



**КАФЕДРА ЦИФРОВОЇ ЕКОНОМІКИ ТА
БІЗНЕС-АНАЛІТИКИ**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА
ФАКУЛЬТЕТ УПРАВЛІННЯ ФІНАНСАМИ ТА БІЗНЕСУ

ЗАТВЕРДЖЕНО

на засіданні кафедри цифрової економіки
та бізнес-аналітики
протокол № 6 від “21” січня 2020 р.

Зав. кафедри _____ Шевчук І.Б.
(підпис)

**ЗАСОБИ ДІАГНОСТИКИ
З НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

Об’єктно-орієнтоване програмування

(назва навчальної дисципліни)

галузь знань: _____ 05 “Соціальні та поведінкові науки”
(шифр та найменування галузі знань)

спеціальність: _____ 051 “Економіка”
(код та найменування напрямку підготовки)

спеціалізація: _____ Інформаційні технології в бізнесі
(найменування спеціалізації)

освітній ступінь: _____ бакалавр
(бакалавр/магістр)

Укладач:

_____ Шевчук І.Б., зав. каф., д.е.н., доцент
(ПІБ, посада, науковий ступінь, вчене звання)

ЛЬВІВ 2020

Який правильний синтаксис для main методу?

- а. public void main()
- б. public static void main(String args)
- в. public void main(String[] args)
- г. public static void main(String[] args)

Що таке присвоєння?

- а. Додавання цілого до числа;
- б. Отримання добутку двох чисел;
- в. Призначення імені змінній;
- г. Призначення значення змінній.

Яке з цих слів не є ключовим словом у мові java?

- а. static
- б. String
- в. new
- г. try

Об'єкт це:

- а. Що завгодно;
- б. Алгоритм;
- в. Програма;
- г. Контейнер даних.

Ключові слова java можна писати в будь-якому регістрі – великими та малими літерами?

- а. Так.
- б. Ні.

Об'єктно-орієнтоване програмування це:

- а. Бути об'єктивним до своєї роботи;
- б. Створювати додаток, ґрунтуючись на об'єктах предметної області, виявлених під час вичення завдання;
- в. Спочатку придумати алгоритм написання коду і план тестування, а вже потім приступити до написання коду;
- г. Створювати додаток виключно із класів java.

Який ти змінної найкраще підходить для такого значення – 5.5?

- а. int
- б. double
- в. boolean
- г. string

Якщо умова полягає в тому, щоб обидві частини порівняння були правдиві, то який оператор потрібно використати?

- а. !

- б. ||
- в. &&
- г. !=

Що надрукує такий код: `INT[] NUMS = {1,2,3,4,5,6};`
`system.out.println(NUMS[1] +NUMS[3]);`

- а. 6
- б. 2+4
- в. 1+3
- г. 4

Абстрактні методи можуть бути

- а. В звичайному класі;
- б. Тільки в абстрактному класі
- в. Немає правильної відповіді.
- г. Всі відповіді правильні.

Коли клас наслідується від іншого класу, він отримує:

- а. Дані і методи базового класу;
- б. Методи і повідомлення базового класу;
- в. Тільки дані базового класу.

Як оголошується та ініціалізується масив?

- а. `int [] myList = {"1", "2", "3"};`
- б. `int [] myList = (5, 8, 2);`
- в. `int myList [] [] = {4,9,7,0};`
- г. `int myList [] = {4, 3, 7};`

Що виведе програма?

```
try{
int x = 0;
int y = 5 / x;
}catch (Exception e){
System.out.println("Exception");
} catch (ArithmeticException ae){
System.out.println("Arithmetic Exception");
}
System.out.println("finished");
```

- а. finished
- б. Exception
- в. Compilation fails
- г. Arithmetic Exception

Що виведе наступний фрагмент коду?

```
public void print() {  
    int t;  
    System.out.println(t);  
}
```

- а. Помилка компіляції;
- б. Помилка виконання;
- в. 0;
- г. Випадкове значення.

Що виведе наступний фрагмент коду?

```
int[] arr=new int[9];  
System.out.println(arr[0]);
```

- а. Помилка, так як індекс повинен бути більший 0;
- б. 0;
- в. Будь-яке значення;
- г. Помилка, так як масив не ініціалізований.

Що буде результатом виконання наступного методу?

```
public void print() {  
    String s1= «abc»;  
    String s2= «abc»;  
    System.out.println(«s1==s2 is:»+s1==s2);  
}
```

- а. s1==s2 is: true
- б. false
- в. s1==s2 is: false
- г. true

Що буде результатом виконання наступного методу?

```
public void print() {  
    List<String> list = new ArrayList<>();  
    list.add(«milk»);  
    list.add(«bread»);  
    list.add(«sausage»);  
    Stream<String> stream = list.stream();  
    list.add(«eggs»);  
    stream.forEach(System.out::println);  
}
```

- а. milk, bread, sausage, ConcurrentModificationException
- б. milk, bread, sausage, eggs

- в. milk, bread, sausage
- г. ConcurrentModificationException

Що буде результатом виконання наступного методу?

```
public void throwException() {  
    try {  
        throw 10;  
    }  
    catch(int e) {  
        System.out.println(«Got the Exception » + e);  
    }  
}
```

- а. Помилка компіляції;
- б. Got the Exception 0;
- в. Помилка виконання;
- г. Got the Exception 10.

Коли виконується блок finally{}?

- а. Тільки тоді, коли є будь-яка помилка в блоці try{ }
- б. Тільки тоді, коли є необроблювана помилка в блоці try{ }
- в. Блок finally виконується завжди перед виходом із блоку try/finally або try/catch/finally
- г. Завжди, коли метод завершує обробку

Що буде результатом виконання наступного методу?

```
public static void main(String[] args) {  
    try {  
        int a[]= {1, 2, 3, 4};  
        for (int i = 1; i <= 4; i++) {  
            System.out.println («a[» + i + «]=» + a[i] + «n»);  
        }  
    }  
    catch (Exception e) {  
        System.out.println («error = » + e);  
    }  
    catch (ArrayIndexOutOfBoundsException e) {  
        System.out.println («ArrayIndexOutOfBoundsException»);  
    }  
}
```

- а. Помилка компіляції
- б. Помилка ArrayIndexOutOfBoundsException
- в. Масив роздрукується

г. Загальна помилка виконання

Що буде результатом виконання наступного методу?

```
public class Test {  
    public void main(String[] args) {  
        int x = 10*20-20;  
        System.out.println(x);  
    }  
}
```

- а. Метод не виконається;
- б. Помилка виконання;
- в. Prints 0;
- г. 180

Що з перерахованого відповідає за конвертацію незалежного від платформи байт-коду в машинний код?

- а. JDK
- б. JVM
- в. JRE
- г. JIT

Що надрукує наступний код?

```
int i = 0;  
System.out.print(++i);  
System.out.print(i++);  
System.out.print(i);
```

- а. 112
- б. 122
- в. 011
- г. 012

Яким буде результат компіляції та виконання даного коду:

```
public class Switch {  
    public static void main(String[] args) {  
        int i = 3;  
        switch(i) {  
            default:  
                System.out.print("default");  
            case 1:  
                System.out.print("1");  
        }  
    }  
}
```

```
        case 2:
            System.out.print("2");
        }
    }
}
```

- a. default12
- б. default
- в. 1
- г. 2

Чи скомпіюється наступний код, якщо два класи будуть оголошені у файлі Test.java?

```
//Test.java
public class Car{
    public String myCar = "Ferrari" ;
}
public class Test{
    public static void main(String ... args){
        Car myCar = new Car();
        System.out.println(myCar.myCar);
    }
}
```

- a. Так
- б. Ні

Що виведе на консоль наступний код:

```
public class Main {
    public static void main (String []args) {
        int i = 0;
        i += i++;
        System.out.println(i);
    }
}
```

- a. 0
- б. 1
- в. 2
- г. Помилка компіляції

Яке твердження є вірним по відношенню до даного коду:

```
public class Person {
```

```

private String name;
public Person(String name) {
    this.name = name;
}
public boolean equals(Person p) {
    return p.name.equals(this.name);
}
}

```

- а. Метод equals неправильно перевизначає метод equals класу Object.
- б. Виникне помилка компіляції в методі equals, так як з цього методу не є p.name.
- в. Для того, щоб клас коректно працював в будь-якому контексті, він повинен перевизначити ще й метод hashCode класу Object
- г. У методі equals не вистачає виклику super.equals (this.name).

Каким будет результат выполнения данного кода?

```

public class Compare {
    public static void main(String[] args) {
        String s1 = new String("Hello");
        String s2 = new String("Hello");

        if (s1 == s2) {
            System.out.println("True");
        } else {
            System.out.println("False");
        }
    }
}

```

- а. True
- б. False
- в. Помилка компіляції
- г. Помилка часу виконання (RuntimeException)

Визначте значення змінної s.

```

int s = 1;
int k = 0;
do {
    s += 2;
} while(s != 100);
System.out.print("s= "+s);

```

- а. 99
- б. Нескінченність;
- в. 101

Визначте значення змінної s.

```
int s = 1;
for(int i = 5; i > 0; i--) {
s += i;
if(i != 0) { break; } }
System.out.print("s = " + s);
```

- а. 15
- б. 16
- в. 6
- г. 5

Що буде результатом виконання наступного коду?

```
boolean a = true;
boolean b = false;
boolean c = true;
if(a == true) {
if(b == true) {
    if(c == true) {
        System.out.println("1");
    } else System.out.println("2");
} else if(a&&(b=c)) {
    System.out.println("3");
}
else {
    System.out.println("4");
}
```

- а. 2.
- б. 4
- в. 3
- г. 1

Скільки разів буде виконано код?

```
short i = 32766;
do {
    System.out.print(i);
    i++;
} while(i < 32767);
```

- а. Помилка компіляції;
- б. Ітерація не виконується;
- в. Виконається одна ітерація;
- г. Занадто багато ітерацій.

Який вихід з наступного коду?

```
for(short i = 32766; i < 32767; i++) {  
    System.out.print(i);  
}
```

- а. Помилка компіляції;
- б. Ітерація не виконується;
- в. Виконається одна ітерація;
- г. Занадто багато ітерацій.

Який модифікатор доступу говорить про те, що це закритий член класу?

- а. private
- б. public
- в. protected
- г. friend

Результатом компіляції коду буде:

```
public class Overload{  
    public void method(Object o) {  
        System.out.println("Object");  
    }  
    public void method(java.io.FileNotFoundException f) {  
        System.out.println("FileNotFoundException");  
    }  
    public void method(java.io.IOException i) {  
        System.out.println("IOException");  
    }  
    public static void main(String args[]) {  
        Overload test = new Overload();  
        test.method(null);  
    }  
}
```

- а. Помилка компіляції
- б. Помилка часу виконання
- в. «Об'єкт»
- г. «FileNotFoundException»

Що буде виведено в результатах виконання даного фрагменту коду:

```
Float f1 = new Float(Float.NaN);  
Float f2 = new Float(Float.NaN);  
System.out.println( "" + (f1 == f2)+ " "+f1.equals(f2)+ " "+(Float.NaN ==  
Float.NaN) );
```

- а. false false false
- б. false true false
- в. true true false
- г. false true true

Що буде виведено в результаті виконання даного фрагменту коду:

```
class Mountain {
    static String name = "Himalaya";
    static Mountain getMountain() {
        System.out.println("Getting Name ");
        return null;
    }
    public static void main(String[ ] args) {
        System.out.println( getMountain().name );
    }
}
```

- а. Буде виведено «Himalaya», але не буде виведено «Getting Name»,
- б. Буде виведено «Getting Name» та «Himalaya»
- в. Нічого не буде виведено
- г. Буде виведено «Getting Name», а потім NullPointerException

Що буде виведено в результаті виконання даного фрагменту коду:

```
Integer a = 120;
Integer b = 120;
Integer c = 130;
Integer d = 130;
System.out.println(a==b);
System.out.println(c==d);
```

- а. true true
- б. false false
- в. false true
- г. true false

Даний код:

```
public static void doIt(String String) { //1
    int i = 10;
    i : for (int k = 0 ; k < 10; k++) { //2
        System.out.println( String + i); //3
        if( k*k > 10) continue i; //4
    }
}
```

- а. Не скомпілюється через рядок 1
- б. Не скомпілюється через рядок 2
- в. Не скомпілюється через рядок 3
- г. Скомпілюється та запуститься без проблем

Результатом виконання коду буде:

```
public class Main {
    static void method(int... a) {
        System.out.println("inside int...");
    }
    static void method(long a, long b) {
        System.out.println("inside long");
    }
    static void method(Integer a, Integer b) {
        System.out.println("inside INTEGER");
    }
    public static void main(String[] args) {
        int a = 2;
        int b = 3;
        method(a,b);
    }
}
```

- а. Помилка компіляції
- б. «inside int...»
- в. «inside long»
- г. «inside INTEGER»

Що буде виведено в результаті виконання даного фрагменту коду:

```
class Super { static String ID = "QBANK"; }
class Sub extends Super{
    static { System.out.print("In Sub"); }
}
class Test{
    public static void main(String[] args) {
        System.out.println(Sub.ID);
    }
}
```

- а. Він не скомпілюється
- б. Результат залежить від реалізації JVM
- в. Буде виведено «QBANK»
- г. Буде виведено «In Sub» та «QBANK»

Є два класи:

```
//in file A.java
package p1;
public class A{
    protected int i = 10;
```

```

    public int getI() { return i; }
}

//in file B.java
package p2;
import p1.*;
public class B extends A{
    public void process(A a) {
        a.i = a.i*2;
    }
    public static void main(String[] args) {
        A a = new B();
        B b = new B();
        b.process(a);
        System.out.println( a.getI() );
    }
}

```

- а. Буде виведено «20»
- б. Буде виведено «10»
- в. Код не скомпілюється
- г. Помилка часу виконання

Що таке ООП?

- а. методологія програмування, що базується на представленні програм у вигляді сукупності об'єктів, кожен із яких є екземпляром конкретного класу, а клас створює ієрархію наслідування;
- б. так називають будь-який тип програмування, в якому використовують поняття високого рівня і, на відміну від Асемблера, в якому не працюють напряму з комірками пам'яті ПК;
- в. просто гарне поняття. Якщо вдуматися, то воно не несе додаткового смислового навантаження, просто програмісти люблять аббревіатури, так сфери їх знань виглядають складнішими;
- г. особливо одинокий програміст.

Що таке клас?

- а. рівень складності програми. Всі оператори діляться на класи в залежності від складності їх використання;
- б. базовий елемент об'єктно-орієнтованого програмування в мові Java;
- в. просто одна з можливих назв змінної;
- г. таке поняття є тільки в C ++, в Java такого поняття немає.

Як оголосити клас в кодї?

- а. class MyClass { }
- б. new class MyClass { }

- в. `select * from class MyClass { }`
- г. `MyClass extends class { }`

Що виведеться в результаті даного коду?

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

if (a > b) {
    System.out.println("a > b");
} else if (a < b) {
    System.out.println("a < b");
} else {
    System.out.println("a = b");
}
```

- а. код написаний з помилкою, нічого не виведеться;
- б. `a > b`
- в. `a = b`
- г. `a < b`

Для чого використовується оператор NEW?

- а. для створення нової змінної;
- б. для оголошення нового класу;
- в. для створення екземпляра класу;
- г. це антагоніст оператора OLD.

Що виведе цей код?

```
String a = "my test";
String b = "this is";
String c;

switch (a) {
    case "MY_TEST":
        c = a + b;
        break;

    case "My Test":
        c = b + a;
        break;

    case "my_test":
        c = a + a;
        break;

    default:
        c = a + a;
}

System.out.println(c);
```

- а. `this is my test`
- б. `this isthis is`
- в. `my testmy test`
- г. `my test this is`

Що означає ключове слово `extends`? Що:

- а. дані клас успадковує від іншого;
- б. це додатковий модуль класу, який розширює його властивості;
- в. два клас роблять одне і те ж;
- г. це найбільший клас в програмі.

Що робить цей код?

```
int[] a = new int[] {4,5,6,7,8};
int[] b = new int[a.length];

for (int i = 0; i < a.length; ++i) {
    b[i] = a[i] * 2;
}
```

- а. подвоює всі значення масиву a;
- б. подвоює всі значення масиву b;
- в. заповнює масив b подвоєними значеннями масиву a;
- г. заповнює масив a подвоєними значеннями масиву b.

Код викличе помилку: Exception <...>
java.lang.ArrayIndexOutOfBoundsException:
Що вона означає?

```
int[] a = new int[] {1, 2, 3};
int[] b = new int[a.length];
b[4] = 4;
```

- а. переповнення пам'яті;
- б. вихід за рамки меж масиву;
- в. спроба вставити в масив некоректне значення;
- г. насправді цей код не викличе помилку.

Що означає перевантаження методу в Java (overload).

- а. зміна поведінки методу класу щодо батьківського;
- б. зміна поведінки методу класу щодо дочірнього;
- в. кілька методів з однаковою назвою, але різним набором параметрів;
- г. кілька різних класів з однаковим методом.

Що означає перевизначення методу в Java (override).

- а. зміна поведінки методу класу щодо батьківського;
- б. зміна поведінки методу класу щодо дочірнього;
- в. кілька методів з однаковою назвою, але різним набором параметрів;
- г. кілька різних класів з однаковим методом.

Чим відрізняються static-метод класу від звичайного методу класу?

- а. звичайний метод класу можна перевантажити, а static-метод не можна;
- б. звичайний метод класу можна перевизначити, а static-метод не можна;
- в. звичайний метод класу працює від об'єкта класу, а static-метод від всього класу;
- г. Static-метод класу можна викликати тільки всередині класу, а звичайний - в будь-якій частині коду.

Як викликати static-метод всередині звичайного?

- а. ніяк, static-метод можна викликати тільки від об'єкта класу;
- б. можна, треба перед цим перевантажити звичайний метод класу;
- в. можна, треба перед цим перевизначити звичайний метод класу;
- г. можна, нічого додатково робити не треба.

Як викликати звичайний метод класу всередині static-методу?

- а. ніяк, static-метод не працює з об'єктом класу;
- б. можна, треба перед цим перевантажити звичайний метод класу;
- в. можна, треба перед цим перевизначити звичайний метод класу;
- г. можна, нічого додатково робити не треба.

Для чого необхідно ключове слово this? Це:

- а. покажчик на перевизначений метод класу. Його не можна опускати при виклику, інакше перевизначення не спрацює;
- б. покажчик на поточний об'єкт класу всередині самого класу. Його можна опускати при виклику методу класу, але краще цього не робити;
- в. не ключове слово;
- г. ключове слово для виклику звичайного методу всередині static-методу. Його не можна опускати, інакше виклик не спрацює і буде помилка.

Що поверне метод, оголошений наступним чином:

```
public static int getAmount()
```

- а. не ясно, треба дивитися код методу;
- б. поверне static-поле класу;
- в. поверне посилання на об'єкт класу this;
- г. поверне цілочисельне значення.

Як виправити цей код, щоб він вивів "I Love Programming Tests" коректно?

```
String[] text = new String[] {
    "I ",
    "Love ",
    "Programming ",
    "Tests"
};

int i = 1;
while (i < text.length) {
    System.out.print(i);
}
```

- а. всередині while додати i ++ в кінці, інакше буде нескінченний цикл;
- б. в System.out.print передавати text [i];
- в. оголосити int i = 0;
- г. виконати всі попередні пункти.

Як можна знищити об'єкт в Java?

- а. цього не можна зробити вручну;
- б. присвоїти null всім посиланням на об'єкт;

- в. викликати метод finalize () до об'єкта;
- г. викликати деструктор об'єкта.

Виберіть всі правильні варіанти створення масивів.

- а. integer ia [][]=(4,5,6)(1,2,3);
- б. int[] ia=new int[15];
- в. float fa =new float [20];
- г. char[] ca="Some string"

Який результат буде отримано після компіляції і виконання даного коду:

```
public class Main {
    public static void main(String[] str) {
        outer:
        for (int i = 0; i < 2; i++) {
            for (int j = 0; j < 2; j++) {
                System.out.println("Hello");
                continue outer;
            }
            System.out.println("outer");
        }
        System.out.println("Good-Bye");
    }
}
```

- а. Hello
- б. Hello outer Hello outer Good-Bye
- в. Hello Hello Good-Bye
- г. Hello Good-Bye

Що буде виведено на екран?

```
public class Test {
    public static void main(String[] args) {
        int f;
        f = 5;
        f = ++f + f++;
        System.out.println(f);
    }
}
```

- а. 13
- б. 11
- в. 10
- г. 12

В якому рядку виникне перша помилка компіляції?

```
public class Test {
    /* 1 */ public final static class A {}
    /* 2 */ static private class B {}
    /* 3 */ abstract static class C {}
    /* 4 */ static final private class D {}
    /* 5 */ final public abstract class E {}
    /* 6 */ static final abstract class F {}
}
```

- а. 2
- б. 3
- в. 4
- г. 5

Що надрукує наступний код?

```

public class Test {

    private String name;

    Test(String name) {
        this.name = name;
    }

    public void test(final Test test) {
        test = new Test("three");
    }

    public String toString() {
        return name;
    }

    public static void main(String[] args) {
        Test t1 = new Test("one");
        Test t2 = new Test("two");
        t1.test(t2);
        System.out.println(t2);
    }
}

```

- а. Помилка компіляції
- б. two
- в. one
- г. three

На яку кількість різних об'єктів будуть посилатися змінні b1 - b11 після виконання даного коду:

```

Boolean b1 = true;
Boolean b2 = true;
Boolean b3 = true;
Boolean b4 = true;
Boolean b5 = true;
Boolean b6 = true;
Boolean b7 = true;
Boolean b8 = true;
Boolean b9 = new Boolean(true);
Boolean b10 = new Boolean(true);
Boolean b11 = new Boolean(true);

```

- а. 4
- б. 8
- в. 1
- г. 3

Дано код. Яким буде виведення на консоль?

```

public class A {
    public static void main(String[] args) {
        boolean a = true;
        boolean b = false;
        boolean c = true;
        if (a || b && c)
            System.out.print("Hello ");
        if (a && !b && c)
            System.out.print("World");
    }
}

```

- а. На консоль буде виведено "Hello World"
- б. На консоль буде виведено "Hello"
- в. Виникне помилка компіляції
- г. На консоль буде виведено " World"

Що буде надруковано в результаті виконання даного коду?

```

public class Cycle {
    public static void main(String...args) {
        int i = 1;
        do while(i < 1)
            System.out.println("i = " + ++i);
        while (i > 1);
    }
}

```

- а. Помилка компіляції
- б. i=1
- в. i=2
- г. Програма виконається, але нічого не виведе на екран

Що станеться в результаті компіляції і виконання програми:

```
class HashTest {
    private static Set<String> set = new LinkedHashSet<String>();
    public static void main(String[] args) {
        set.add("one");
        set.add("two");
        set.add("three");
        set.add("/u000a");
        set.add("/u000d");
        set.add("/u000c");
        set.add("1");
        set.add("2");
        set.add("3");
        for (String string : set) {
            System.out.print(string + " ");
        }
    }
}
```

- а. Помилка компіляції
- б. /u000a /u000c /u000d 1 2 3 one three two
- в. one three two /u000a / u000d /u000c /1 2 3
- г. u000c /

Який результат виконання наступного коду:

```
class A {
    public void m(int k) {
        System.out.println("class A, method m : " + ++k);
    }
}

class B extends A {
    public int m(int k) {
        System.out.println("class B, method m : " + k++);
        return k;
    }
}

public class MainClass {
    public static void main(String args[]) {
        A a = new B();
        a.m(34);
    }
}
```

- а. class A, method m : 34
- б. class A, method m : 35
- в. class B, method m : 34
- г. Помилка компіляції

Що буде результатом компіляції і виконання наступного коду?

```
public class Base {

    public Object print() {
        return "Object from base method";
    }

    public static void main(String[] args) {
        Base test = new Child();
        System.out.println(test.print());
    }
}

class Child extends Base {

    public String print() {
        return "String from child method";
    }
}
```

- а. Object from base method
- б. String from child method
- в. Помилка компіляції
- г. Помилка часу виконання

Яким буде результат компіляції і виконання наступного коду:

```
import java.util.List;
import java.util.ArrayList;

public class Test {
    public static void main(String[] args) {
        List<String> values = new ArrayList<String>() {
            {
                add("one");
                add("two");
                add("three");
            }
        };
        System.out.print("values: ");
        for (String value : values) {
            System.out.print(value + " ");
        }
    }
}
```

- а. Помилка компіляції
- б. Помилка часу виконання
- в. values:
- г. values: one two three

[Продовжити перегляд](#)

СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Алгоритми і структура даних: Навчальний посібник / В.М.Ткачук. - Івано-Франківськ : Видавництво Прикарпатського національного університету імені Василя Стефаника, 2016. 286 с.
2. Алгоритми та структури даних. Навчальний посібник / Т. О. Коротєєва. Львів : Видавництво Львівської політехніки, 2014. - 280 с.
3. Блинов И.Н., Романчик В. С. Java. Методы программирования : уч.-мет. пособие / И. Н. Блинов, В. С. Романчик. ? Минск : издательство "Четыре четверти", 2013. 896 с.
4. Блинов, И.Н. Java 2: практ. рук. / И.Н. Блинов, В.С. Романчик. - Мн.: УниверсалПресс, 2005. - 400 с.
5. Блинов, И.Н. Java. Промышленное программирование : практ. пособие / И.Н. Блинов, В.С. Романчик. - Минск : УниверсалПресс, 2007. - 704 с.
6. Васильев А. Н. Java. Объектно-ориентированное программирование: Учебное пособие. СПб.: Питер, 2011. 400 с.
7. Гамма, Э., Хелм, Р., Джонсон, Р., Влссидес, Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования. ? СПб. : Питер, 2007. ? 366 с.
8. Глоба Л. С. Розробка інформаційних ресурсів та систем [Електронний ресурс] : конспект лекцій / Л. С. Глоба, Т. М. Кот. - Київ : НТУУ "КПІ", 2014. - 318 с.
9. Грязнова В. О., Єфіменко С. В. Основи методології програмування. - К.: ВПЦ "Київський університет", 2010.
10. Давыдов В.Г. Программирование и основы алгоритмизации: Учеб. пособие. / В.Г. Давыдов. М.: Высш. шк., 2003. 447 с.
11. Інженерія якості програмного забезпечення: навч. посібник / Г.В Табунщик, Р.К. Кудерметов, Т.І. Брагіна. - Запоріжжя: ЗНТУ, 2013. - 180 с.
12. Кингсли-Хьюджес Э., Кингсли-Хьюджес К. Справочник программиста. - М.: ООО "ИД Вильямс", 2007.
13. Ларман, К. Применение UML 2.0 и шаблонов проектирования. Введение в объектно-ориентированный анализ и проектирование. 3-е изд. ? СПб. : Вильямс, 2012. 736 с.
14. Перри, Б. У. Java сервлеты и JSP. Сборник рецептов. М. : Кудиц-пресс, 2009. 768 с.
15. Рудаков А. В. Технология разработки программных продуктов : учебник для студ. сред. проф. образования / А. В. Рудаков. 7-е изд., стер. М. : Издательский центр "Академия", 2012. - 208 с.
16. Технології створення програмних продуктів та інформаційних систем : навч. посібник / М. Ю. Карпенко, Н. О. Манакова, І. О. Гавриленко ; Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова. - Харків : ХНУМГ ім. О. М. Бекетова, 2017. - 93 с.
17. Хорстманн, К. С., Корнелл, Г. Библиотека профессионала. Java 2 : Том 1. Основы. 8-е изд. М. : Вильямс, 2013. 816 с.
18. Шевчук І. Б. Інформаційні технології в регіональній економіці: теорія і практика впровадження та використання : монографія. Львів : Видавництво ННБК "АТБ", 2018. 448 с.

19. Шилдт Г. Полный справочник по Java SE 6. М.: Вильямс, 2010. 1040 с.
20. Эккель Б. Философия Java. 4-е изд. СПб. : Питер, 2011. 640 с.
21. Васильев А.Н. Самоучитель Java с примерами и программами. СПб.: Наука и Техника, 2011. 352 с.
22. Ковалюк Т.В. Основы програмування. / Ковалюк Т.В. Київ: ВНУ Києв, 2005. 400 с.
23. Николайчук Я. М. Проектування спеціалізованих комп'ютерних систем : навч. посібник / Я. М. Николайчук, Н. Я. Возна, І. Р. Пітух. - Тернопіль : ТзОВ "Терно-граф", 2010. - 392 с.
24. Хорстманн К.С., Корнелл Г. Java 2. Библиотека профессионала. Т. 2, Тонкости программирования. М.: Вильямс, 2010. 992 с.
25. Шилдт Г.б. Java: руководство для начинающих. М.: Вильямс, 2008. ? 720 с.
26. Java ? Учебник для начинающих программистов : [Электрон. ресурс]. - Режим доступа: <http://proglang.su/java>
27. Популярные технологии программирования в 2017 году : [Электрон. ресурс]. - Режим доступа: <https://vc.ru/dev/21483-what-language-2017>
28. Программирование на Java: [Электрон. ресурс]. - Режим доступа: <https://www.intuit.ru/studies/courses/16/16/info>
29. Рагулин П.Г. Информационные технологии: электронный учебник : [Электрон. ресурс]. - Режим доступа: http://window.edu.ru/catalog/pdf2txt/007/41007/18312?p_page=4
30. Теоретические основы технологии программирования: [Электрон. ресурс]. - Режим доступа: <http://bourabai.kz/alg/technology.htm>
31. Технология программирования : [Электрон. ресурс]. - Режим доступа: https://studref.com/441961/informatika/tehnologiya_programmirovaniya
32. Топ-5 полезных видеокурсов по Java : [Электрон. ресурс]. - Режим доступа: <https://javarush.ru/groups/posts/528-top-5-poleznychkh-videokursov-po-java-chastjh-1-->
33. Уроки по основам языка программирования JAVA для начинающих : [Электрон. ресурс]. - Режим доступа: <https://www.fandroid.info/tutorial-po-osnovam-yazyka-programmirovaniya-java-dlya-nachinayushhih/>
34. Энциклопедия языков программирования : [Электрон. ресурс]. - Режим доступа: <http://progopedia.ru/>
35. Язык программирования Java и среда NetBeans : [Электрон. ресурс]. - Режим доступа: <https://www.intuit.ru/studies/courses/569/425/info>