

دستور کار کارگاه برنامه نویسی پیشرفته

به نام خدای رنکین کمان



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

دستور کار کارگاه کامپیوتر

آرین محسنی

فروردین 1402

فهرست

2	فهرست
3	سیستم رأیگیری
3	کلاس Person
3	کلاس Vote
5	کلاس Voting
6	کلاس VotingSystem
8	کلاس Main

سیستم رأیگیری

در این تمرین با ساختن کلاس های مختلف مانند کلاس Voting و VotingSystem و Person و تعامل بین آن ها و ساختن اشیا از آن ها این کار را انجام می دهیم.

کلاس Person

این کلاس شامل ۲ فیلد و تعدادی متود است به شکل زیر:

```
class Person {  
  
    private String firstname;  
    private String lastname;  
  
    public Person(String firstname,String lastname){  
        this.firstname = firstname;  
        this.lastname = lastname;  
    }  
  
    public String getFirstName(){  
        return firstname;  
    }  
  
    public String getLastName(){  
        return lastname;  
    }  
  
    public String toString(){  
        return firstname + "-" + lastname;  
    }  
}
```

کلاس Vote

این کلاس شامل 2 فیلد به صورت final و چند متود (Structor و getter و setter است)

```
class Vote {

    private final Person voter;
    private final String date;

    public Vote (Person voter,String date){
        this.voter = voter;
        this.date = date;
    }
    public Person getVoter() {
        return voter;
    }
    public String getDate() {
        return date;
    }

    @Override
    public String toString() {
        return "" + voter + "-" + date ;
    }

    @Override
    public boolean equals(Object o){

        if ( this == o) return true;
        if ( !(o instanceof VotingSystem)) return false;
        VotingSystem that = (VotingSystem) o;

        return Objects.equals(getVotingList(),that.getVotingList());
    }
    @Override
    public int hashCode(){
        return Objects.hash(getVotingList());
    }
}
```

کلاس Voting

در این کلاس از چند فیلد (hashmap و arraylist) استفاده میکنیم و سپس متود های getter و

setter

```
class Voting {  
  
    private int type;  
    private String question;  
    private HashMap<String, HashSet<Vote>> choices;  
    private boolean isAnonymous;  
    private ArrayList<Person> voters;  
  
    public String getQuestion() {  
        return question;  
    }  
  
    public ArrayList<String> getChoices() {  
        return new ArrayList<>(choices.keySet());  
    }  
  
    public Voting(int type, String question, boolean isAnonymous) {  
        this.type = type;  
        this.question = question;  
        this.isAnonymous = isAnonymous;  
        choices = new HashMap<>();  
        voters = new ArrayList<>();  
    }  
  
    public void setQuestion(String question) {  
        this.question = question;  
    }  
  
    public void createChoice(String choice) {  
        choices.put(choice, new HashSet<>());  
    }  
  
    public void vote(Person voter, ArrayList<String> voter_choices){  
        if (type == 0) {
```

دستور کار کارگاه برنامه نویسی پیشرفته

```
        System.out.println("You can only select one choice for this voting.");
    }
    if (!voters.contains(voter)) {
        voters.add(voter);
    }
    for (String choice : voter_choices) {
        if (!choices.containsKey(choice)) {
            System.out.println("Invalid choice: " +
choice);
        }

        Vote vote = new Vote(voter, "");

        if (!isAnonymous) {
            vote = new Vote(voter, "some date");
        }

        choices.get(choice).add(vote);
    }
}
```

کلاس VotingSystem

این کلاس متشکل از یک arraylist برای لیست رای دهی و متود های مربوط به ساختن رای و چاپ آن ها می باشد.

```
class VotingSystem{

    private ArrayList<Voting> votingList;

    public VotingSystem(ArrayList<Voting> votingList) {
        this.votingList = votingList;
    }

    @Override
```

دستور کار کارگاه برنامه نویسی پیشرفته

```
public int hashCode() {
    final int prime = 31;
    int result = 1;
    result = prime * result + ((votingList == null) ? 0 :
votingList.hashCode());
    return result;
}

@Override
public boolean equals(Object obj) {
    if (this == obj)
        return true;
    if (obj == null)
        return false;
    if (getClass() != obj.getClass())
        return false;
    VotingSystem other = (VotingSystem) obj;
    if (votingList == null) {
        if (other.votingList != null)
            return false;
    } else if (!votingList.equals(other.votingList))
        return false;
    return true;
}

public void createVoting(String question,boolean isAnonymous,int
type,ArrayList<String> choices){
    Voting voting = new Voting(type,question,isAnonymous);
    for (String choice : choices) {
        voting.createChoice(choice);
    }
    votingList.add(voting);
}
public ArrayList<Voting> getVotingList() {

    return votingList;
}
public void printResults(int index){
}
}
```

کلاس Main

```
public class vote {  
    public static void main(String[] args) {  
  
        VotingSystem test = new VotingSystem(new ArrayList<>());  
        Vote vote = new Vote(new Person("ali", "mohammadi"), "some date");  
        System.out.println(vote);  
    }  
}
```