به نام خدا

پاسخنامه تمرین دوم درس برنامه نویسی پیشرفته

نیمسال دوم ۱۳۹۹-۱۴۰۰

فهرست سوالات

سوال اول ۲
سوال دوم ۴
سوال سوم ۵
سوال چهارم ۸

سوال اول

الف)

- State: هر شیء دارای حالت یا State است که با مقادیر Field های خود تعریف می‌شود؛ در واقع مقادیر Field های آن معادل حالت شیء است که در حال حاضر در آن قرار دارد.
- Behavior: رفتار یا Behavior یک شیء به Method های آن اشاره دارد که چه کارهایی می‌تواند انجام بدهد.
- Identity: شیءها هر کدام هویت یا Identity خاص خود را دارند که آنها را از هم جدا می‌کند که بعد از Instantiate کردن آن ساخته می‌شود و بعد از آن هیچوقت تغییر نمی‌کند. می‌توان هویت یک شیء را شماره آدرسی از حافظه در نظر گرفت که در طول اجرای برنامه به آن اختصاص پیدا می‌کند.

ب)

راه اصلی تعامل شیءها با یکدیگر، صدا زدن Method های یکدیگر است.

ج)

- Overloading: در جاوا می‌توان چند Method مختلف با یک اسم ولی با امضا یا Signature در یک کلاس داشت، که امضای یک Method همان تعداد پارامترهای ورودی اش و همچنین نوع آن‌ها است (ترتیب نوع‌ها اهمیت دارد).
- Casting: یک فرایند است که یک نوع داده را (به دو روش دستی و خودکار) به نوع داده دیگری تبدیل می‌کند.
- Modularization: ماژولار بودن خاصیت سیستمی است که به مجموعه ای از واحدهای منسجم تقسیم شده است، هر واحد کار مختص خود را انجام می‌دهد و وابستگی کمی به هم دارند که به هم متصل می‌شوند.
- Abstraction: به پنهان کردن پیچیدگی‌های یک سیستم یا یک Class از کسانی که از آن استفاده می‌کنند، گفته می‌شود؛ برای مثال در برنامه نویسی شیء گرا، هر Method کار خاصی را انجام می‌دهد ولی کسی که آن را صدا می‌زند، فقط باید بداند که آن Method چه کاری انجام می‌دهد و چه پارامترهایی می‌خواهد و لازم نیست بداند که درون تابع چه دستوراتی اجرا می‌شوند.

د)

Immutable Object به شیء ای گفته می‌شود که State یا حالتش از زمان ساخته شدنش، قابل تغییر نیست و تنها راه برای داشتن یک شیء از آن کلاس با حالتی متفاوت، ساختن یک شیء دیگر است. برای مثال در جاوا، کلاس String و تمام Wrapper Class ها (Integer، Byte، ...) از نوع Immutable هستند.

سوال دوم

1. If we write our own constructor, the compiler also creates a default constructor in Java. **(false)**
   If we don't define a constructor in a class, then compiler **creates** default constructor (with no arguments) for the class. And if we write a constructor with arguments or no-arguments then the compiler does not create a default constructor.

2. Actions on ArrayList may be slower than standard arrays. **(true)**

3. We can have an ArrayList of primitive types like int, char and … **(false)**
   Primitive data types cannot be stored in ArrayList but can be in Array**.** ArrayList expects only Objects data types and all the operations done in Collections, like iterations, can be performed only on Objects and not Primitive data types.

4. We can define more than one constructor in a class. **(true)**

5. In an instance method or a constructor, "this" is a reference to the current object. **(true)**

6. Local variables in Java must be definitely assigned to before they are accessed, or it is a compile error. **(true)**

7. Variables declared in the body of a particular method are known as instance variables and can be used in all methods of the class. **(false)**
   Such variables are called local variables and can be used only in the method in which they're declared.

8. The values assigned inside the constructor overwrite the values initialized with declaration. **(true)**

سوال سوم

الف)

۱. بله ولی در صورتی که تایپ پارامترها حتما متفاوت باشند. زیرا اگر تنها return type تغییر کند ولی تایپ پارامترها و اسم متودها یکسان باشند، overloading محسوب نمی‌شود و با compile error مواجه می‌شویم.

باید توجه شود که در overloading، متدها باید نام یکسان ولی پارامترهای ورودی متفاوتی داشته باشند.

۲.

Sr. No.	Key	Constructors	Methods
1	Purpose	Constructor is used to create and initialize an Object .	Method is used to execute certain statements.
2	Invocation	A constructor is invoked implicitly by the System.	A method is to be invoked during program code.
3	Invocation	A constructor is invoked when new keyword is used to create an object.	A method is invoked when it is called.
4	Return type	A constructor can not have any return type.	A method can have a return type.
5	Object	A constructor initializes an object which is not existent.	A method can be invoked only on existing object.
6	Name	A constructor must have same name as that of the class.	A method name can not be same as class name.
7	Inheritance	A constructor cannot be inherited by a subclass.	A method is inherited by a subclass.

۳. بله، در جاوا توانیم کانستراکتورهای متفاوت با نام‌های یکسان ولی با پارامترهای ورودی متفاوت داشته باشیم. دلیل استفاده از constructor overloading در این است که گاهی می‌خواهیم یک object را در نوع‌های مختلف initialize کنیم. به عنوان مثال یک جا می‌خواهیم صرفا از آن، یک نمونه بدون هیچ اطلاعات اضافه ای ساخته شود و در جایی دیگر می‌خواهیم یک نمونه از آن را با داشتن اطلاعاتی از آن بسازیم، به همین خاطر به constructor هایی با نوع‌های مختلف نیاز داریم.

ب)

۱. در متد main، خط اول، متغیر id به صورت string داده شده است در حالی که باید integer باشد:

```
person p1 = new person("Ted", "Mosby", 123456);
```

۲. نام کلاس person باید به Person تغییر پیدا کند.

۳. برای رعایت تمیزی و خوانایی کد، نام فیلد FirstName باید به صورت camel case نوشته شود، یعنی firstName.

۴. برای رعایت تمیزی و خوانایی کد، نام فیلد last_name بهتر است به صورت lastName نوشته شود.

۵. در constructor، هیچ return type نداریم و باید public void person به public person تغییر پیدا کند.

۶. assignment ها در constructor باید به این صورت نوشته شوند زیرا نام پارامترهای ورودی با فیلدها یکسان است:

```
this.FirstName = FirstName;
this.last_name = last_name;
this.id = id;
```

۷. در متد express، در هنگام overload کردن با عوض کردن return type به اروری بر نمی‌خوریم ولی بهتر است این کار انجام نشود.

ورژن صحیح کد:

```
package com.company;

public class Main {

   public static void main(String[] args) {
       Person p1 = new Person("Ted", "Mosby", 123456);
       p1.express();
       p1.express("Happy");
       p1.print();
   }

   static class Person {
       private final String firstName;
```

```
    private final String lastName;
    private final int id;

    public Person(String firstName, String lastName, int id){
        this.firstName = firstName;
        this.lastName = lastName;
        this.id = id;
    }

    public void express(){
        System.out.println("I feel neutral");
    }

    public void express(String state) {
        System.out.println("I feel " + state + " today");
    }

    public void print(){
        System.out.println("Person{" +
                "firstName='" + firstName + '\'' +
                ", lastName='" + lastName + '\'' +
                ", id=" + id +
                '}');
    }
}
```

سوال چهارم

۱)

تفاوت‌های بین arraylist و linkedlist:

- در arraylist از آرایه داینامیک (با قابلیت تغییر سایز) استفاده می‌شود ولی در لینکدلیست از doubly linked list استفاده می‌شود.

- Arraylist implements List but linkedlist implements both List and Queue
- Arraylist is slow at manipulation because it needs bit manipulation but Linkedlist is faster cause it doesn't need bit manipulation
- Retrieving data is faster in Arraylist due to having indexed positions
- Arraylist -> faster in storing and retrieving
- Linkedlist -> faster in manipulating

تفاوت بین array و arraylist:

Array	ArrayList
An array is a dynamically-created object. It serves as a container that holds the constant number of values of the same type. It has a contiguous memory location.	The ArrayList is a class of Java Collections framework. It contains popular classes like Vector, HashTable, and HashMap.
Array is static in size.	ArrayList is dynamic in size.
An array is a fixed-length data structure.	ArrayList is a variable-length data structure. It can be resized itself when needed.

Initialization	It is mandatory to provide the size of an array while initializing it directly or indirectly.	We can create an instance of ArrayList without specifying its size. Java creates ArrayList of default size.
Performance	It performs fast in comparison to ArrayList because of fixed size.	ArrayList is internally backed by the array in Java. The resize operation in ArrayList slows down the performance.
Primitive/ Generic type	An array can store both objects and primitives type.	We cannot store primitive type in ArrayList. It automatically converts primitive type to object.
Iterating Values	We use for loop or for each loop to iterate over an array.	We use an iterator to iterate over ArrayList.
Type-Safety	We cannot use generics along with array because it is not a convertible type of array.	ArrayList allows us to store only generic/ type, that's why it is type-safe.
Length	Array provides a length variable which denotes the length of an array.	ArrayList provides the size() method to determine the size of ArrayList.
Adding Elements	We can add elements in an array by using the assignment operator.	Java provides the add() method to add elements in the ArrayList.
Single/ Multi-Dimensional	Array can be multi-dimensional.	ArrayList is always single-dimensional.

موارد بالا همگی قابل قبول است

https://www.javatpoint.com/difference-between-array-and-arraylist

۲) با توجه به اینکه طبق صورت سوال تغییرات تعداد بیماران(مرخص یا اضافه شدن) زیاد است بهتر است از linkedlist استفاده شود.

سرعت اضافه یا کم کردن نود در لینکدلیست بیشتر از arraylist است زیرا به شیفت دادن ایندکس‌ها نیاز ندارد.

از آنجایی که محدودیتی در تعداد بیماران نیست و تعداد متغیر است از آرایه نباید استفاده شود.

۳)

The ArrayList in java does not provide the checks for duplicate references to the same object. Therefore, we can insert the same object or reference to a single object as many times as we want. If we wish we can check if an element already exists in ArrayList or not with the help of the contains() method.

۴) در هنگام تعریف آرایه یک سایز ثابت به آن اختصاص می‌یابد که در فیلد آرایه ذخیره می‌شود. arraylist یک کالکشن پویا است و سایز آن متغیر است و متد size تعداد آبجکت‌های موجود در کالکشن را بر می‌گرداند