

Java - PDF z odpowiedziami (C) Grelix

Część odpowiedzi może zawierać błędy!

Jaka klasa w graficznym interfejsie użytkownika odpowiedzialna jest za czcionki?
Font

Który menedżer rozkładu odpowiada za rozmieszczenie elementów w kartach?
CardLayout

Jak możemy umiejscowić elementy w menedżerze rozkładu FlowLayout?
Nie możemy, menedżer sam określa ich rozmieszczenie

W jakim pakiecie dostępne są klasy i interfejsy związane z obsługą zdarzeń?
java.awt.event

Jaki komponent odpowiada za etykiety tekstowe?
JLabel

Jaki jest typ parametru we wszystkich metodach zdefiniowanych w interfejsie MouseListener?
MouseEvent

Jaki jest domyślny menedżer rozkładu w komponencie JPanel?
FlowLayout

Który z komponentów umożliwia zaprezentowanie w formie graficznej danych w postaci hierarchicznej?
JTree

Które pakiety są związane z graficznym interfejsem użytkownika?
java.awt
javax.swing
java.awt.event

Czy obiekty klasy JFrame są domyślnie widoczne na ekranie?
Fałsz

Komponent odpowiedzialny za podpowiedź to w Swingu?
JToolTip

Zmiana stanu komponentu związana jest z interfejsem?
ActionListener

Z jakim interfejsem związany jest wybór wartości z listy?
ListSelectionListener

Która metoda umożliwia zmianę rozmiaru okna?
setSize

Który z menedżerów rozkładu umożliwia rozmieszczenie komponentów wg kierunków geograficznych?
BorderLayout

Czy komponent JCheckBox jest tak zwanym komponentem lekkim?
Prawda

Jaka klasa w graficznym interfejsie użytkownika odpowiedzialna jest za czcionki?
Font

Która metoda umożliwia zmianę widoczności okna?
setVisible

Jak możemy umiejscowić elementy w menedżerze rozkładu FlowLayout?
Nie możemy, menedżer sam określa ich rozmieszczenie

Czy etykiety tekstowe mogą posiadać ikony?
Prawda

Który menedżer rozkładu odpowiada za rozmieszczenie elementów w kartach?
CardLayout

W jakim pakiecie dostępne są klasy i interfejsy związane z obsługą zdarzeń?
java.awt.event

Podaj nazwę klasy umożliwiającej odczyt zawartości pliku tekstowego.
FileReader

Jaki będzie rezultat działania poniższego programu:

```
import java.io.*;
public class MojProgram {
    public static void main(String[] args){
        String nazwaFolderu = "C:"+File.separator;
        File folder = new File(nazwaFolderu);
        ...
    }
}
```

...

Zostanie wyświetlona zawartość folderu głównego dysku systemowego

Czym charakteryzuje się wyjątek kontrolowany?
Wywodzi się z klasy Exception
Musi zostać obsłużony w kodzie programu

Klasa File zawiera metodę isDirectory(), umożliwiającą sprawdzenie, czy wskazywany zasób jest folderem. Jakiego typ wartości zwraca ta metoda?
boolean

Wskaż wyjątki niekontrolowane:
ArrayIndexOutOfBoundsException
OutOfMemoryError
StackOverflowError

W skład klasy File wchodzi metoda umożliwiająca utworzenie nowego folderu. Podaj jej nazwę.
mkdir

Jaki będzie rezultat działania programu:

```
public class MyProg{
    public static void main(String[] args){
        int x = 0;
        int y = 5/x;
        System.out.println(x);
    }
}
```

zgłoszony wyjątek ArithmeticException (dzielenie przez zero)

Co oznacza skrót URL?

Uniform Resource Locator

Podaj nazwę klasy reprezentującej w systemie pojedynczy plik lub katalog oraz umożliwiającą,

dzięki dostępnym w niej metodom, operowanie na plikach i katalogach systemu.

File

Wskaż klasy bazowe odnoszące się do wyjściowego strumienia danych

java.io.Writer

java.io.OutputStream

Podaj nazwę klasy zawartej w pakiecie java.net, umożliwiającą dostęp do zasobów zgromadzonych

w sieci Internet.

URL

Podaj nazwę bloku instrukcji, który zostanie wykonany niezależnie od powstałych w bloku try{}

błędów:

finally

Jaki będzie rezultat działania poniższego programu:

```
import java.net.*;
```

```
import java.io.*;
```

```
public class ZasobyInternetu{
```

```
    public static void main(String[] args){
```

```
        String adres = "http://www.uek.krakow.pl";
```

```
        try
```

```
        {
```

```
            URL url = new URL(adres);
```

```
            ...
```

```
        }
    }
}
```

wyświetlenie zawartości zasobu z adresu http://www.uek.krakow.pl

Jaką funkcję pełni instrukcja 'throw'?

Przechwytuje wyjątki kontrolowane

Umożliwia programowe zgłoszenie wyjątku

Jaki wyjątek zgłasza metoda charAt() klasy String? Sprawdź w dokumentacji Javy.

IndexOutOfBoundsException

Instrukcja try..catch:

Umożliwia przechwycenie zgłoszonego przez program wyjątku

Która z wymienionych klas jest nadklasą dla klasy Exception?

Throwable

Czy poniższy kod jest poprawny?

```
interface Testowy {
```

```
    public int dodaj(int a, int b);
```

```
    public void toString();
```

```
}
```

```
class Test implements Testowy {
```

```
    public int dodaj(int a, int b) {
```

```
        return a+b;
```

```
    }
```

```
}
```

Prawda

Czy klasa String jest klasą finalną?

Prawda

Jaką nazwę posiadają klasy anonimowe?

Nie posiadają żadnej nazwy

```
interface Base {  
    boolean m1 ();  
    byte m2(short s);  
}
```

Które z poniższych fragmentów kodu skompilują się?

```
abstract class Class2 implements Base{}
```

Czy klasy opakowujące są klasami finalnymi?

Prawda

Czy klasa wewnętrzna posiada dostęp do składowych prywatnych klasy, w której została

utworzona?

Prawda

Klasa implementująca interfejs musi utworzyć

Wszystkie metody zdefiniowane w tym interfejsie

```
interface DoMath {  
    double getArea(int rad_;  
    ...
```

Który z poniższych kodów pozwoli na kompilację programu?

```
Abstract class AllMath implements DoMath, MathPlus {public double getArea(int rad) { return  
  
rad * rad * 3.14; }}
```

Czy można zdefiniować klasy wewnętrzne wewnątrz metod?

Prawda

Które z poniższych deklaracji są błędne (KlasaA, KlasaB, KlasaC oznaczają nazwy klas, natomiast

InterfejsA, InterfejsB, InterfejsC to nazwy interfejsów)?

```
Class KlasaC extends KlasaA, KlasaB
```

```
Interface InterfejsB extends InterfaceA
```

```
Interface InterfejsA implements InterfejsB
```

```
Interface InterfejsC extends InterfaceA, InterfaceB
```

Czy klasa wewnętrzna posiada dostęp do składowych prywatnych klasy, w której została

utworzona?

Prawda

Która deklaracja jest prawidłowa w ciele interfejsu?

```
Public static short stop = 23;
```

Jakie interfejsy implementuje klasa String?

Serializable

CharSequence

Comparable

Które z poniższych zdań są prawdziwe:

Wszystkie metody w klasie abstrakcyjnej muszą być zadeklarowane jako abstrakcyjne
Modyfikator "final" użyty w deklaracji klasy oznacza, iż klasa nie może być dziedziczona

Które trzy nagłówki metod zamieszczone w ciele interfejsu są prawidłowe?

```
Boolean setFlag(Boolean [] test);  
Public void main(String [] args);  
Public float getVol(floatx);
```

Ile interfejsów może implementować klasa?

Dowolną liczbę

Co się stanie jak uruchomisz poniższy program?

```
public class Outer {  
    public String name = "Outer";
```

...

Błąd kompilacji w miejscu tworzenia instancji klasy Inner

Czy możliwe jest tworzenie obiektów na bazie klasy abstrakcyjnej?

Fałsz

W jakim przypadku klasa implementująca interfejs nie musi tworzyć wszystkich metod

występujących w tym interfejsie?

Nie ma takiej możliwości

Które z atrybutów elementu <applet> należy bezwzględnie umieścić na tworzonej stronie

internetowej zawierającej aplet?

code, height,width

Podaj nazwę metody umożliwiającej rysowanie łańcucha znaków:

drawString

Który z pakietów Javy musi zostać zaimportowany w celu dostępu do większości komponentów

graficznych?

java.awt.*

Które wywołanie metody drawLine spowoduje narysowanie linii pomiędzy punktem x=20, y=30 a

punktem x=70, y=50?

gr.drawLine(20,30,70,50);

Określi prawidłową kolejność czynności prowadzących do wyświetlenia apletu w oknie

przeglądarki internetowej:

utworzenie kodu w języku Java

kompilacja utworzonych plików źródłowych

utworzenie dokumentu HTML

otwarcie dokumentu HTML w przeglądarce internetowej

Który z pakietów powinien zostać zaimportowany dla dostępu do klasy Applet i jej składowych?

java.applet.Applet

Wskaż poprawne stwierdzenia związane z apletem:
Możliwe jest uruchomienie apletu w przeglądarce internetowej

W jakich jednostkach wyrażane są wymiary apletu?
Piksel

Które z poniższych metod nie są dziedziczone z klasy Aplet:
Cancel

Które z podanych poniżej stwierdzeń dotyczących apletu i aplikacji są prawdziwe?
Aplety nie posiadają metody main
Aplety zwykle posiadają ograniczenia w dostępie do lokalnych zasobów komputera

Podaj nazwę atrybutu elementu APPLET określającego plik zawierający kod programu w języku

java.
code

W celu narysowania prostokąta wypełnionego kolorem, jak w bieżących ustawieniach, należy użyć

metody:
fillRect()

Podaj nazwę metody, która wywoływana jest po załadowaniu apletu:
start

Podaj nazwę metody wywoływanej podczas ładowania apletu:
init

Zmienna 'g' wskazuje na obiekt klasy Graphics. W celu zmiany koloru rysowania na czerwony

należy użyć:
g.setColor(Color.red)

Jaka metoda apletu jest wywoływana w momencie jego rysowania na ekranie monitora?
Paint()

Co powinno być wartością atrybutu 'code' zawartego w elemencie 'applet':
nazwa pliku (.class) zawierającego kod programu do wykonania

Jaki będzie rezultat działania poniższego programu:

```
class B{
    static int x = 10;
}
public class A extends B{
    public static void main(String[] args){
        B klasaB = new B();
    }
}
...
20 20
```

Które z podanych stwierdzeń są prawdziwe:
Nie jest możliwe dziedziczenie z więcej, niż jednej klasy bazowej

Dziedziczenie realizowane jest poprzez użycie słowa kluczowego 'extends'

Realizacja dziedziczenia polega na przejęciu na nowo tworzoną klasę wszelkich cech i zachowań ze wskazanej klasy już istniejącej

Jaki będzie efekt kompilacji i uruchomienia programu:

```
class Pojazd{
    public void ruszaj(){
        System.out.println("Pojazd: ruszaj");
    }
}
class Auto extends Pojazd{
    ...
}
Pojazd: ruszaj Auto: ruszaj Auto:ruszaj
```

Czy język programowania Java umożliwia dziedziczenie na podstawie wielu klas (tzw.

dziedziczenie wielobazowe)?

Nie

Gdzie w konstruktorze można umieścić odwołanie do konstruktora zdefiniowanego w klasie

nadrzędnej?

Jako pierwszą instrukcją konstruktora

Dziedziczenie realizowane jest poprzez użycie słowa kluczowego:

extends

Tworzony program zawiera następującą instrukcję:

```
x = Math.round(25.4);
```

Czy powyższa instrukcja jest poprawna, jeśli wcześniej nie zostanie wykonana instrukcja:

```
import java.lang
```

Tak

Jakie słowo kluczowe umożliwia dostęp do składowych klasy bazowej z klasy dziedziczącej?

super

Dopasuj nazwy metod klasy Object do ich znaczenia

Zwraca reprezentację obiektu w formie łańcucha tekstowego toString()

Zwraca nazwę klasy na podstawie której powstał obiekt getClass()

Tworzy kopię obiektu clone()

porównuje dwa obiekty equals(Object)

Jaki będzie rezultat kompilacji i uruchomienia poniższego programu:

```
class Komunikator{
    public final void zadzwon(){
        System.out.println("drnnn");
    }
}
public class Telefon extends Komunikator{
    public static void main
    ...
}
wyświetlenie 'drnnn'
```

Wskaż słowa kluczowe, które odnoszą się do dziedziczenia klas:

Extends

Super

Jaki będzie rezultat uruchomienia programu:

```
class A
{
    final public int metoda(int a, int b) {return 0;}
}
class B extends A
...
```

Błąd kompilacji (metoda finalna nie może być przeciążona)

Które z podanych linii umieszczonych w miejscu komentarza nie spowodują błędu kompilacji i

wykonania:

```
class A{}
public class B extends A{
    static boolean b1=false;
    static int i = -1;
    static double d = 10.1;
    public static void main
...
m=b;
```

Wywołanie metody dodaj w taki sposób:

Obliczenia.dodaj(14, 34);

oznacza, że metoda dodaj jest:

Instancyjna

Statyczna

Z jakim specyfikatorem dostępu związana jest enkapsulacja/hermetyzacja danych?

private

Jak informacja zostanie wyświetlona na konsoli po wykonaniu poniższego kodu?

Student[] studenci = new Student[10];

System.out.println(studenci[4]);

Pusta linia

Ile można stworzyć konstruktorów danej klasy?

Dowolną liczbę

Aby odwołać się z wnętrza metody do obiektu na rzecz którego ta metoda została wywołana

stosujemy słowo kluczowe

This

Jaki operator umożliwia tworzenie obiektów?

new

Jaki nagłówek powinna mieć metoda dostępowa do pola nazwisko?

```
class Student {  
    private String nazwisko;  
}  
public String pobierzNazwisko()
```

Co zwraca konstruktor danej klasy?
Referencję do obiektu danej klasy

Która z podanych domyślnych wartości dla określonego typu jest nieprawdziwa?
Pies -> null
boolean -> true
int -> 0

Dlaczego w Javie można utworzyć tylko trzy konstruktory w danej klasie?
w Javie można utworzyć dowolną liczbę konstruktorów w danej klasie

Do wizualnej reprezentacji klas i ich zależności stosuje się diagramy:
UML

Jaki specyfikator dostępu ma konstruktor danej klasy?
Public

W kodzie źródłowym jak wiele klas można zadeklarować jako publiczne?
Tak wiele jak potrzebujemy

Nazwa konstruktora danej klasy
jest taka sama jak nazwa klasy

Dostęp do zmiennych prywatnych danej klasy posiadają
Wszystkie metody występujące w tej klasie

Które z poniższych stwierdzeń są prawdziwe:
Konstruktor nie może zwracać wartości
Dopuszczalny jest brak jawnie zadeklarowanego konstruktora
Nazwa konstruktora musi być zgodna z nazwą klasy

Stosowanie metod dostępowych i modyfikujących związane jest z pojęciem:
enkapsulacji/hermetyzacji danych

Co się stanie kiedy skompilujesz i uruchomisz poniższy program?

```
public class MyMain{
    public static void mian(String argv) {
        System.out.println("Witaj świecie!");
    }
}
```

Program się nie skompiluje, błędna forma metody main

Które z poniższych sposobów zapisu komentarzy nie są poprawne?

/ treść komentarza/

Które z wymienionych nazw zmiennych są zgodne z przyjętą konwencją tworzenia nazw:

nrNadwozia
silnik
kolorKaroserii
markaPojazdu
hamuj

Aby skompilować aplikację Test.java w linii poleceń należy wydać następujące polecenie?

javac Test.java

Zintegrowane środowisko programistyczne to inaczej?

IDE

Które z poniższych komend są poprawne:

javac MojProgram.java
java MojProgram

Jakie deklaracje są ostateczne dla stworzenia działającej aplikacji w języku Java?

deklaracja klasy, konstruktora i metody main()

Jaką nazwę powinien mieć plik z kodem źródłowym zawierającym następujący program:

```
public class Madryt{
    public static void main (String[] args) {
        System.out.println("Hiszpania to piękny kraj" );
    }
}
```

Madryt.java

Znaki w Javie kodowane są w standardzie

Unicode

Które instrukcje deklaracji tablicy są prawidłowe:

String tab[] = {"a","b","c"};

Które z poniższych instrukcji są poprawne:

double d=999d;

Jakie będzie rezultat działania poniższego programu:

```
public class Rand{
    public static void main(String argv[]){
        int iRand;
```

```

        iRand=Math.random();
        System.out.println(iRand);
    }
}

```

Które z poniższych są modyfikatorami dostępu:

public
private
protected

Metoda max potrzebuje dwóch parametrów typu int. Poprawny nagłówek tej metody to?

public void max(int a,b)
public void max(int a, int b)

Definicja i ciało metody

nie może wystąpić poza klasą

może wystąpić w ciele innej metody

Klasa StringTokenizer służy do:

dzielenia łańcuchów tekstowych na poszczególne znaki

Zmienna s zawiera łańcuch znaków "Java". Aby określić pozycję znaku "v" w podanym łańcuchu użyjesz instrukcji:

s.indexOf('v');

Jaki będzie rezultat wykonania programu:

```

public class Obliczenie{
    public static void main(String[] args){
        int liczba1=4;
        int liczba2=10;
        wykonajOperacje(liczba1, liczba2);
        System.out.println(liczba1+" "+liczba2);
    }
    public static void wykonajOperacje(int x1, int x2){

```

```

        x1=x2*20;
    }
}
4 10

```

Które z wymienionych typów danych reprezentują wartości liczbowe:

float
double
long
int

Jaki będzie rezultat wykonania operacji: System.out.println(4|3);

7

Jaki tekst zostanie wyświetlony, jeśli poniższy program zostanie uruchomiony w następujący sposób:

```

java MojProgram niebieski rower
public class MojProgram{
    public static void main(String argv[]){
        System.out.println(argv[2]);
    }
}

```

Zgłoszony wyjątek: "java.lang.ArrayIndexOutOfBoundsException

Plik zadania.java zawiera następujący kod programu:

```
public class Zadania{
    public static void main(String[] args){
        int a=4; int b= 7;
        double c=3,2;
        a++;
        b+=5
        System.out.print("Wartość a wynosi: " +a);
    }
}
```

linie 1,3,4 są poprawne

obydwie linie 6 i 7 nie są poprawne

Które z poniższych linii kodu zostaną skompilowane bez informacji o błędzie:

int x = 017;

Jaka będzie wartość zmiennej x po wykonaniu poniższych instrukcji:

```
int x=5;
x +=2<3?4:5;
```

9

Jaki będzie rezultat działania poniższego programu:

```
final int c =15;
c+=4;
System.out.println(c);
błąd kompilacji
```

Zapis Rower r1= new Rower("Romet",12,'x'); to stworzenie obiektu i wywołanie konstruktora
Rower(String nazwa,int cena, char model)

Parametry typu int są przekazywane do metod poprzez
wartość

Które z wymienionych nierówności są prawdziwe?

0x01>01

8>07

Które z podanych zdań są prawdziwe:

operator / jest używany do wykonywania dzielenia jednej wartości przez drugą

symbol # nie może być użyty jako pierwszy znak w nazwie zmiennej

symbol \$ nie może być użyty jako pierwszy znak w nazwie zmiennej

Które z poniższych deklaracji parametru w statycznej metodzie main są prawidłowe:

String[] args

Które z następujących linii programu zostaną skompilowane bez informacji o błędzie:

int i=10;

Który z wymienionych specyfikatorów określa metodę statyczną:

static

Dany jest program:

```
1. public class Test{
2.
3. return((a*c)/(c+a));
```

4.}

5.}

które z poniższych linii mogą zostać wstawione w wierszu 2, aby proces kompilacji przebiegł prawidłowo?

```
double oblicz(float a, byte c){
```

```
float oblicz(float a, byte c){
```

Która z poniższych klas nie jest klasą opakowującą?

String

Instrukcja: Integer.parseInt("1532") zwróci nam:

wartość typu long

Metoda uczelnia() nic nie zwraca i nie ma parametrów. Który z poniższych nagłówków metody jest poprawny?

void uczelnia()

Nagłówek metody: public static long podajLiczbe(); oznacza:

metoda podajLiczbe może zostać wywołana nawet gdy nie istnieje żaden obiekt klasy

Jaki będzie rezultat wykonania poniższego programu:

```
public class Test{
    public static void main (String[] args){
        int[] a={1};
        Test t = new Test();
        t.dodaj(a);
        System.out.println(a[a.length-1]);
    }
    void dodaj(int[] i){
        i[i.length-1]++;
    }
}
```

błąd kompilacji

Dany jest program:

```
1. public class MojaKlasa{
2.
3. double podatek =383.2;
4. System.out.println("Wartość podatku: "+podatek);
5. }
6. }
public static void oblicz(){
```

Obiekty w języku Java tworzone są za pomocą?

klas

Które z poniższych instrukcji wyświetlą wartość równą -4,0:

```
System.out.println(Math.ceil(-4,7));
```

Jaki będzie rezultat wykonania programu:

```
public class Pytanie{
    private int i = pobierz();
    private int j =10;
    private int pobierz(){
        return j;
    }
    public static void main(String args[]){
        System.out.printl((new Pytanie()).i;
    }
}
```

```
}  
0
```

Klasa zawiera następującą deklarację metody: `double M(int a){}`
które z poniższych deklaracji metod przeciążonych są poprawne:
`double M(int x, int y){}`

Które z poniższych linii kodu mogą zostać umieszczone w miejsce komentarza, przy zachowaniu poprawności programu:

```
public class Test{  
    public void zapisz(int i, String s){}  
    //tu umieść jedną z linii kodu  
}  
public void zapisz(int i, String imie){}
```

Jaki jest domyślny menadżer rozkładu w komponencie JPanel ?
BorderLayout

Klasa File zawiera metodę `isDirectory()`, umożliwiającą sprawdzenie, czy wskazywany zasób jest folderem.
Jakiego typu wartość zwraca ta metoda?
boolean

Do jakiej kategorii należy wyjątek `FileNotFoundException`?
-wyjątek kontrolowany
-wyjątek wywodzący się z klasy `Exception`

Z jakiej klasy wywodzą się wyjątki kontrolowane?
exception

Jaki wyjątek zgłasza metoda `charAt()` klasy `String`?
`IndexOutOfBoundsException`

Które z wyrażeń logicznych umieszczonych w miejscu WYRAŻENIE spowodują, iż po wykonaniu poniższych instrukcji wartości zmiennej `x` będzie -3:

```
int x=5;  
while(WYRAŻENIE)  
    x-=4;
```

```
x>-3  
x>0
```

Jaka będzie wartość `x` po wykonaniu poniższego programu:

```
int x=5;  
for (int i=1; i<8; i+=4){  
    x+=1;  
}
```

Odpowiedź :11

Jaki rezultat zostanie wyświetlony na konsoli:

```
int i=1;  
switch (i--){  
    case 0;  
        System.out.println ("zero");  
    case 1;  
        System.out.println("jeden");  
    case 2;
```

```

        System.out.println("dwa");
    case 3;
        System.out.println("inna cyfra");

```

jeden dwa inna cyfra

Wskaż instrukcje iteracyjne:

for

do..while

Jak będzie wartość zmiennej 'j' po wykonaniu programu:

```

public class Abc
{
    public static void main(String args[])
    {
        int i=1, j=0;
        switch(i)
        {
            case 2: j+=6;
            case 4: j+=1;

        default: j+=2;
        case 0: j+=4;
        }
        System.out.println("j="+j);
    }
}

```

6

Która z poniższych instrukcji może umieszczona w miejscu WSTAW, aby kompilacja programu przebiegła bez błędów

```

public class Test{
    public static void main (String[] args){
        WSTAW
        switch(x){
            default:
                System.out.println("Brawo!!!Nic nie czaisz");
        }
    }
}

```

```

int x=2;
charx ='k';

```

Dany jest poniższy kod programu:

```

public class Przedmioty{
    public static void main(String[] args){
        String[]st = new String [4];
    for (int x =0; x<4; x++){
        st[x]=args[x];
    }
}

```

.....

Jaki będzie rezultat jego działania, gdy zostanie uruchomiony w następujący sposób: java Przedmiot Podstawy Informatyki

błąd wykonania: ArryIndexOutOfBundException

Które z poniższych stwierdzeń dotyczących instrukcji do..while są poprawne:

iteracja zostanie wykonana przynajmniej 1 raz

Jakie będzie rezultat działania poniższego programu(co zostanie wyświetlone):

```
int i =9;
switch (i){
    default:
        System.out.println("coś")
        case 0;
        System.out.printl("zero")
        case 1;
    .....
}
```

coś, zero

Wskaż instrukcje poprawne składniowo:

```
if(true) x=4;
if (true) (x=4;) else (x=5;)
if(true) x=4; else {x=5;}
```

Które z wymienionych instrukcji mogą zostać użyte wewnątrz instrukcji "switch":

break
continue

Wktórych z poniższych instrukcji sterujących możliwe jest użycie instrukcji "break":

switch
while
for

Które instrukcje są składniowo poprawne:

```
if(true) System.out.println("java");
if(5==5) System.out.println("java");
if (zmienna.equal("java")) System.out.println("java");
if (5!=8) System.out.println("java");
```

Jaka będzie wartość x po wykonaniu poniższego kodu programu:

```
int x=5;
do
x+=2;
while (x>4&& x<8);
```

Odpowiedz : 9

Wskaż wartości zmiennych "i" oraz "k" po wykonaniu poniższego kodu programu:

```
int i=2;
int k=6;
if(i>3)
i*=2;
k++;
i-=1;

k=7 i i=1
```

Lekcja 1:

1.

Jaką nazwę powinien mieć plik źródłowy zawierający taki kod:

```
public class DrugiProgram {  
  
    public static void main(String[] args) {  
        // wyświetlenie prostego napisu  
        System.out.println("Drugi program w Javie");  
    }  
}
```

odp: DrugiProgram.java

2.

Jakie rozszerzenie mają pliki źródłowe programów napisanych w Javie? (Podaj odpowiedź bez kropki)

odp: java

3.

Jakie rozszerzenie mają pliki skompilowane programów napisanych w Javie? (podaj odpowiedź bez kropki)

odp: class

4.

Jakie polecenie służy do skompilowania pliku źródłowego Test.java?

odp: javac Test.java

5.

Jakie polecenie służy do uruchomienia skompilowanego programu Test.class w Javie?

odp: java Test

6.

Jak wygląda nagłówek metody, od której rozpoczyna się wykonywanie programu w Javie?

odp: public static void main(String[] args)

7.

Które z poniższych komentarzy są poprawne w języku Java?

odp: // To jest komentarz.
/* To jest komentarz */

8.

Które z wymienionych zmiennych nie są zgodne z przyjętą konwencją tworzenia nazw?

odp: KolorOczu
Gimnastykaporanna

Lekcja 2:

1. Ile różnych wartości zawartych jest typie logicznym?

odp: 2

2. Które z wymienionych typów danych reprezentują wartości liczbowe

odp: long, double, int, float

3. Wskaż linie programu, które zostaną skompilowane bez informacji o błędzie:

odp: int i = -3

4. Czy poniższy fragment programu jest poprawny?

```
...  
String imie = "Joanna";  
String Imie = "Mateusz";  
...
```

odp: tak

5. Jaki będzie rezultat działania poniższych linii programu:

```
final int c = 15;
c += 4;
System.out.println(c);
```

odp: błąd kompilacji

6. Podaj wartość wyrażenia:

18/5
odp: 3

7. Podaj wartość wyrażenia:

1 + (2 - 3) * 4 % 5
odp: -3

8. Dana jest następująca klasa:

```
public class ZeroPrint{

    public static void main(String argv[]){
        int i =0;
        // tu umieść jedną z linii
    }
}
```

Które z poniższych linii umieszczone w miejscu komentarza wyświetlą wartość 0 ?

odp: System.out.println(i);
System.out.println(i++);
System.out.println(i--);

9. Które z wymienionych nierówności są prawdziwe:

odp: 8 > 07

10. Jaka będzie wartość zmiennej x po wykonaniu poniższych instrukcji

```
byte x = 3;
x += 1;
odp: 4
```

11. Jaka jest wartość wyrażenia:

(int)(2.3+7.2)
odp: 9

12. Program zawiera następującą deklarację zmiennej łańcuchowej:

```
String nazwaUczelni = "Uniwersytet Ekonomiczny w Krakowie";
```

Wprowadź wartość, która zostanie wyświetlona po wykonaniu poniższej instrukcji:

```
System.out.println(nazwaUczelni.length());
```

odp: 34

13. W łańcuchu znakowym:

```
"Uniwersytet Ekonomiczny w Krakowie, ul.Rakowicka 27, 31-510 Kraków"
```

naależy zastąpić każdy znak przecinka (,) znakiem średnika (;). Która z metod klasy String umożliwia wykonanie opisanej operacji. Podaj nazwę tej metody.

Wykaz metod klasy String dostępny jest w dokumentacji API Java.

odp: replace

14. Podaj nazwę metody klasy String, która przyjmuje wartość prawdy (true), gdy łańcuch znakowy nie zawiera żadnego znaku.

odp: isEmpty

15. Jaka będzie wartość zmiennej miasto po wykonaniu poniższych instrukcji:

```
String miasto = "Cesarsko-Królewskie ";  
miasto += "Miasto Kraków";  
miasto = miasto.substring(9,30);  
odp: Królewskie Miasto Kra
```

16. Program zawiera następującą deklarację tablicy:

```
String[] nazwyKolorow= {"zielony","czarny","fioletowy","niebieski","czerwony"}
```

Jaka jest wartość elementu tablicy nazwyKolorow[3] ?

odp: niebieski

17. Kod programu zawiera następującą deklarację tablicy wielowymiarowej:

```
int[][] liczbaNaturalna = {{43,2,1,76,4},{3,56,1,55,63},{54,6,8,12,44}}
```

Jaka będzie wartość elementu tablicy liczbaNaturalna[2][1] ?

odp: 6

Lekcja 3:

1. Jaki będzie rezultat działania poniższego kodu programu (jaka informacja zostanie wyświetlona na monitorze):

```
int i = 0;  
if (i++ > 0) {  
  
    System.out.println("X");  
} else {  
    System.out.println("Y");  
}  
}
```

odp: Y

2. Jaki będzie rezultat działania poniższego kodu programu?

```
int i = 2;  
switch (i) {  
  
    case 1: System.out.println("jeden"); break;  
    case 2: System.out.println("dwa");  
    case 3: System.out.println("trzy");  
    default: System.out.println("inna liczba"); }  
}
```

odp: dwa trzy inna liczba

3. Jaki będzie rezultat działania poniższego programu (co zostanie wyświetlone):

```
int i=9;  
switch (i) {  
  
    default:  
        System.out.println("coś");  
    case 0:  
        System.out.println("zero");  
}
```

```
break;
case 1:
System.out.println("jeden");
case 2:
System.out.println("dwa");
}
```

odp: coś zero

4. Ile linii tekstu zostanie wyświetlonych po wykonaniu poniższego kodu programu:

```
int x = 5;
while(x-- > 3) {

System.out.println("Kolejna wyświetlona linia");
}
```

odp: 2

5. Które wyrażenie logiczne umieszczone w miejscu WYRAŻENIE spowoduje, iż po wykonaniu poniższych instrukcji wartością zmiennej x będzie wynosić -3:

```
int x = 5;
while(WYRAŻENIE)
```

```
x -= 4;
```

odp: $x > -3$

6. Jaki będzie rezultat działania poniższego kodu programu:

```
int x = 5;
for (int i=1; i<8; i+=4) {

x += i;
}
System.out.println(x);
```

odp: 11

7. Jaki rezultat zostanie wyświetlony na konsoli po wykonaniu poniższego programu:

```
public class Liczby {

public static void main(String [] args) {
int j = 2, k = 3, z = 10;
for (; j<6; j++) {
k = (++k + z);
System.out.println(k);
}
}
}
```

odp: 14 25 36 47

8. W której z poniższych instrukcji sterujących nie jest możliwe użycie instrukcji "break" ?

if
switch
while
for

Wpisz nazwę instrukcji.

odp: if

Lekcja 4:

1. Które deklaracje zmiennej utworzą referencje?

odp: Scanner wejscie; String napis;

2. Który z poniższych zapisów utworzy obiekt klasy Monitor?

odp: Monitor lcd = new Monitor();

3. Podaj określenie typu dla metod, które nie zwracają żadnej wartości

odp: void

4. Jaka instrukcja kończy działanie metody?

odp: return

5. Czy metoda liczbaStudentow():

```
int liczba = Student.liczbaStudentow();
```

jest metodą instancyjną?

odp: Nie

6. Mamy daną metodę:

```
int oblicz(int a, int b) {  
    a++;  
    b--;  
    return (a+b);  
}
```

Jaka wartość zostanie przypisana do zmiennej wynik?

```
int wynik = oblicz(10, 12);
```

odp: 22

7. Czy modyfikator private pozwala na użycie metody poza klasą, w której ta metoda została utworzona?

odp: Nie

8. Poniżej podane zostały przykładowe nagłówki metody przeciążonej włączZawor(). Czy zapis jest poprawny?

```
boolean włączZawor(String nazwa, int typ)  
int włączZawor(String identyfikator, int numer)
```

odp: Nie

9. Jaką instrukcję należy umieścić na początku programu, aby móc korzystać z klasy Scanner?

odp : import java.util.Scanner;

10. Jakiego typu wartości zwraca metoda ceil(double) z klasy Math?

odp: double

11. Które z poniższych klas należą do kategorii klas opakowujących?

odp: java.lang.Boolean, java.lang.Long

12. Jakie są różnice pomiędzy typem double, a Double? Który z wymienionych typów posiada większe możliwości?

odp: Double

Lekcja 5:

1. Czy niezainicjalizowane pola posiadają wartość początkową? Jeśli tak, to które wartości dla typów wymienionych po myślniku są poprawne?

odp: false - boolean, null - String, null - Integer;

2. Które z poniższych metod nie są konstruktorami w klasie Test?

odp: public test() {}, public int Test() {return 0;}, public void Test() {},

3. Czy każda klasa posiada konstruktor?

odp: Tak

4. Ile może występować konstruktorów w danej klasie?

odp: nie ma ograniczeń

5. Za pomocą jakiego operatora tworzone są nowe obiekty?

odp: new

6. Czy zawsze podczas tworzenia obiektu wywoływany jest konstruktor?

odp: Tak

7. Poniższy kod programu zawiera definicję klasy wraz z jej składowymi.

```
class PenDrive {  
    private String nazwa;  
    private int pojemnosc;  
  
    public PenDrive(String n, int p){  
        nazwa = n;  
        pojemnosc = p;  
    }  
}
```

Czy możliwe jest utworzenie nowego obiektu klasy PenDrive w następujący sposób?

```
PenDrive mojPendrive = new PenDrive();
```

odp: Nie

8. Jaki nagłówek powinna mieć metoda dostępowa do pola nazwa?

```
class Laptop {  
    private String nazwa;  
}
```

odp: public String pobierzNazwe();

9. Która z poniższych instrukcji umieszczonych zamiast komentarza przypisanie w klasie PenDrive jest poprawna? Chcemy przypisać do pola wartość przekazaną przez użytkownika podczas tworzenia obiektu.

```
class PenDrive {
```

```
private String nazwa;
```

```
public PenDrive(String nazwa){  
    // przypisanie  
}  
}
```

odp: this.nazwa = nazwa;

10. Jaka instrukcja wywoła wewnątrz klasy konstruktor bezparametrowy?

odp: this();

11. Aby w pełni wykorzystać enkapsulację danych pola klasy powinny mieć jaki specyfikator dostępu?

odp: private

Lekcja 6:

1. Określ związek pomiędzy obiektami: samochód, silnik

odp: kompozycja

2. Jaki będzie rezultat działania poniższego programu:

```
class B {  
  
    static int x = 10;  
}  
  
public class A extends B {  
    public static void main(String [] args) {  
        B klasaB = new B();  
        A klasaA = new A();  
        klasaB.x = 20;  
        System.out.println("wynik " + klasaA.x + " " + klasaB.x);  
    }  
}
```

odp: program wyświetli 20 20

3. Czy język programowania Java umożliwia dziedziczenie na podstawie wielu klas (tzw. dziedziczenie wielobazowe) ?

odp: nie

4. W jakim pakiecie została zdefiniowana klasa String ?

odp: java.lang

5. Jaka instrukcja umożliwia wykorzystanie w tworzonym programie klas z wybranego pakietu?

odp: import

6. Tworzony program zawiera następującą instrukcję:

```
x = Math.round(25.4);
```

Czy powyższa instrukcja jest poprawna, jeśli wcześniej nie zostanie wykonana instrukcja:

```
import java.lang
```

odp: tak

7. Podaj nazwę operatora umożliwiającego sprawdzenie, czy obiekt należy do wskazanej klasy.

odp: instanceof

Lekcja 7:

1. Jakie słowo kluczowe informuje, że klasa implementuje jakiś interfejs?

odp: implements

2. Czy poniższy kod jest poprawny?

```
interface Testowy {  
    public int dodaj(int a, int b);  
    public String toString();  
}
```

```
class Test implements Testowy {  
    public int dodaj(int a, int b) {  
        return a+b;  
    }  
}
```

odp: Tak

3. Czy poprawny jest poniższy kod?

```
public interface Tester {  
}
```

odp: Tak

4. Czy jest możliwe tworzenie interfejsów pochodnych?

odp: Tak

5. Które z poniższych deklaracji są błędne (KlasaA, KlasaB, KlasaC oznaczają nazwy klas, natomiast InterfejsA, InterfejsB, InterfejsC to nazwy interfejsów)?

```
odp: class KlasaC extends KlasaA, KlasaB  
interface InterfejsC extends InterfaceA, InterfaceB
```

6. Poniższy kod programu zawiera definicję klasy abstrakcyjnej wraz z jej składowymi. Czy jest ona poprawna?

```
public abstract class Zwierze {  
    private String nazwa;  
  
    public abstract void je() {  
        System.out.println ("Aktualnie je.");  
    }  
    public abstract void idzie();  
    public abstract void pije();  
}
```

odp: Nie

7. Czy możliwe jest utworzenie obiektów na bazie klasy abstrakcyjnej?

odp: Nie

8. Czy nazwa klasy, która dziedziczy z klasy abstrakcyjnej i nie dostarcza implementacji metod zawartych w nadklasie musi zostać poprzedzona identyfikatorem abstract?

odp: Tak

9. Czy poniższy kod programu jest poprawny?.

```
final class A {  
    private String nazwa;  
  
    public String pobierzNazwe() {  
        return this.nazwa;  
    }  
}
```

```
class B extends A {  
    private int liczba;
```

```
    public int pobierzLiczbe() {  
        return liczba;  
    }  
}
```

odp: Nie

10. Które z poniższych klas są klasami finalnymi? Skorzystaj z dokumentacji.

odp: String

Lekcja 8:

1. Jaki wyjątek zgłasza metoda charAt() klasy String? Sprawdź w dokumentacji Javy.

odp: IndexOutOfBoundsException

2. Z jakiej klasy wywodzą się wyjątki kontrolowane?

odp: Exception

3. Który rodzaj wyjątku zostanie zgłoszony w przypadku próby dostępu do nieistniejącego pliku na dysku lokalnym komputera?

odp: wyjątek kontrolowany

4. Podaj nazwę klasy umożliwiającej odczyt zawartości pliku tekstowego.

odp: FileReader

5. Podaj nazwę pakietu, który zawiera podstawowy zbiór klas przeznaczonych do zapisu i odczytu danych. (sprawdź w API Java, w jakim pakiecie znajdują się wymieniane wcześniej klasy).

odp: java.io

6. W skład klasy File wchodzi metoda umożliwiająca utworzenie nowego folderu. Podaj jej nazwę.

odp: mkdir

7. Klasa File zawiera metodę isDirectory(), umożliwiającą sprawdzenie, czy wskazywany zasób jest folderem. Jakiego typu wartość zwraca ta metoda?

odp: boolean

ekcja 9:

Aplety:

1. Podaj nazwę atrybutu elementu APPLET określającego plik zawierający kod programu w języku java.

odp: code

2. Podaj nazwę metody, która wywoływana jest po załadowaniu apletu.

odp: start

3. Podaj nazwę metody umożliwiającej wyświetlenie łańcucha znaków.

odp: drawString

4. Czy możliwe jest ustanowienie połączenia apletu z dowolnym komputerem przyłączonym do sieci Internet?

(odpowiedź na powyższe pytanie możesz znaleźć pod adresem <http://java.sun.com/docs/books/tutorial/deployment/applet/QandE/questions.html>)

odp: nie

Graficzny interfejs użytkownika:

1. Jaka klasa zazwyczaj jest wykorzystywana do utworzenia głównego okna aplikacji?

odp: JFrame

2. Czy główne okno aplikacji domyślnie jest widoczne?

odp: Nie

3. W skład jakiego pakietu wchodzi klasa JFrame?

odp: javax.swing

4. Która z klas będących kontenerami umożliwia tworzenia pasków przesuwanych?

odp: JScrollPane

5. Która z klas będących komponentami umożliwia tworzenie przycisków?

odp: JButton

6. Który z menadżerów rozkładu umożliwia rozmieszczenie elementów według kierunków geograficznych?

odp: BorderLayout

7. Ciało jakiej metody musi mieć klasa implementująca interfejs ActionListener?

odp: public void actionPerformed(ActionEvent e)

8. Jaka metoda dodaje do komponentu obsługę zdarzeń związanych z interfejsem MouseListener

odp: addMouseListener()

9. Która metoda musi zostać przesłonięta, aby było możliwe umieszczenie w aplikacji własnych elementów graficznych?

odp: paintComponent()