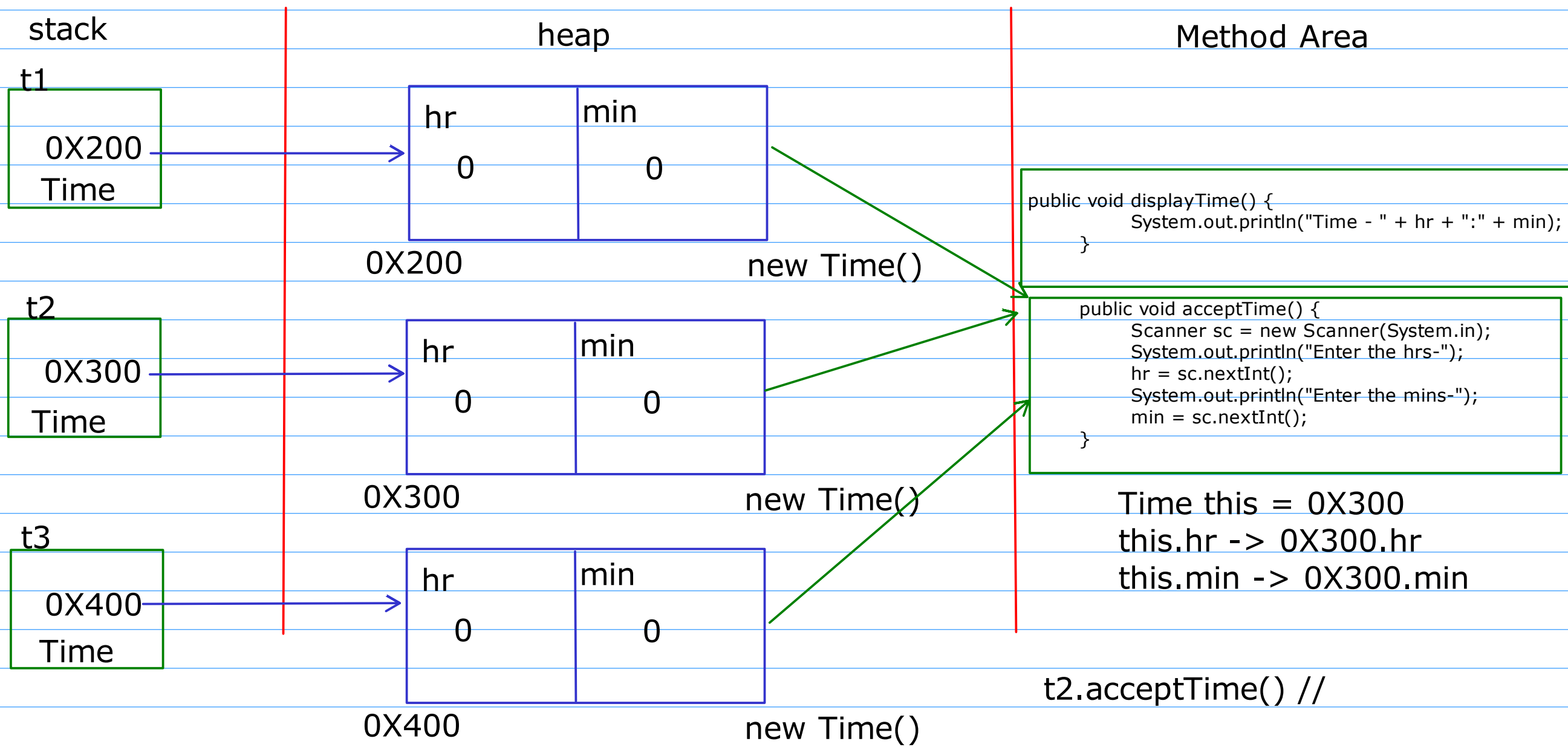- only within the class
default
- only within the same package
protected
- in the same package
- in the subclass(child class) in other pakages
public
- every where

```java
class Time{
    // fields

    // methods
}
```

stack   heap   Method Area

t1

| 0X200 | hr | min |
| Time | 0 | 0 |

0X200   new Time()

```java
public void displayTime() {
    System.out.println("Time - " + hr + ":" + min);
}
```

t2

| 0X300 | hr | min |
| Time | 0 | 0 |

0X300   new Time()

```java
public void acceptTime() {
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter the hrs-");
    hr = sc.nextInt();
    System.out.println("Enter the mins-");
    min = sc.nextInt();
}
```

Time this = 0X300
this.hr -> 0X300.hr
this.min -> 0X300.min

t3

| 0X400 | hr | min |
| Time | 0 | 0 |

0X400   new Time()

t2.acceptTime() //

this reference
- it is a reference that is passed internally to all the non static methods of the class
- it points to the current calling object

Types of methods
1. Constructor
2. Setter
3. Getter
4. Facilitator

#Constructor
- It is a special method of a class
- Why it is so special?
- The name of ctor is same as that of class name
- It does not have any return type
- It gets automatically called when the object is created.

Types of constructor
1. Default / Paramaterless ctor
2. Parameterized ctor