

## ПРОГРАМОВІ ВИМОГИ ДЛЯ ПРОХОДЖЕННЯ ПІДСУМКОВОЇ АТЕСТАЦІЇ

1. Основи Java Collections Framework. Інтерфейс Map та його реалізації.
2. Що таке супроводження програмного забезпечення.
3. Реалізація MVC (написати програмний код).
4. Основи Java Collections Framework. Інтерфейс Set та його реалізації.
5. Що таке аутсорсинг.
6. Реалізація абстрагування(написати програмний код).
7. Основи Java Collections Framework. Інтерфейс List та його реалізації.
8. Що таке Scrum.
9. Реалізація наслідування(написати програмний код).
10. Основи Java Collections Framework. Інтерфейси Iterable та Collection.
11. Що таке життєвий цикл програмного забезпечення?
12. Поліморфізм. Реалізація перезавантаження методів та наслідування (написати програмний код).
13. Синтаксис мови Java. Одновимірні і багатовимірні масиви.
14. Що таке верифікація та валідація.
15. Реалізація перевизначення методів батьківського класу (написати програмний код).
16. Синтаксис мови Java. Лексична структура: розгалуження і цикли.
17. Що таке тестування програмного забезпечення.
18. Реалізація інтерфейсів (написання програмного коду).
19. Синтаксис мови Java. Лексична структура: ідентифікатори та типи даних.
20. Що таке Git?
21. Реалізація принципу SOLID (одного з принципу на вибір реалізувати програмний код).
22. Шаблон об'єктно-орієнтованого проектування MVC – Model-View-Controller.
23. Що таке патерни програмування?
24. Реалізація патернів (написання програмного коду одного з патерну на вибір).
25. Поведінкові шаблони об'єктно-орієнтованого проектування.
26. Що таке SOLID?
27. Реалізація інкапсуляції ( написання програмного коду).
28. Структурні шаблони об'єктно-орієнтованого проектування.
29. Якою командою можна дізнатись про мережеві налаштування вузла?
30. Написати рівні моделі OSI та описати їх значення.
31. Твірні шаблони об'єктно-орієнтованого проектування; проблема, яку вирішує, мотивація щодо вирішення, структура класів, особливості реалізації, зв'язки з іншими шаблонами, переваги та недоліки.
32. Що таке інтеграція та тестування в конструюванні програмного забезпечення?
33. Реалізувати 3 потоки та вивести їхні назви.
34. Принципи об'єктно-орієнтованого програмування SOLID.
35. Що таке архітектура ПЗ в конструюванні програмного забезпеченні?
36. Написати, що роблять команди: pwd, cd /, ls, mkdir, mkdir каталог/каталог1.
37. Уніфікована мова моделювання UML. Діаграма діяльності (activity diagram).
38. Що таке GRUD?

39. Реалізувати зчитування файлу test.txt з системи за допомогою Java, Python 3, C#.
40. Уніфікована мова моделювання UML. Діаграми варіантів використання UML (Usecase diagram).
41. Що таке REST?
42. Написати, що роблять команди: ls -R, find, cp, rm, mv.
43. Уніфікована мова моделювання UML. Діаграми класів UML.
44. Що таке MVC?
45. Написати, що роблять команди: find -name "о\*", cp -v, rm \*!ot, rm dir, pwd.
46. Обробка виняткових ситуацій в об'єктно-орієнтованому програмуванні. Використання ключових слів try, catch, throw.
47. Що таке програмна документація?
48. Написати, що роблять команди: find /etc -name "о\*", cp -i, rm -v, rm -Rf, cd.
49. Методи класу. Перевантаження (overloading) і перевизначення (overriding) методів.
50. Поняття процесу та потоку.
51. Написати, що роблять команди: find /etc -name "о\*" 2>/dev/null, cp -R, rm -i, touch, ls -R.
52. Види операцій над об'єктом. Методи класу. Конструктор і деструктор.
53. Які функції виконує маршрутизатор.
54. Написати, що роблять команди: cat, less, zip, history, lsof.
55. Множинне успадкування (наслідування) в об'єктно-орієнтованому програмуванні.
56. Які типи кабелів використовуються в локальних мережах?
57. Написати, що роблять команди: cat file1.txt file2.txt, head, zip -(0-9), lsof -u.
58. Доступ до членів класів в об'єктно-орієнтованому програмуванні. Специфікатори доступу public, private, protected при наслідуванні.
59. Файли, каталоги, реалізація файлових систем.
60. Написати, що роблять команди: cat -n file1.txt file2.txt, unzip, alias, lsof -c [програма], dpkg -r [пакет].
61. Абстрактний клас і інтерфейс. Призначення. Спільні риси, відмінності.
62. Що таке криптографія.
63. Написати, що роблять команди: head file1.txt file2.txt, unzip -v, unalias, free, dpkg -I.
64. Наслідування в об'єктно-орієнтованому програмуванні.
65. Що таке Agile?
66. Написати, що роблять команди: cat file1.txt file2.txt > file3.txt, zip -(p або e), !nnn, lsof [файл], apt -get install [пакет].
67. Методологія ООП. Відношення між класами.
68. Що таке Kanban?
69. Написати, що роблять команди: head -t file1.txt, unzip -l, ps aux, df, yum update.
70. Основні характеристики об'єкта: стан, поведінка, індивідуальність.
71. Основні характеристики комп'ютерних мереж.
72. Написати, що роблять команди: tail, unzip -t, ps U, yum install [пакет], pwd.
73. Методологія ООП. Поняття об'єкта і класу.
74. Віртуалізація багатопроцесорних систем.

75. Написати, що роблять команди: `head -c 100 file1.txt`, `ps axjf`, `du`, `yum remove` [пакет], `mkdir`.
76. Складові частини об'єктного підходу. Поняття ієрархії.
77. Топологія мереж. Види топологій мереж.
78. Написати, що роблять команди: `tail -t file1.txt`, `gzip -c`, `top`, `rpm -ihv` [пакет], `apt -get remove` [пакет].
79. Складові частини об'єктного підходу. Поняття модульності.
80. Аутентифікація. Інсайдерські атаки.
81. Написати, що роблять команди: `tail file1.txt file2.txt`, `gzip`, `kill`, `rpm -e` [пакет], `apt -get upgrade`.
82. Складові частини об'єктного підходу. Поняття інкапсуляції.
83. Файлова система Linux. Безпека в операційній системі Linux.

84. Розгляньте уривок коду:

```
int x;  
int a = 5;  
int b = 8;  
x = ++a + b++;
```

Яким буде значення змінної `x` після його виконання?

85. Складові частини об'єктного підходу. Поняття абстрагування.
86. Що таке розподілені обчислення?
87. Розгляньте уривок коду:

```
public class Foozit {  
    public static void main(String[] args)  
        Integer x = 0;  
        Integer y = 0;  
        for(Short z = 0; z < 5; z++)  
            if((++x > 2) || (++y > 2))  
                x++;  
        System.out.println(x + " " + y);  
    }  
}
```

Яким буде результат?

88. Синтаксис мови Java. Лексична структура: ідентифікатори та типи даних.
89. Які переваги паралельної обробки даних?
90. Розгляньте уривок коду:

```
for (int i = 0; i < 2; i++) {  
    for (int j = 0; j < 3; j++) {  
        if (i == j) {  
            continue;  
        }  
        System.out.println("i = " + i + " j = " + j);  
    }  
}
```

Що виведе дана програма?