

1. ჩამოთვლილთაგან რომელი მოსაზრებაა მცდარი?
  - ა) IO Streams არის ინფორმაციის წაკითხვა/ჩაწერის ნაკადები. Input Stream გამოიყენება ინფორმაციის წყაროდან წასაკითხად, ხოლო Output Stream ინფორმაციის წყაროში ჩასაწერად
  - ბ) InputStream-ებთან მუშაობისას ხშირად საჭიროა IOException-ის ან მისი რომელიმე შვილი კლასის დამუშავება, რადგან InputStream-ების უმეტესი მეთოდი თუ კონსტრუქტორი გაისვრის აღნიშნულ Exception-ებს
  - გ) IO სტრიმთან მუშაობის დასრულების შემდეგ აუცილებელია შექმნილი სტრიმის დახურვა, წინააღმდეგ შემთხვევაში დაფიქსირდება კომპილაციის შეცდომა - კოდი არ დაკომპილირდება
  - დ) FileReader და FileWriter არის Character Stream-ის მაგალითები, პირველი გამოიყენება ინფორმაციის წაკითხვისთვის ხოლო მეორე გატანისთვის
  
2. ჩამოთვლილთაგან რომელი მოსაზრებაა მცდარი?
  - ა) ინკაფსულაცია მიიღწევა კლასების და ობიექტების დახმარებით, მონაცემების მათში ინკაფსულირებით. დაცულობის თვალსაზრისით მიღებულია, რომ კლასის ატრიბუტები იყოს private და მასზე წვდომისთვის შეიქმნას ცალკე public მეთოდები
  - ბ) მემკვიდრეობითობა არის პოლიმორფიზმის თანდაყოლილი პრინციპი, ის ვერ იარსებებს პოლიმორფიზმის გარეშე, რადგან მემკვიდრე კლასი ყოველთვის ცვლის მშობლის ყველა მეთოდს
  - გ) პოლიმორფიზმი არის პრინციპი, რომელიც იძლევა საშუალებას, რომ არსებობდეს ერთი გარეგნული სახე, თუმცა გააჩნდეს რამდენიმე განსხვავებული დანიშნულება, ეს ყველაზე მეტად გადაფარვით(Overriding) მიიღწევა
  - დ) აბსტრაქციის მისაღწევად გამოიყენება აბსტრაქტული კლასები და ინტერფეისები, მათი დახმარებით შესაძლებელია მეთოდის აღწერის და იმპლემენტაციის დაცალკეება
  
3. ჩამოთვლილთაგან რომელია სწორი JDK, JRE და JVM-თან მიმართებაში
  - ა) JVM არის ჯავას ვირტუალური მანქანა რომელიც შედგება JDK-სა და JRE-სგან, რომელიც გამოიყენება კოდის კომპილაციისთვის
  - ბ) JDK შედგება დეველოპერული საშუალებებისგან და JRE-სგან, თავის მხრივ JRE შეიცავს JVM-ს რომელიც არის ბაიტ კოდის გამშვები ვირტუალური მანქანა
  - გ) JRE და JDK არის ჯავას ვირტუალური მანქანები, რომლებიც პასუხისმგებელი არიან კოდის კომპილაციასა და გაშვებაზე. JVM გამოიყენება დამატებითი ფუნქციონალისთვის, მაგალითად დოკუმენტაცია, დებაგინგი და ა.შ.
  - დ) JVM, JRE და JDK არის ერთიდაიგივე ტერმინი და სამივე პასუხისმგებელია Java-ს სორს კოდის კომპილაციაზე
  
4. ჩამოთვლილთაგან რომელი მოსაზრებაა მცდარი?
  - ა) Java-ს გააჩნია ოთხი პლათფორმა: Java SE, Java EE, Java ME და JavaFX

ბ) არსებობს ოთხი ტიპის Inner კლასები: Nestet Inner Class, Statis Nester Inner Class, Method Local Inner Class და Anonymous Inner Class

გ) ობიექტზე ორიენტირებულ დაპროგრამებაში არის ოთხი ძირითადი პრინციპი: ინკაფსულაცია, მემკვიდრეობითობა, სერიალიზაცია და აბსტრაქცია

დ) ზოგადად არსებობს რამდენიმე სახის მონაცემთა სტრუქტურა, მაგალითად: წრფივი მონაცემთა სტრუქტურები, ხეები, ჰეშ სტრუქტურები და ა.შ.

5. ჩამოთვლილთაგან რომელი მოსაზრებაა სწორი?

ა) შეუმოწმებელი(Unchecked) გამონაკლისი შემთხვევა არ ფიქსირდება კომპილაციის დროს, შესაბამისად არ არის აუცილებელი მისი try-catch-ით დამუშავება. შემოწმებადი(Checked) გამონაკლისი შემთხვევა აუცილებლად უნდა დამუშავდეს, წინააღმდეგ შემთხვევაში მოხდება კომპილაციის შეცდომა

ბ) try-catch ბლოკს ყოველთვის მოყვება finally ბლოკი, რათა აუცილებლად შესრულდეს „საბოლოო“ ქმედებები

გ) საკუთარი გამონაკლისი შემთხვევის კლასის შექმნა შეუძლებელია, შესაძლებელია მხოლოდ არსებული კლასების გამოყენება

დ) try ბლოკის შემდეგ უნდა მოდიოდეს ზუსტად ერთი catch ბლოკი, შესაძლებელია სწორედ ამ catch-ში გაწერილი გამონაკლისი შემთხვევის დამუშავება

6. ჩამოთვლილთაგან რომელი მოსაზრებაა მცდარი?

ა) Object კლასს გააჩნია მეთოდი finalize, რომელიც გამოიძახება garbage collector-ის მიერ ობიექტის განადგურებისას

ბ) Object კლასის equals მეთოდი აბრუნებს boolean მნიშვნელობას, ის ერთმანეთს ადარებს ორ ობიექტს. მიზანშეწონილია კლასებში მისი გადაფარვა

გ) როდესაც System.out.println-ს გადაეცემა რაიმე ობიექტი, ის იძახებს ამ ობიექტის კლასის toString მეთოდს, რომელიც იდურად უნდა აბრუნებდეს კლასის აღწერის სტრიქონულ მნიშვნელობას

დ) ნებისმიერი კლასის შექმნისას აუცილებელია გადაიფაროს Object კლასის მეთოდები equals, hashCode და toString, წინააღმდეგ შემთხვევაში დაფიქსირდება კომპილაციის შეცდომა

7. ჩამოთვლილთაგან რომელი მოსაზრებაა სწორი?

ა) String-ის რამდენიმე მეთოდი თავის თავშივე ცვლის სტრიქონის მნიშვნელობას, მაგალითად t.concat(s) ჩანაწერის შესრულების შემდეგ t სტრიქონს ემატება s სტრიქონი

ბ) StringBuilder-ის და StringBuffer-ის მეთოდები არის სინქრონიზირებული, შესაბამისად მათ შორის არანაირი განსხვავება არ არის

გ) თუ String ობიექტი იქმნება new-ს გამოყენებით(String s = new String("Test")), ის ინახება heap-ში, ხოლო თუ იქმნება მინიჭებით(String s = "test") - String constant pool-ში

დ) String არის სიმბოლოების მასივი, მისი გამოყენება შესაძლებელია როგორც მასივი, მაგალითად `t` არის რაიმე String, `s[0]` დააბრუნებს მის პირველ სიმბოლოს

8. ჩამოთვლილთაგან რომელი მოსაზრებაა სწორი?

ა) კოლექციების იერარქიაში Collection ინტერფეის გააჩნია ოთხი შვილი ინტერფეისი: List, Queue, Set და Map

ბ) Set ინტერფეისი წარმოადგენს გროვას, რომელიც კრძალავს ერთნაირი ელემენტების შენახვას. HashSet იმლემენტირებულია Hash Table-ების დახმარებით, ხოლო TreeSet ხეებით

გ) Map არის key-value სტრუქტურა რომლის key ყოველთვის არის String, ხოლო value შესაძლებელია იყოს ნებისმიერი კლასი

დ) List-ს ორი იმპლემენტაცია LinkedList და ArrayList არის მსგავსი სტრუქტურები, მიუხედავად ალგორითმებში სხვაობისა ისინი არ განსხვავდებიან მეთოდების დროის შეფასებით

9. ჩამოთვლილთაგან პრიმიტიული ტიპები რომელშია დალაგებული მეხსიერების/ინტერვალის ზრდის მიხედვით?

ა) long int short byte

ბ) byte short int long

გ) byte int short long

დ) short byte int long

10. ჩამოთვლილთაგან რომელი მოსაზრებაა მცდარი?

ა) Enum არის სპეციალური კლასი, რომელიც წარმოადგენს რაიმე კონსტანტების ჯგუფის ჩამოთვლას. მას გააჩნია მეთოდი values, რომელიც აბრუნებს ყველა ჩამოთვლილ კონსტანტას

ბ) პრიმიტიული ტიპების Wrapper კლასებს გააჩნიათ სტრიქონში გადაყვანის და სტრიქონიდან მათი მიღების მეთოდები, მაგალითად Integer-ისთვის ეს მეთოდებია toString და parseInt

გ) Static Import-ით შესაძლებელია დაიმპორტებული კლასის სტატიკური ატრიბუტები და მეთოდები გამოვიძახოთ პირდაპირ, კლასის მითითების გარეშე

დ) Generic მეთოდში აღწერილი ჯენერიკული კლასი შეგვიძლია გამოვიყენოთ მხოლოდ გადასაცემ პარამეტრებში და მეთოდის ტანში, მისი დასაბრუნებელ მნიშვნელობად გამოყენება არ შეიძლება