Основи програмування

Лабораторна робота №3

Тема: Управління потоком виконання

CEM KOYLUOGLU -IP-98

Контрольні питання:

1. Що таке структурне програмування?
2. Structured programming is **a paradigm that aims to make programs easier to comprehend from a reader's point of view**. It does this by linearising the flow of control through a program. In structured programming, execution follows the writing order of the code.
3. Назвіть основні конструкції структурного програмування.
4. Structured programming is a program written with only the structured programming constructions: **(1) sequence, (2) repetition, and (3) selection**. Sequence. Lines or blocks of code are written and executed in sequential order. Repetition.

3. Назвіть основні принципи структурного програмування.

- Keep It Simple, Stupid (KISS) ...
- Write DRY Code. ...
- Open/Closed. ...
- Composition Over Inheritance. ...
- Single Responsibility. ...
- Separation of Concerns. ...
- You Aren't Going to Need It (YAGNI) ...
- Document Your Code.

5. Поясніть, чому в більшості мов програмування не радять використовувати оператор GOTO, а в мові Java його взагалі немає?
6. Use of goto statement is highly discouraged in any programming language **because it makes difficult to trace the control flow of a program**, making the program hard to understand and hard to modify. ... Any program that uses a goto can be rewritten to avoid them.

5. Що таке спагеті-код?

To me, a more modern example of spaghetti code is **when you have 20 dlls and every DLL references each other in one way or another**. Your dependency graph looks like a huge blob, and your code hops all over the place with no real order. Everything is inter-dependent.

7. Що таке проектування зверху-вниз? В чому його переваги?
8. It **helps bring clarity in the difference between two systems or subsystems**. As the modules are broken into smaller pieces, there is no scope for mistakes, as it helps ensure that the highest quality is achieved after the problem is solved.

7. Що таке область видимості ідентифікатора?

C++ Reference Material | Scope of an Identifier. Any identifier used in a C++ program (such as the name of a variable or object, the name of a type or class, or the name of a named constant) has a scope, i.e., a **region of the program in which that name can be used**.

9. Що таке блок? Як блоки впливають на область видимості?
10. In computer programming, a block or code block or block of code is **a lexical structure of source code which is grouped together**. ... A programming language that permits

the creation of blocks, including blocks nested within other blocks, is called a block-structured programming language

11. Visibility **Controls determine whether a block should be visible to users on the frontend of your website**. ... Only show blocks to logged-out users. Only show blocks to users with specific roles (Administrator, Subscriber, Customer, Member etc.) Only show blocks to specific users. Great for personalization.

9. Наведіть приклади, у яких випадках краще використовувати наступні оператори та конструкції:
- if
- if-else
- ланцюги if-else if -else if...
- switch
- тернарна умовна операція ?:

```
const result = (merit) => {
  return merit > 100 && merit < 150 ? 'Pass'
    : merit > 150 && merit < 200 ? 'Second Class'
    : 'First Class';
}
console.log(result(201));
```

10. Наведіть приклади, у яких випадках краще використовувати наступні оператори та конструкції:
- while
- do-while
- for

```
class whileLoopDemo {

    public static void main(String args[])
    {
        // initialization expression
        int i = 1;

        // test expression
        while (i < 6) {
            System.out.println("Hello World");

            // update expression
            i++;
        }
    }
}
class forLoopDemo
```

```
{
    public static void main(String args[])
class dowhileloopDemo
{
    public static void main(String args[])
    {
        int x = 21;
        do
        {
            // The line will be printed even
            // if the condition is false
            System.out.println("Value of x:" + x);
            x++;
        }
        while (x < 20);
    }
}


    {
        // for loop begins when x=2
        // and runs till x <=4
        for (int x = 2; x <= 4; x++)
            System.out.println("Value of x:" + x);
    }
}
```

12. Чим оператор break відрізняється від оператора continue?
13. **Break leaves the loop completely and executes the statements after the loop**. Whereas Continue leaves the current iteration and executes with the next value in the loop. break completely exits the loop. continue skips the statements after the continue statement and keeps looping.

12. Навіщо у мові Java є мітки, якщо немає goto?

Java does not support goto, it is reserved as a keyword just in case they wanted to add it to a later version. Unlike C/C++, Java does not have goto statement, but java supports label. ... We can specify label name with break to break out a specific outer loop. Similarly, label name can be specified with continue

13. До якого результату призведе виконання наступного фрагменту коду? Перевірте. Поясніть.

```
boolean a = false;
boolean b = false;
if (a=false) {
   System.out.println("a is false");
}
if (b=true) {
   System.out.println("b is true");
}
if (a=b) {
   System.out.println("a = b");
}
```

ANSWER:

b is true

a = b

14. До якого результату призведе виконання наступного фрагменту коду? Перевірте. Поясніть.

```
int a = 1;
a++;
++a;
switch(a) {
   case 1: System.out.println("1");
   case 2: System.out.println("2");
   case 3: System.out.println("3");
   case 4: System.out.println("4");
}
```

ANSWER: 3-4

15. До якого результату призведе виконання наступного фрагменту коду (true/false)? Перевірте. Поясніть.

```
double sum = 0;
// Let's find: 1 + 1/2 + 1/3 + 1/4 + 1/5 + ...
for(int i=1; i<10; i++) {
   sum = sum + 1/i;
}
System.out.println(sum > 1);
```

ANSWER: false