

**Програмові вимоги до Державного екзамену
для студентів спеціальності 113 «Прикладна математика»**

Чисельні методи

1. Наближені величини. Поняття похибки. Абсолютна і відносна похибки. Граничні похибки наближеного числа.
2. Значущі цифри числа. Вірні цифри. Зв'язок абсолютної і відносної похибок з кількістю вірних цифр числа.
3. Округлення чисел. Правила округлення. Похибка.
4. Загальні формули обчислення похибок. Похибки елементарних функцій.
5. Похибки арифметичних операцій.
6. Наближене розв'язування алгебраїчних та трансцендентних рівнянь. Методи відокремлення коренів.
7. Метод половинного поділу розв'язування алгебраїчних та трансцендентних рівнянь.
8. Метод хорд наближеного розв'язування алгебраїчних та трансцендентних рівнянь.
9. Метод дотичних (Ньютона) наближеного розв'язування алгебраїчних та трансцендентних рівнянь.
10. Комбінований метод хорд та дотичних.
11. Метод Гауса розв'язування систем лінійних алгебраїчних рівнянь. Схема єдиного ділення. Застосування до обчислення визначників і відшукування обернених матриць.
12. Методи Холецького та квадратних коренів розв'язування систем лінійних алгебраїчних рівнянь.
13. Інтерполяція функцій. Постановка задачі, основні поняття.
14. Інтерполяційна формула Лагранжа. Оцінка похибки.
15. Числове диференціювання функцій.
16. Методи числового інтегрування. Формули прямокутників. Похибка.
17. Методи числового інтегрування. Формула трапецій. Похибка.
18. Числове розв'язування диференціальних рівнянь. Метод послідовних наближень (метод Пікара).
19. Чисельне розв'язування диференціальних рівнянь. Метод Ейлера.

Моделювання виробничих і економічних процесів

1. Задача про використання сировини (оптимальне планування виробництва).
2. Задача про оптимальний розкрій матеріалів.
3. Задача про дієту.
4. Задача про оптимальне використання потужностей.
5. Задача про оптимальні інвестиції.
6. Постановка і умова розв'язності транспортної задачі.
7. Методи збалансування транспортної задачі.
8. Методи знаходження початкового опорного плану транспортної задачі.
9. Метод потенціалів знаходження оптимального плану транспортної задачі. Цикл перерахунку.
10. Задача транспортного типу, що містить обмеження на перевезення за деякими маршрутами.
11. Транспортна задача із фіксованими обсягами перевезень за окремими маршрутами.
12. Задачі з обмеженнями знизу на обсяги перевезень.
13. Лінійна регресійна модель: означення. Поле кореляції.
14. Числові характеристики регресійної залежності: коваріація, коефіцієнт кореляції, коефіцієнт детермінації).
15. Знаходження параметрів лінійної регресійної моделі.

16. Адекватність моделі : означення, дослідження.
17. Прогнозування економічних процесів за економетричними моделями.
18. Нелінійні регресійні моделі та методи їх лінеаризації.
19. Алгоритм динамічного програмування. Задача про розподіл інвестицій.

Алгоритмічні мови та програмування

1. Який з наведених типів даних не відноситься до базових в мові C/C++:
 - int
 - long
 - char
 - void
 - double
2. Функція malloc та оператор new виконують резервування динамічної пам'яті. Котра з наведених відмінностей є справедливою:
 - оператор new підтримується тільки мовою C++, а функція malloc як мовою C так і C++;
 - функцію malloc не можна використовувати для багатомірних масивів, тоді як new використовується для будь-яких складних типів;
 - оператор new використовується для роботи з динамічними масивами і вказівниками, malloc використовується як для статичних, так і до динамічних масивів;
 - функція malloc використовується тільки для числових типів, а оператор new – для будь-яких;
 - функція malloc є застаріла і не підтримується новими стандартами
3. Вкажіть правильний порядок формування виконавчого модуля для мови C/C++:
 - вихідний текст, обробка препроцесором, синтаксичний аналіз, компіляція (об'єктний код), лінкування (зв'язування);
 - вихідний текст, лінкування (зв'язування), синтаксичний аналіз, компіляція (об'єктний код), обробка препроцесором;
 - вихідний текст, синтаксичний аналіз, лінкування (зв'язування), компіляція (об'єктний код), обробка препроцесором;
 - вихідний текст, обробка препроцесором, синтаксичний аналіз, формування .exe файлу, вихідний текст, синтаксичний аналіз, лінкування (зв'язування), обробка препроцесором, компіляція (об'єктний код).
4. Котрі з наведених виразів не належать до директив препроцесора:
 - extern
 - include
 - define
 - ifndef
 - endif
5. Котре з наведених ключових слів є модифікатором знаку:
 - unsigned
 - long
 - short
 - static
 - void
6. Для оголошення функції `\nint main(int argc, char* argv[])\n`, яке значення отримує аргумент argc при запуску програми:

- аргумент не несе ніяких даних;
 - значення кількості екземплярів програми запущених на виконання;
 - кількість аргументів командного рядка, які передані програмі в якості параметрів;
 - розмір динамічної пам'яті доступної для використання програмою;
 - вказівник на строку символів, які є параметрами командного рядка.
7. Модифікатор типу пам'яті `\n static` `\n` призначений:
- для створення статичного масиву;
 - для оголошення змінних які створюються на початку роботи програми і знищуються по її завершенню;
 - для створення локальних змінних в рекурсивних функціях;
 - дає завдання компілятору розмістити змінну в області пам'яті доступній тільки для читання;
 - даний модифікатор не є самодостатній і може використовуватися тільки з модифікатором `long`.
8. Який тип пам'яті мають змінні за замовчуванням:
- `register`
 - `static`
 - `auto`
 - `public`
 - `signed`
 - `bool`
9. Що є справедливим для наведеного фрагменту програми:
- ```
\n char c='1';\n int i;\n c++;\n i=(int)c;
```
- явне приведення типу `char` до типу `int` є надлишковим для C/C++
  - вираз `\nc++;` є неправильним, оскільки `c` має тип `char`
  - символу `c` не можна виконувати присвоєння при ініціалізації
  - типу `int` неможливо присвоїти значення типу `char`
  - фрагмент коректний і не потребує корегувань
10. Як охарактеризувати фрагмент коду: `\n char *s1;\n s1=new char [50];\n strcpy(s1,"Hello");\n s1=new char[10];`
- фрагмент некоректний за синтаксисом мови C/C++;
  - фрагмент повністю коректний;
  - оператор `new` не підтримується для змінних типу `char`
  - фрагмент коректний за синтаксисом, але під час виконання може виникнути помилка доступу до пам'яті при копіюванні строки;
  - синтаксис коректний, але має місце втрата адреси на виділену область пам'яті.
11. Що таке блок коду в розумінні мови C/C++
- це ділянка програми, яка повністю охоплює тіло функції, але без визначення параметрів;
  - це ділянка програми, яка повністю охоплює тіло функції, включаючи визначення параметрів;
  - вся частина програми, яка визначена у функції `main()`
  - ділянка програми обмежена операторами `{` та `}`
  - частина програми визначена у одному файлі.
12. Як можна охарактеризувати наведений фрагмент програми: `\n struct MyStruct {\n\t int i;\n\tchar c;\n\t float f [40];\n} m_s;`
- це правильне оголошення складного типу – структури;
  - фрагмент неправильний, так як структура не може містити статичний масив;

- фрагмент не правильний, так як не можна виконувати оголошення структури і її визначення;
- фрагмент неправильний, так як після закінчення оголошення структури необхідно включити оператор ; ( тобто };)
- фрагмент неправильний, так як C++ замість структур використовує класи.

13. Яку дію виконує в мові C/C++ оператор ^

- оператор не підтримується мовою C/C++
- це оператор піднесення до степеня;
- це оператор "порозрядне логічне АБО";
- це оператор порозрядного зсуву вправо;
- це оператор доступу до елемента структури.

14. Яку дію має оператор >> в базовому синтаксисі мови C

- зсув значення змінної вправо на певну кількість розрядів
- виконання операції введення змінної з консолі за допомогою потоку cin
- виконання операції виводу значення змінної чи константи cout
- це оператор перенапрявлення даних у певний ініціалізований потік
- виконання нарощення змінної на певне значення

15. Який з вказаних операторів, виконує операцію "логічного І" на мові C/C++

- &&
- AND
- And
- And
- &&&&

16. Яку область видимості за замовчуванням мають функції на мові C/C++

- область видимості у межах блоку в якому вони об'явлені;
- файлову область видимості;
- статичну область видимості;
- всі функції, незалежно від їх об'явлення мають область видимості в межах виклику з функції main
- всі функції, незалежно від їх об'явлення визначаються препроцесором.

17. Що таке область видимості змінної:

- ділянка програми, в межах якої змінна чи константа є доступна;
- область пам'яті, в якій зберігається змінна і є доступною для модифікації;
- файл, в якому визначена змінна;
- блок коду, в якому визначена змінна чи константа.

18. Вкажіть правильну послідовність формування коментарю:

- // Це текст коментаря\n /\*\n Це також \n коментар \n \*/
- /\* Це текст коментаря\n /\*\n Це також \n коментар \n /\*
- /\* Це текст коментаря \*/ \n // Це також \n коментар \n //
- { Це текст коментаря } \n { \n Це текст коментаря \n }

19. Що таке структура мови програмування:

- припустимі способи групування лексем у вирази, оператори та інші значення одиниці мови;
- набір символів алфавіту мови та операторів, за допомогою яких формується програма;
- набір констант, змінних та операторів з яких складається програма;
- поділ вихідного тексту програми на лексеми та розділювачі.

20. Що таке операція розіменування адреси:

- це доступ до змінної по її адресі;
- це операція побітного заперечення;
- це операція доступу до елемента структури;
- це операція доступу до елемента об'єднання;
- операція взяття адреси змінної.

21. Яке з визначених об'явлень створить змінну, яка містить символ:

- int c;
- char c;
- char \*c;
- float c;
- bool c;
- character c;

22. Котре з оголошень правильно використовує модифікатор розміру short:

- short int i;
- short char c;
- short char \*s;
- short float f;
- short double d;

23. Котре з оголошень правильно використовує модифікатор розміру long:

- long char c;
- long char \*s;
- long float f;
- long double d;
- long void \*v;

24. Яке призначення оператора sizeof:

- це модифікатор знаку, який визначає беззнакову змінну;
- визначення розміру в байтах змінної, структури, об'єднання;
- це ключове слово базового типу змінної;
- визначення розміру в байтах для масивів;
- визначає розмір доступної оперативної пам'яті.

25. Яке значення ключового слова unsigned:

- це модифікатор знаку який визначає беззнакову змінну;
- це модифікатор розміру, який зменшує розмір змінної до двох байт;
- це ключове слово базового типу змінної;
- це модифікатор файлової області видимості;
- це ключове слово застаріле і не підтримується новими компіляторами.

26. Яке значення ключового слова extern:

- повідомлення компілятору, що пам'ять для змінної виділена в іншому місці програми;
- визначення класу пам'яті статичної змінної;
- модифікатор типу даних, для змінних типу PASCAL;
- явне приведення типу даних масиву, до одиночної змінної
- вказує на використання змінної, яка зберігається у файлі іншої мови програмування.

27. Яка відмінність між статичними і динамічними масивами:

- статичний масив використовується тільки для збереження цифрових значень;
- динамічний масив використовується тільки для збереження цифрових значень;

- розмір динамічного масиву можна змінювати в процесі роботи програми, статичного ні;
- розмір статичного масиву можна змінювати в процесі роботи програми, динамічного ні;
- вказати розмір статичного масиву можна за допомогою змінної, динамічного тільки константи.

28. Яка відмінність між статичним масивом і вказівником:

- значення вказівника є змінною, статичний масив є вказівником константою;
- для статичного масиву не можна проводити ініціалізацію при оголошенні, для динамічного можна;
- статичний масив може містити значення тільки базових типів, вказівник - базових та похідних;
- для доступу до елементів статичного масиву можна використовувати операцію індексування [], для доступу по вказівнику – тільки операцію розіменування \*()

29. Вкажіть правильний синтаксис оператора for мови C/C++

- for (початкове\_присвоєння; умова\_виконання; оператор\_нарощення) {...}
- for (початкове\_присвоєння; умова\_виконання; step N) {...}
- for початкове\_присвоєння to максимальне\_значення do begin ... end;
- for (початкове\_присвоєння; умова\_виконання; оператор\_нарощення); {...}
- for початкове\_присвоєння; умова\_виконання; оператор\_нарощення begin ... end;

30. Вкажіть правильний синтаксис оператора if мови C/C++:

- if (умова) then {...}\n else {...}
- if умова then begin ... end; \n else begin ... end;
- if умова then begin ... end; \n else begin ... end;
- if (умова) {...}\n else {...}
- if (умова) {...}\n {...}

31. Яка відмінність між оператором do {...} while(умова); і оператором while (умова) {...}

- оператор do – while не підтримується мовою C/C++
- блок коду do – while виконується мінімум один раз, а блок коду while може не виконатися жодного разу;
- оператори рівноцінні;
- оператор do – while використовується для організації циклічних алгоритмів, а оператор while – для розгалужень;

32. Яку дію виконує оператора break:

- передача управління за межі блоку коду в операторах switch, for, while, do;
- повернення керування у точку виклику функції;
- передача управління на наступний етап циклу в операторах for, while, do;
- зупинка виконання програми до натиснення користувачем певної клавіші;
- передача управління на наступну умову в операторі case;

33. Що може бути використано у якості мітки M в операторі \n switch (...) \n { case M: ...;}

- будь яке цифрове значення;
- тільки константи які однозначно приводяться до типу int (char, int, long int)
- змінні та константи цілого типу
- константи базових типів
- змінні та константи типу int та char

34. Яке призначення файлу stdio.h

- автоматично генерується середовищем розробки програм для об'явлення користувачьких змінних та функцій;
- містить прототипи функцій стандартної бібліотеки вводу-виводу;
- призначений для підключення бібліотеки потокового вводу з клавіатури та виводу даних на консоль;
- містить прототипи функцій для роботи з файловою системою.

35. Що таке рекурсивна функція:

- функція, яка не має параметрів, та не повертає ніякого значення;
- функція, яка викликає сама себе;
- це функція main, яка немає аргументів;
- функція, яка призначена для обробка рекурсивних послідовностей.

36. Який файлу заголовку необхідно включити в програму для використання бібліотеки математичних функцій:

- `stdio.h`
- `math.h`
- `string.h`
- `conio.h`
- `iostream.h`

37. Який з наведених прикладів правильно демонструє використання функції `scanf`

- `int i,j;\n scanf ("%d ",&&i, "%d",&&i);`
- `int i,j;\n scanf ("%d %d ",&&i,&&j);`
- `int i,j;\n scanf ("%d",i);\n scanf ("%d",j);\n`
- `int i,j;\n scanf ("%f %f ",&&i,&&j);`

38. Який з специфікаторів виводу є оптимальний для виводу на консоль змінної типу `int` за допомогою функції `printf`

- `%d`
- `%c`
- `%f`
- `%s`
- `%lf`

39. Який з специфікаторів виводу є оптимальний для виводу на консоль строки символів

- `%d`
- `%c`
- `%f`
- `%s`
- `%lf`

40. Яка стандартна функція використовується для пошуку входження підстроки в строку:

- `strstr`
- `strsearch`
- `strcpy`
- `strfind`
- `strcat`

41. Яка стандартна функція використовується для копіювання однієї строки в іншу:

- `strstr`
- `strcpy`
- `strcat`
- `strcopy`

- strmerge

42. Яке значення прийме змінна k в наведеному фрагменті:\n \n int i,j,k;\n i=10; j=20;\n k=j>10 ? j/i : i/j; \n

- 10
- 0.5
- 20
- Фрагмент не коректний за синтаксисом

43. Яке значення прийме змінна k в наведеному фрагменті:\n int i, j, k;\n i=2; j=2; k=2;\n i+=3; j\*=2; \n k\*=i\*j;

- 40
- фрагмент не коректний за синтаксисом
- 20
- 2

44. Яке значення буде виведено на консоль:\n char sss[4]="abc"; \n printf( "%c", \*(sss+1) );

- фрагмент не коректний за синтаксисом
- символ a
- символ b
- символ c
- нульовий символ.

45. Яке значення буде виведено на консоль:\n char sss[4]="abc"; \n printf( "%c", sss[3] );

- символ a
- символ b
- символ c
- нульовий символ;
- фрагмент не коректний за синтаксисом

46. Яке призначення оператора return

- повернення управління з функції в точку її виклику;
- вихід за межі блоку коду в операторах for, while
- повернення управління на початок циклу;
- передача аргументів функції, що буде викликана;
- завершення оператора goto

47. Яка із команд умовного переходу мови C записана правильно:

- if (a=0) then b=9;
- if (a=0) then b=9 else b=-9;
- if a==0 then b==9;
- if (a==0) b=9; else b=-9;
- if (a==0) b=9 else b=-9;

48. Де написано правильну організацію друку 10 літер 'A', ... , 'J'?

- for (ch='A'; ch='J') printf("%c ",ch);
- for (ch='A'; ch<='J'; ch++) printf("%c ",ch);
- for (ch='A'; ch<='J'; ch++) printf("%ch ",c);
- for (ch=='A'; ch='J'; ch=ch+'A') printf("%c ",ch);
- for (ch='A'..'J') printf("%c ",ch);

49. Вкажіть правильну організацію циклу з післяумовою:

- repeat { ... } until (умова);
- do { ... } while (умова);

- repeat { ... } while (умова);
- while (умова) do { ... };
- do { ... } until (умова);

50. Які слова використовуються в конструюванні команди вибору:

- if, then, else
- switch, case, default
- switch, mode, delete
- if, switch, def
- if, else, define

51. Вкажіть правильну організацію циклу з передумовою:

- if (умова) do { ... };
- do { ... } while (умова);
- repeat { ... } while (умова);
- while (умова) { ... };
- do { ... } until (умова);

52. Який оператор призначений для знаходження остачі?

- mod
- ??
- &&
- %
- ||

53. Яким буде результат команди: `int a \n if (3!=5) a=4/5; else a=5/4;`

- 0.8
- 1.25
- 5
- 4

54. Яка функція призначена для виведення на друк?

- print
- scan
- gets
- printf
- scanf

55. Яка функція на мові C не призначена для вводу значення з клавіатури?

- scanf
- keypressed
- gets
- getch

56. Яка функція (описана в файлі "string.h") призначена для "склеювання" рядків?

- то не функція, а оператор +
- то не функція, а оператор ADD
- StringConcatenate()
- strcpy()
- strcat()

57. Яка функція (описана в файлі "string.h") вимірює довжину рядка?

- length()
- strlen()

- strlen()
- len()
- strsize()
- sizeof()

58. Що виконає цикл `for(int i=4; i<3;i++) printf("A")`?

- нічого, адже умова не виконається навіть при старті
- надрукує одну літеру A
- надрукує AAA
- надрукує AAAA
- це нескінченний цикл, бо умова неправильна, отже "зависне"

59. Яким буде значення змінної A після виконання таких команд: `int A=13; A/=4;`

- 13
- 7.5
- 3.25
- 134

60. Виберіть конструкцію, яка не організовує нескінченний цикл:

- `for (i=1;1;i=1) i=1;`
- `for (i=0;0;i=0) i=0;`
- `while (1) i=1;`
- `while (2) i=2;`
- `do i=2; while(2);`

61. Яка з директив препроцесора призначена для означення констант, макросів?

- `#include`
- `#define`
- `#pragma`
- `#undef`
- `#macro`

62. Якої директиви препроцесора не існує?

- `#include`
- `#pragma`
- `#ifdef`
- `#define`
- `#select`

63. Вкажіть правильне оголошення одновимірного масиву із 10 елементів.

- `int a = array[10];`
- `int a = new[10];`
- `int a[10];`
- `int a[0][1][2][3][4][5][6][7][8][9];`
- `int a[0..9];`

64. Вкажіть правильне створення одновимірного динамічного масиву довжиною 100 елементів:

- `int *A = new int[100];`
- `int A = new[100];`
- `int A = new int [0..99];`
- `int A[100] = new[100];`
- `int *A = new array of [0..99];`

65. Розмір якого типу (для Win32, компілятор MS Visual C++, Borland C++ Builder) не дорівнює 4 байтам?
- int
  - unsigned int
  - long
  - unsigned long
  - double
66. Вкажіть, де наведено неправильне звернення до останнього елемента масиву S[100].
- a = S[100];
  - a = \*(S+99);
  - a = S[99];
  - a = S[100-1];
  - a = \*((S+100)-1);
67. На який елемент масиву вказуватиме вказівник p після виконання команд double A[100]; double \* p; p=A; p+=8;
- A[0]
  - A[1]
  - A[2]
  - A[7]
  - A[8]
68. Розмір якого типу завжди дорівнює 2 байти?
- char
  - BYTE
  - int
  - unsigned short
  - unsigned long
69. Де неправильно задекларовано (оголошено) функцію користувача:
- int Function1(int a, int b, int c);
  - Function2(int \*a, int \*b, int \*c);
  - void Function3 (int, int, int);
  - Function4(void a, void b, void c);
  - int Function5(int a, int \*b, int \*\*c);
70. Яке з ключових слів не належить мові C / C++ ?
- if
  - do
  - else
  - then
  - while
71. Яке з ключових слів не належить мові C / C++ ?
- new
  - void
  - unsigned
  - extern
  - intern
72. Яке оголошення є некоректним?
- unsigned int a = 123;
  - int \*a = NULL;

- `double a[5] = {1.2, 2.4, 3.6, 6.8, 9.9};`
- `char *a = 's';`
- `unsigned char A = 'A';`

73. Нехай є два вказівники на один і той же тип. Чи можна порівнювати ці вказівники?

- ні;
- так (на рівність-нерівність, більше-менше);
- так, тільки на рівність-нерівність (на більше-менше не можна порівнювати);
- якщо вказівник вказує на число, то можна, а якщо не на число – то ні;
- якщо один з них рівний NULL, то можна, а інакше – ні.

74. Що таке вказівник?

- це змінна, яка містить адресу пам'яті певного об'єкту;
- це група операторів призначені для роботи з пам'яттю;
- це оператор послідовного обрахунку;
- це змінна, яка вказує на саму себе;
- це адресний оператор.

75. Яка із вказаних операцій не підтримується мовою C?

- арифметичні;
- побітові;
- присвоєння;
- пам'яті;
- змішаного типу.

76. Що робить арифметична операція ++ «інкремент»?

- збільшує значення вказівника на один байт;
- збільшує значення числової змінної на 1;
- зменшує значення числової змінної на 1;
- виконує нарощення значення числової змінної на вказане значення;
- підтримується тільки C++

77. Для чого призначений оператор switch?

- для переходу залежності від значення певного константного виразу
- для виведення керування за межі блока коду в оператор while.
- для передачі управління з функції в точку її виклику
- для розіменування адреси.
- це оператор виконання розгалуження на основі виконання операції порівняння двох чи більше операндів об'єднаних логічними умовами

78. Де правильно написана функція відкриття файлу:

- `fileopen`
- `fopen`
- `fclose`
- `filew`
- `fileclose`

79. Знайдіть не коректний вираз:

- `if(p==c) cout<<c;`
- `for(i=0;i<10;i++) cout<<i;`
- `for(i=0, j=100;i<j;i++, j--) cout<<i;`
- `if(p=c) cout<<c; \n else cout<<p;`

80. Де правильно використана функція піднесення до степеня змінної x:

- $x^2$
- `sqr(x)`
- `sqrt(x)`
- `pow(x,2)`

81. Який результат буде виведений на екран після виконання наступних дій `int x;`  
`x=1.2;`  
`cout<<x;`?

- 1.2
- 0.2
- Компілятор повідомить про помилку невідповідності типів.

82. Який з принципів не відноситься до базових принципів об'єктно-орієнтованого програмування:

- функціональної декомпозиції;
- поліморфізму;
- інкапсуляції;
- спадкування.

83. Інкапсуляція означає:

- реалізація механізму, єдиний інтерфейс, багато методів;
- організація механізму, отримання одним об'єктом властивостей іншого об'єкту;
- розділення програми на ряд функцій, які виконують незалежну дію;
- створення платформи-незалежного інтерфейсу користувача;
- забезпечення механізму об'єднання даних та коду їхньої обробки, для захисту того та іншого від неправильного застосування.

84. Принцип поліморфізму означає:

- реалізація механізму, єдиний інтерфейс, багато методів;
- організація механізму, отримання одним об'єктом властивостей іншого об'єкту;
- розділення програми на ряд функцій, які виконують незалежну дію;
- створення платформи-незалежного інтерфейсу користувача;
- забезпечення механізму захисту внутрішніх даних та приховування способу виконання певних дій.

85. Принцип спадкування означає:

- реалізація механізму, єдиний інтерфейс, багато методів;
- організація механізму, отримання одним об'єктом властивостей іншого об'єкту;
- розділення програми на ряд функцій, які виконують незалежну дію;
- створення платформи-незалежного інтерфейсу користувача;
- забезпечення механізму захисту внутрішніх даних та приховування способу виконання певних дій.

86. Під об'єктом розуміють:

- екземпляр класу, який об'єднує дані та функції, які маніпулюють ними;
- синтаксичну конструкцію мови програмування, яка однозначно ідентифікується компілятором;
- функцію, яка виконує закінчену дію;
- об'єктний файл;
- модель складної системи користувачських даних.

87. Під класом розуміють:

- опис змінних та методів їхньої обробки, які моделюють певні об'єкти;
- послідовність виклику функцій у головній програмі;
- екземпляр виконавчого модуля, який запущений на виконання;

- ключове слово class, яке однозначно ідентифікується компілятором C/C++;
- будь-який тип користувацьких даних, який не належить до базових;

88. Методом класу(функцією членом) називають:

- функцію, яка є членом класу і має доступ до його даних;
- функцію, яка має доступ до внутрішніх даних класу;
- перевантажений оператор класу;
- функцію, яка в якості параметрів використовує об'єкти класу;
- функція, яка є дружньою до двох чи більше класів;

89. До специфічних методів класу належать:

- конструктори та деструктор;
- оператор присвоєння та конструктор копії;
- всі перевантажені оператори;
- будь-який метод класу, який виконує специфічні дії;
- функції, які мають доступ до закритих членів класу;

90. Конструктором називають:

- метод класу, який має таке ж саме ім'я, як і клас і створює об'єкт класу;
- будь-який об'єкт класу, який не має параметрів;
- метод класу, який повертає в якості значення об'єкт класу;
- метод класу, який знищує об'єкт класу;
- метод класу, який приймає в якості значення об'єкт класу;

91. Деструктором називають:

- метод класу, який має таке ж саме ім'я, як і клас і створює об'єкт класу;
- будь-який об'єкт класу, який не має параметрів;
- метод класу, який повертає в якості значення об'єкт класу;
- метод класу, який знищує об'єкт класу;
- метод класу, який приймає в якості значення об'єкт класу.

## **Бази даних та інформаційні системи**

1. Поняття інформаційної системи. Класифікація інформаційних систем.
2. Поняття бази даних. Основні функції систем керування базами даних.
3. Класифікація систем керування базами даних. Автономні інформаційні системи.
4. Реляційна модель даних. Поняття моделі даних. Типи даних.
5. Об'єкти реляційних баз даних. Сутність зв'язування таблиць. Види зв'язків між таблицями.
6. Проектування інформаційних систем. Рівні подання бази даних.
7. Система керування базами даних MS Access.
8. Загальний порядок створення таблиць. Типи даних і властивості полів. Створення структури таблиці. Модифікація структури таблиць.
9. Ключові поля. Індокси. Зв'язування таблиць.
10. Введення дання у таблицю. Пошук і заміна даних у таблицях. Сортування й фільтрація записів.
11. Загальні відомості про запити. Оператори і вирази.
12. Запити на основі однієї таблиці. Редагування запитів.
13. Запити для багатьох таблиць.
14. Запити з полями, що обчислюються.
15. Запити з функціями. Запити зі змінними критеріями.
16. Перехресні запити. Запити на створення таблиць, оновлення та видалення даних.
17. Засоби створення форм. Елементи керування та їх властивості.

18. Створення форм за допомогою конструктора форм. Модифікація форм.
19. Автоматичне створення звіту. Створення звіту в режимі конструктора.
20. Модифікація звітів.
21. Імпорт та експорт даних.
22. Адміністрування бази даних.
23. Загальні відомості про мову SQL. Типи даних. Оператори та вирази.
24. Основні можливості мови SQL. Створення найпростіших баз даних. Створення й модифікація таблиць.
25. Маніпулювання даними в таблицях. Найпростіші запити.
26. Запити з умовою. Використання функцій у запитах.
27. Запити з групуванням рядків. Запити з упорядкуванням рядків.