

```
Kintamasis: m1.m1A();
```

```
Turime dvi klases:  
class A {  
    public void m1A() {  
        System.out.print("a");  
    }  
}  
  
class B extends A {  
    public void m1A() {  
        System.out.print("b");  
        super.m1A();  
    }  
}
```

ba

Kokia metodo m1 paskirtis ir ką atlieka kodo eilutės A ir B ?

```
public void m1() {  
    Object[] a = this.toArray(); // A  
    Arrays.sort(a); // B  
    int i = 0;  
    for (Node<E> e1 = first; e1 != null; e1 = e1.next) {  
        e1.element = (E) a[i++];  
    }  
}
```

M1 Rikiuoja tiesinį sąrašą

Kodo eilutė B - rikiuoja masyvą

Kodo eilutė A Perraso tiesinio sąrašo reikšmes į masyvą

Kokia metodo m2 paskirtis ir ką atlieka kodo eilutė A ?

```
public E m2(int n) {  
    if (n < 0 || n >= size)  
        return null;  
    current = first.findNode(n); // A  
    return current.element;  
}
```

Metodas m2 Grąžina n-ojo elemento reikšmę tiesiniame sąraše

Kodo eilutė A Suranda n-ąjį mazgą tiesiniame sąraše

Kokia metodo m1 paskirtis ir ką atlieka kodo eilutės A ir B ?

```
public boolean m1(E e) {  
    if (e == null)  
        return false;  
    Node<E> e1 = new Node(e, null); // A  
    e1.next = first;  
    first = e1; // B  
    return true;  
}
```

Metodas m1 Įdeda naują elementą į tiesinio sąrašo pradžią

Kodo eilutė A Sukuria naują tiesinio sąrašo mazgą

Kodo eilutės B Nustato naują tiesinio sąrašo pradžią

Kokia metodo m1 paskirtis ir ką atlieka kodo eilutės A ir B ?

```
public boolean m1(E e) {  
    if (e == null)  
        return false;  
    Node<E> e1 = new Node(e, null); // A  
    e1.next = first;  
    first = e1; // B  
    return true;  
}
```

Metodas m1 Įdeda naują elementą į tiesinio sąrašo pradžią

Kodo eilutė A Sukuria naują tiesinio sąrašo mazgą

Kodo eilutės B Nustato naują tiesinio sąrašo pradžią

Kokia metodo m2 paskirtis ir ką atlieka kodo eilutė A ?

```
public Node<E> m2(int n) {  
    Node<E> e = this;  
    for (int i = 0; i < n; i++) {  
        e = e.next; // A  
    }  
    return e;  
}
```

Metodas m2 Grąžina n-ąjį tiesinio sąrašo mazgą

Kodo eilutė A Paima sekantį tiesinio sąrašo mazgą

Kokia metodo m1 paskirtis ir ką atlieka kodo eilutės A ir B ?

```
public void m1() {  
    Object[] a = this.toArray();  
    Arrays.sort(a); // A  
    int i = 0;  
    for (Node<E> e1 = first; e1 != null; e1 = e1.next) { // B  
        e1.element = (E) a[i++]; // B  
    } // B  
}
```

Metodas m1 Rikiuoja tiesinį sąrašą

Kodo eilutė A Rikiuoja masyvą

Kodo eilutės B Perrašo masyvą į tiesinį sąrašą

```
Ar duota klasė B gali paveldėti duotą klasę A (B extends A)?  
package paketas;  
final class A { }  
  
package paketas;  
class B { }
```

Ne

Ar klasė gali turėti private konstruktorių?

Taip, tai yra naudinga

```
Į kuriuos catch blokus įmanoma patekti?  
  
BufferedReader klaviatura = new BufferedReader(new InputStreamReader(System.in));  
try {  
    eil = klaviatura.readLine();  
    int x = Integer.parseInt(eil);  
    if (false) throw new FileNotFoundException();  
}  
catch (NumberFormatException e) { }  
catch (IndexOutOfBoundsException e) { }  
catch (EOFException e) { }  
catch (FileNotFoundException e) { }  
catch (IOException e) { }
```

NumberFormatException

IOException

Iš kur galima pasiekti klasės kintamąjį be jokio prieigos raktinio žodžio?

Iš tos pačios klasės | Iš bet kurios klasės iš to pačio paketo | Iš subklasės iš to pačio paketo

Iš kur galima pasiekti klasės private kintamąjį?

Iš tos pačios klasės

Iš kur galima pasiekti klasės **protected** kintamąjį?

Iš tos pačios klasės

Iš bet kurios klasės iš to pačio paketo

Iš subklasės iš to pačio paketo

Iš subklasės iš kito paketo

```
Jonas Petras = new Jonas();
Petras.substring(0,1);

Kokiai klasei priklauso metodas substring(0,1) ?
```

Jonas

Ką klasės kintamojo apraše reiškia raktinis žodis **final**?

Pradinė tokio kintamojo reikšmė negali būti pakeista

Kada klasė turi įdiegti **KTUable** interfeisą?

Kai klasės objektus planuojame talpinti į sąrašą **ListKTUx**.

```
Kas atsitiks kompiliuojant ir vykdant programą su tokiu kodo fragmentu?

String s = null;
if (s.equals("testas"))
    System.out.print("Gerai");
```

Susikompiliuos, tačiau programos vykdymo metu gausime **"NullPointerException"** klaidą

```
Kas atsitiks, jei pamėginsite sukompiliuoti ir paleisti programą?

public class Agg{
    static public long i=10;
    public static void main(String argv[]){
        switch(i){
            default:
                System.out.println("no value given");
            case 1:
                System.out.println("one");
            case 10:
                System.out.println("ten");
            case 5:
                System.out.println("five");
        }
    }
}
```

Išves "ten" ir po to "five"

```
Kas bus atspausdinta įvykdžius kodą:

A kintamasis = new A();
kintamasis.metaA();

Turime klasę:
class A {
    public String metaA() {
        System.out.print("a");
        return "a";
    }
}
```

A

```
Kas bus atspausdinta ivykdžius kodą:

A kintamasis = new B();
kintamasis.metaA();

Turime dvi klases:
class A {
    public void metaA() {
        System.out.print("a");
    }
}

class B extends A {
    public void metaA() {
        System.out.print("b");
        super.metaA();
    }
}
```

ba

```
Kas bus atspausdinta ivykdžius kodą:

A kintamasis = new B();
kintamasis.metaA();

Turime dvi klases:
class A {
    public void metaA() {
        System.out.print("a");
    }
}

class B extends A {
    public String metaA() {
        return "b";
    }
}
```

bus kompiliavimo klaida

```
Kas bus atspausdinta ivykdžius kodą:

A kintamasis = new B();
kintamasis.metaA();

Turime dvi klases:
class A {
    public void metaA() {
        System.out.print("a");
    }
}

class B extends A {
    public void meta(int b) {
        System.out.print("b");
    }
}
```

a

```
Kas bus atspausdinta ivykdžius kodą:

B kintamasis = new B();
kintamasis.metaA();

Turime dvi klases:
class A {
    public void metaA() {
        System.out.println("a");
    }
}

class B extends A {
    public void metB() {
        System.out.println("b");
    }
}
```

a

```
Kas bus atspausdinta ivykdžius kodą:

B kintamasis = new B();
kintamasis.metaA();

Turime dvi klases:
class A {
    public void metaA() {
        System.out.print("a");
    }
}

class B extends A {
    public void meta(int b) {
        System.out.print("b");
    }
}
```

a

```
Kas bus atspausdinta įvykdžius kodą:  
A kintamasis = new B();  
kintamasis.meta();  
  
Turime dvi klases:  
  
class A {  
    public void meta() {  
        System.out.println("a");  
    }  
}  
  
class B extends A {  
    public void meta() {  
        System.out.println("b");  
    }  
}
```

B

```
Kas bus atspausdinta įvykdžius šį kodo fragmentą?  
  
String stri = new String("Stringas yra eilutė");  
Scanner sc = new Scanner(stri);  
sc.useDelimiter("yra");  
System.out.println (sc.next());
```

Stringas

```
Kas bus atspausdinta po šio programos fragmento?  
  
public void amethod(){  
    int k = 10;  
    switch(k){  
        case 10:  
            System.out.print("ten");  
        case 20:  
            System.out.print("twenty");  
        break;  
        default:  
            System.out.print("default");  
        break;  
    }  
}
```

Tentwenty

```
int mas[] = new int[5];  
try {  
    mas[3] = (int)Double.parseDouble ("5");  
    if (false)  
        throw new EOFException("");  
    System.out.println ("Klaidos nera");  
}  
catch (NumberFormatException e) {  
    System.out.println ("Blogai");  
}  
catch (IndexOutOfBoundsException e) {  
    System.out.println ("Klaida");  
}  
catch (EOFException e) {  
    System.out.println ("Negerai");  
}
```

Numatytasis pažymys*

1

Bendras grįžtamasis ryšys (?)



Klaidos nera

```
try {
    mas[5] = (int)Double.parseDouble("5");
    if (true)
        throw new EOFException("");
    System.out.println ("Klaidos nera");
}
catch (NumberFormatException e) {
    System.out.println ("Blogai");
}
catch (IndexOutOfBoundsException e) {
    System.out.println ("Klaida");
}
catch (EOFException e) {
    System.out.println ("Negerai");
}
```

Klaida

Kas bus atspausdinta po tokių sakinių įvykdymo?

```
class KintamiejiIrMetodai {
    int x;
    public void keisti (int x) {
        this.x = x - 1;
        x++;
        System.out.println ("x = " + x);
    }
}
```

```
KintamiejiIrMetodai kim = KintamiejiIrMetodai();
kim.keisti(7);
```

x = 8

Kas bus atspausdinta sukompiliavus ir įvykdžius toki javos kodą?

```
class A {
    void spausdinti() { System.out.println("Tėvas"); }
}

class B extends A {
    void spausdinti() { System.out.println("Vaikas"); }

    public static void main (String a[]) {
        A obj = new B();
        obj.spausdinti();
    }
}
```

Vaikas

Kas bus atspausdinta sukompiliavus ir įvykdžius javos kodą?

```
class A {
    void spausdinti(double i, double j) {
        System.out.println("Tėvas");
    }
}

class B extends A {
    void spausdinti(double i) {
        System.out.println("Vaikas");
    }

    public static void main (String a[]) {
        B obj = new B();
        obj.spausdinti(5, 5);
    }
}
```

Tėvas

Kas gali turėti **abstract** metodus?

Abstrakčios klasės

Sąsajos

Kas gali turėti **static** tipo kintamųjų?

Klasės

Abstrakčios klasės

Sąsajos

Kas šiame klasės apraše yra blogai?

```
class Metodai {  
    double y;  
    public void funkcija (double x) {  
        z = Math.sqrt(x);  
    }  
}
```

klasės metode yra neaprašytas kintamasis z

Kas šiame užrašė yra next() ?

```
sar = sarasas.next().doCheck();
```

Metodas

Kas turi būti parašyta vietoj komentaro ženklų prie kintamojo x ?

```
class A {  
    /* */ x;  
    public static void main (String args[]) {  
        x = (int)Double.parseDouble("5.5");  
    }  
}
```

static int

Kiek .java faile gali būti klasių?

Java kalboje tai neribojama

Kiek daugiausiai klasių Javos kalboje gali paveldėti kita klasė?

Vieną

Kiek daugiausiai sąsajų (interface) Javos kalboje gali įdiegti klasė?

Bet kiek

Kiek objektų sukurama šioje eilutėje?

```
BufferedReader klaviatura = new BufferedReader(new InputStreamReader(System.in));
```

2

Kiek viename .java faile gali būti **public** klasių aprašų (ne vidinių klasių)?

1

Kurie teiginiai apie klasės ListKTU antraštę yra teisingi:

```
public class ListKTU<E extends Comparable<E>> implements ListADT<E>, Iterable<E>, Cloneable {
```

Klasė **ListKTU** įdiegia interfeisus **Cloneable** ir **Iterable**

Bendriniu tipu **E** perduodama klasė privalo įdiegti interfeisą **Comparable**

Klasė **ListKTU** įdiegia interfeisą **ListADT**

Kokie rezultatai laukiami vykdant šį metodą (klasė ListKTU įdiegia interfeisą Iterable<E>)?

```
public static void calculate(ListKTU<Žaidėjas> team){
    int sum = 0;
    Žaidėjas y = new Žaidėjas("Naujas", 0, 0);
    for (Žaidėjas x : team) {
        sum += x.getPataikyte();
        x = y;
    }
    System.out.println("Atsakymas="+sum);
}
```

Atspausdins visų komandos žaidėjų pataikytų metimų sumą ir visi komandos žaidėjai liks tokie patys.

Kokio dydžio (bitais) yra byte tipo kintamasis?

8

Kokio dydžio (bitais) yra char tipo kintamasis?

16

Kokio dydžio (bitais) yra int tipo kintamasis?

32

Kokio dydžio (bitais) yra long tipo kintamasis?

64

Kokio dydžio (bitais) yra short tipo kintamasis?

16

Kokiu simboliu java kalboje baigiasi sąsajos (interface) metodo aprašas?

;

Kokius prieigos raktinius žodžius gali turėti sąsajos (interface) metodas?

Gali neturėti jokio

Public

```
try {
    Scanner s = new Scanner ("abcde 51 4c");
    String str = s.next();
    str = s.next();
    System.out.print (str + " ");
    int i = s.nextInt();
    System.out.print (i + " ");
    i = s.nextInt();
}
catch (InputMismatchException e) {
    System.out.print ("Klaida");
}
catch (NoSuchElementException e) {
    System.out.print ("Trūksta elemento");
}
```

51 Klaida

Koks yra ArrayList klasės metodo `get(int i)` asimptotinis įvertinimas?

O(1)

Koks yra ArrayList klasės metodo `indexOf(Object o)` asimptotinis įvertinimas?

O(n)

Koks yra Arrays klasės metodo `binarySearch(Object[] a, Object key)` asimptotinis įvertinimas?

$O(\log n)$

Koks yra LinkedList klasės metodo `get(int i)` asimptotinis įvertinimas?

$O(n)$

Koks yra LinkedList klasės metodo `indexOf(Object o)` asimptotinis įvertinimas?

$O(n)$

Koks yra rikiavimo **burbuliuko** metodo asimptotinis įvertinimas?

$O(n^2)$

Koks yra rikiavimo **merge** metodo asimptotinis įvertinimas?

$O(n \cdot \log n)$

Koks yra rikiavimo **min-max** metodo (**insertion**) asimptotinis įvertinimas?

$O(n^2)$

Kuriai klasei priklauso metodas `length()` ?

```
class A {
    public static void main (String args[])
    {
        String eil = "Scanner";
        int x = eil.length();
    }
}
```

String

Kurie ciklai yra amžini?

`for (int i=0; ; i++) { i++; }`

`for (; ;) { if (false) break; }`

`for (; ;) { }`

Kurie dvimačio masyvo aprašymo būdai yra teisingi?

`int matrica[][]=new int [3][];`

`int matrica[][]={{0,1,2},{4,5,6},{7,8}};`

`int matrica[][] = new int [3][3];`

Kurie iš pateiktų kintamųjų aprašų yra teisingi?

`int _a;`

`int _____2_w;`

`int _$;`

`int this_is_a_very_detailed_name_for_an_identifier;`

Kurie masyvo sukūrimo būdai yra **neteisingi**?

`int masyvas = new int[10];`

`int [10]masyvas = new int;`

`int masyvas[10] = new int[];`

`double d[] = new Double[5];`

Kurie teiginiai apie duotą programos fragmentą yra teisingi?

```
class A {  
}  
  
class B {  
    public static void main (String a[]) {  
        int i;  
        A kintamasis1;  
        kintamasis1 = new A(i);  
        B kintamasis2 = new B();  
    }  
}
```

B() yra konstruktorius pagal nutylėjimą

A(i) yra vieta, kurioje įvyks klaida

kintamasis2 yra objektas

Kurie teiginiai apie šią programą yra teisingi?

```
class A {  
    int i;  
    A (int ii) {  
        i = ii;  
    }  
    public static void main (String args[]) {  
        A obj = new A();  
        System.out.println (obj.i);  
        return;  
    }  
}
```

Būtinai turi būti aprašytas konstruktorius pagal nutylėjimą (be parametų)

Eilutėje "A obj = new A();" įvyks klaida

Kurios iš pateiktų eilučių, įrašius ją į kodo fragmentą vietoj komentaro, išspausdintų nulį ?

```
public class ZeroPrint {  
    public static void main(String argv[]) {  
        int i = 0;  
        // eilutę įrašyti čia  
    }  
}
```

`System.out.println(i++);`

`System.out.println(i);`

`System.out.println(i--);`

Kurios išraiškos yra teisingos?

`System.out.println(1+1);`

`int i = 2 + '2';`

`String s = "on" +123;`

```

ListKTU <Žaidėjas> team = new ListKTU <>();
Iterator <Žaidėjas> it = team.iterator();
while (it.hasNext()) {
    Žaidėjas x = it.next() ;
    // - naudojame elementą x
}
//----- variantas B -----
ListKTU <Žaidėjas> team = new ListKTU <>();
for (Iterator <Žaidėjas> it = team.iterator(); it.hasNext(); ){
    Žaidėjas x = it.next() ;
    // ... naudojame elementą x
}
//----- variantas C -----
ListKTU <Žaidėjas> team = new ListKTU <>();
for (int i=0; i<team.size(); i++){
    Žaidėjas x = team.get(i) ;
}

```

variantas C

Kuriuos metodus apibrėžia interfeisas **Iterator**?

boolean hasNext()

E next()

void remove()

Pažymėkite metodus, kurie priklauso bazinei klasei **ListKTU**:

E get(int ind)

void clear()

boolean add(E data)

int size()

Pažymėkite metodus, kurie priklauso išplėstinei klasei **ListKTUx**:

void load (String fName)

void save (String fName)

void println (String title)

void add (String dataString)

Pažymėkite tuos rikiavimo algoritmus, kurio asimptotinis įvertinimas $O(N^2)$

MinMax (selection)

Burbulo (bubble)

Įterpimo (insertion)

```

Reader ivedimas = new BufferedReader(new InputStreamReader(System.in));

Kokio tipo objektas yra ivedimas?

```

BufferedReader

```

Turime duotą klasės aprašą

public class MyClass { .... }

Koks gali būti failo, kuriame ši klasė išsaugota, pavadinimas?

```

MyClass.java

Turime klasės

```
public MyClass extends Vector { ... }
```

.java failą, kuris saugomas com.java.test pakete.

Kas turi būti parašyta pirmoje šio javos failo eilutėje (neskaitant komentarų)?

package com.java.test;

Turime klasių A ir B aprašus. Kiek mažiausiai klasė B gali turėti metodų?

```
abstract class A {  
    public void metaA(){}  
    public abstract void metB();  
}
```

```
final class B extends A { }
```

1

Turime sąsajos A ir klasės B aprašus. Kiek mažiausiai metodų gali turėti klasė B ?

```
interface A {  
    public void metaA();  
}
```

```
final class B implements A {  
    // metodai  
}
```

1