

Въпроси:

## Увод в програмирането

1. Как протича създаването на една програма (през какви етапи минава)?
2. Какви видове типове данни има?
3. Защо са ни различните типове?
4. Колко вида побитови операции познавате?
5. Какво представлява функцията?
6. Каква е разликата между формален и фактически параметър?
7. Какво представлява прототипът на функцията?
8. Прехвърлянето на границите на масива при индексирание - възможно ли е?
9. Какво представлява масивът?
10. Какви може да са различните реализации на масивът?
11. Кога се определя размерът му?
12. Можем ли да променяме размерът му?
13. Какво представлява двумерният масив?
14. Какво е указател?
15. Какво е референция?
16. Каква е разликата между указател и референция?
17. Какви видове указатели има?
18. За какво се ползват?
19. Видове памет?
20. За какво се използват?
21. Какъв е процесът на заделяне на памет?
22. Защо използваме динамично заделена памет?
23. Защо е важно да си трием паметта винаги, когато вече не ни трябва?
24. Какво представлява приоритет на операцията?
25. Какво е `while(0)`, а `while(1)`?
26. Какво представлява функцията `main`?
27. Можем ли да имаме програма без `main`?
28. Може ли да има повече от една `main` функция?
29. Какво е рекурсия?
30. Какви проблеми могат да възникнат?

## Обектно-ориентирано програмиране

1. Какво е ООП?
2. Защо го ползваме?
3. Кои са основните парадигми на ООП?
4. Какво е структура?
5. Каква е подразбиращата се област на видимост на структурата?
6. За какво се използва?
7. Може ли да има конструктор/деструктор?
8. Какво беше анонимна структура?

9. Какво е клас?
10. Каква е подразбиращата се област на видимост на класа?
11. Как класът ни помага да запазим енкапсулацията на данните?
12. Какво е Г4?
13. Какво ще стане, ако не напишем имплементацията на Г4?
14. Може ли да има повече от един конструктор?
15. Какво означава default-ен конструктор и кога програмата не го използва?
16. Може ли да има повече от един деструктор?
17. Защо се ползва?
18. Какво представлява инициализиращия списък - кога е ЗАДЪЛЖИТЕЛНО да го ползваме?
19. Вярно ли е, че инициализиращият списък на даден конструктор се изпълнява след като завърши изпълнението на тялото на конструктора?
20. В какъв ред се изпълнява инициализиращия списък?
21. Какви са видовете потоци за работа с файлове?
22. Каква е разликата между текстов и бинарен файл?
23. Какво означава наследяване?
24. Колко вида наследяване има?
25. Защо се ползва?
26. Какви проблеми могат да възникнат и как да се справим с тях?
27. Какво представлява абстрактният клас?
28. Може ли да се създава обекти от абстрактни класове?
29. Какво представлява композицията?
30. Ако класът А реализира design pattern-а Singleton, какво значи това?
31. Как се създават/разрушават наследени обекти?
32. Какво представлява предефинирането на оператори?
33. Защо предефинираме оператори?
34. Какво е унарен/бинарен оператор?
35. Какво е итератор?
36. Какво е изключение - какво прави, кога възниква ?
37. Какво хвърля операторът new?
38. Защо/ Кога е необходимо да използваме изключения?

## Структури от данни и програмиране

1. Занимавали ли сте се с бързи алгоритми?  
binary search vs linear search
2. Писали ли сте структура от данни?  
(динамичен масив) - какво беше специфично за него
3. Защо всичко дотук беше важно да го повторим?