ROZDZIAŁ I - ŚRODOWISKO PROGRAMISTYCZNE

1. Wymień główne zalety języka Java.

Pełna obiektowość, niezależność od architektury (systemu operacyjnego, procesora), funkcjonalność (np. automatyczne zarządzanie pamięcią, obsługa wyjątków, wielowątkowość, tworzenie aplikacji z wykorzystaniem GUI), niezawodność i bezpieczeństwo tworzonych aplikacji

2. Scharakteryzuj sposób tworzenia i uruchamiania programów tworzonych w języku Java.

Tworzony jest kod źródłowy programu w dowolnym edytorze znakowym bądź IDE. Utworzony plik źródłowy ma rozszerzenie .java. Następnie kompilator zmienia kod źródłowy na kod wykonywany przez komputer, w wyniku kompilacji powstaje program w postaci pliku wykonywalnego z rozszerzeniem .class. Uruchomienie programu sprowadza się do wydania komendy java wraz z nazwą programu (np. java Program1). Program zostanie uruchomiony za pomocą Wirtualnej Maszyny Javy.

3. Wskaż różnice pomiędzy JDK i JRE.

JDK - Java Development Kit - zestaw narzędzi pozwalających na utworzenie programu, w skład których wchodzi kompilator, interpreter kodu bajtowego, przeglądarka umożliwiająca uruchaminie apletów, czy dodatkowe programy narzędziowe.

JRE - Java Runtime Environment - Wirtualna Maszyna Javy wraz ze zbiorem standardowych klas; JRE jest niezbędne do uruchomienia jakiegokolwiek programu utworzonego w języku Java

4. Określ różnice występujące pomiędzy zwykłym edytorem znakowym, a IDE.

IDE - zintegrowane środowisko znakowe służące do tworzenia, modyfikowania oraz testowania programu. Zasadnicza różnica pomiędzy zwykłym edytorem znakowym, a IDE jest automatyczne podświetlanie składni, dodawanie propozycji podczas pisania kodu oraz autokorekta.

5. Jakie rozszerzenie nazwy posiadają pliki źródłowe, a jakie pliki skompilowane?

Źródłowe .java / Skompilowane .class

6. Czym jest i jaką funkcję pełni Wirtualna Maszyna Javy?

JVM - Java Virtual Machine - Wirtualna maszyna Javy stanowiąca "wirtualny komputer", na którym uruchamiane są programy napisane w języku Java

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

KolorOczy, umyjTwarz, Gimnastykaporanna, CechaCharakteruKobiety

8. Z jakich elementów składa się kod źródłowy?

Opcjonalny komentarz początkowy, instrukcje deklaracji pakietu bądź/i instrukcje importu użytych klas,

deklaracje interfejsu lub klasy.

9. Czy tworząc programy w Javie należy zwracać uwagę na wielkość znaków?

Tak, składnia języka Java rozróżnia wielkość stosowanych znaków

10. Zapoznaj się z dokumentacją do Javy. Sprawdź opis klasy System oraz metod służących do wyświetlania informacji na standardowym wyjściu (np. print(), println() itd.). Czym różnią się te metody?

Metoda print() różni się od metody println() tym, że ta druga metoda po wyświetleniu danego ciągu znaków daje jeszcze nową linię.

ROZDZIAŁ II - TYPY DANYCH, ZMIENNE I OPERATORY

1. Wskaż różnice, jakie występują pomiędzy zmiennymi, stałymi i literałami.

Zmienne określają obszar pamięci, przechowujący dane, które można modyfikować w trakcie działania programu. Każda zmienna posiada nazwę oraz typ, który okresla zakres dopuszczalnych wartości oraz rodzal możliwych do przeprowadzenia operacji.

Stała, podobnie jak zmienna określa obszar pamięci operacyjnej, służący do przechowywanie danych. Również posiada nazwę oraz typ. W przypadku stałej jednak nie ma możliwości zmiany wcześniej nadanej wartości w trakcie działania programu. Deklaracja stałej poprzedzona jest słowem kluczowym final [np. final double KURS_EURO = 2.48] .

Natomiast literał to ciąg znaków reprezentujący daną wartość o danym typie, wprowadzana przez programistę do kodu, np. do zmiennej. Warto zwrócić uwagę na stosowanie symboliki w celu precyzyjnego określenia typu literału [np. 252L]

2. Jakie funkcje w programie pełni deklaracja zmiennej?

Deklaracja zmiennej pozwala na zarezerwowanie obszaru w pamięci operacyjnej w celu przyszłego przechowywania danych wartości, przypisanych do zmiennej.

3. Na którym z dostępnych typów prymitywnych nie jest możliwe przeprowadzenie operacji rzutowania?

boolean

4. Uszereguj operatory według ich priorytetu:

+ % -- <

Odp. -- %+<

5. Jaką funkcjonalność realizują operatory ++ ?: =/ % ? Podaj przykłady wyrażeń z ich użyciem.

++ operator inkrementacji

Zwiększa wartość danej zmiennej o 1

[np. i++]

?: operator trójkowy (ternary operator), tzw. skrócony if

Pozwala na przypisanie zmiennej wartości w zależności od warunku, prosty krótki if.

minVal = (a < b) ? a : b;

Jeżeli true, to minVal = a.

/= operator przypisania, przypisuje nową wartość zmiennej, na podstawie wcześniejszej wartości wykonuje określoną operację arytmetyczną

wartoscA = 12

wartoscA /= 2;

w efekcie wartoscA będzie przechowywać wartość 6.

% operator modulo

Pozwala na otrzymanie reszty z dzielenia

if (a % b) == 0

System.out.print("TAK");

Jeżeli reszta z dzielenia a przez b wyniesie 0, to wykona się instrukcja print.

6. Określ wartość wyrażenia:

2 > 3 || 4 < 5 ? 6 << 7 : 8 / 9

true

7. Wymień dopuszczalne wartości (minimalne i maksymalne) dla każdego z typów prymitywnych.

boolean yes/no

byte (8 bit) od **-128 do 127**

short (16 bit) od **-32 768 do 32 767**

int (32 bit) od **-2^31 do 2^31-1**

long (64 bit) od **-2^63 do 2^63-1**

8. Podaj typ wartości wyrażenia: 12/7

int

9. Które z wymienionych instrukcji nie spowodują powstania błędu kompilacji?

float f = 1.01; int i = 1/3; double d = 777d;

int i = 1/3 oraz double d = 777d;

Przy dekleracji zmiennej typu float należy na końcu dodać literę f

np. float f = 1.01f;

10. Jaka jest wartość poniższej nierówności? Uzasadnij odpowiedź.

(243) 0xF3 > 0574 (380) [szesnastkowy] > [ósemkowy]

druga pozycja od prawej strony

16^1 * 15 + 16 ^ 0 * 3 = 243

w efekcie wychodzi FALSE

11. Określ kody w standardzie unicode dla wymienionych znaków: B 9 Ć # ś

chara zamieniasz na inta i ci pokazuje kod w standardzie unicode, jak nie poleci samo, to pierdolnij rzutowaniem, np. char dupa = (int) 'B'

12. Jakie oznaczenie (indeks) posiada pierwszy element tablicy?

indeks zerowy, tablica[0]

13. Zapoznaj się z dokumentacją klasy java.lang.String odszukaj metodę umożliwiającą łączenie ciągu znaków. Podaj przykład instrukcji łączącej 2 literały zawierające Twoje imię i nazwisko.

metoda concat()

np. String ja = ("Rafał".concat("Gajda"));

  System.out.println(ja);

14. Jaką informację zwróci wyrażenie tablica.length w przypadki tablicy wielowymiarowej?

Zwróci ilość wierszy tej tablicy

15. Podaj przykład użycia metody porządkującej elementy tablicy typu int.

wcześniej trzeba *import java.util.Arrays;*

Arrays.sort(table);

INSTRUKCJE STERUJĄCE

1. Wymień nazwy wszystkich instrukcji sterujących:

for, switch, do while, while, if, foreach, ? :

2. Podaj definicje algorytmu. Jakie wyrozniamy sposoby jego zapisu?

Algorytm - skonczony uporzadkowany ciag czynnosci niezbednych do realizacji zadania

Sposoby zapisu: schemat blokowy, slowny, lista krokow, pseudokod

3. Przedstaw algorytm wyznaczania silni wartosci N przy uzyciu pseudokodu

4.

5. Czym jest blok instrukcji i jaki jest sposób jego notacji w języku programowania Java?

Blok instrukcji jest oddzielony klamrami, definiuje zasieg zmiennych, jest to wydzielona czesc kodu zrodlowego,

6. Jaki jest zasieg widocznosci zmiennych zadeklarowanych wewnatrz bloku instrukcji?

Zmienne lokalne sa deklarowane wewnatrz bloku, ich zasieg obejmuje czesc programu od miejsca deklaracji do konca tego bloku

Zmienne globalne sa zadeklarowane poza wszystkimi funkcjami. Zyja od poczatku do konca programu.

7. Operatory logiczne w ifie

alternatywa, koniunkcja, negacja,

8. Jaką rolę w instrukcji warunkowej pełni słowo kluczowe else?

Jeżeli warunek if oraz if else nie zostaną spełnione to zostanie wykonany kod po słowie kluczowym else

9. Jaki będzie rezultat działania poniższego kodu programu?

```
int i = 0;
if (i++ > 0) {
System.out.println(i);
```

}

ODP. NIC SIĘ NIE STANIE

10.
Które z wymienionych typów danych może przyjmować wyrażenie instrukcji switch?
Character, Byte, Short, and Integer

12. **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. wypisze na ekranie:

dwa

trzy

inna liczba

13. Wymien i scharekteryzuj skladowe instrukcji for

(zmienna i jej typ, warunek, czynnosc)

{

instrukcje

}

14. Podaj skladnie zapisu petli okreslonej, zadaniem ktorej bedzie wyswietlanie na konsoli wszystkich parzystych liczb naturalnych z przedzialu 50, 100;

for (int i=50; i<=100; i+=2) {

System.out.println(i);

}

15.
Jaki będzie rezultat działania poniższego programu?

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

ODP. wypisze na ekranie 11

16. Omow zasade funkcjonowania odmiany instrukcji for przeznaczonej dla operacji na elementach tablicy

FOREACH - rozszerzona instrukcja for - wykorzystywana przede wszystkim do wypisywania zawartości tablicy

np.

```
int tab[] = {1, 2, 3, 4 , 5};

for (int cokolwiek : tab)

{

System.out.println(cokolwiek);    }
```

17. Wymien zasadnicze roznice pomiedzy petla okreslona for, a petla nieokreslona while?

W petli for z gory znana jest liczba iteracji, natomiast w petli while, liczba iteracji nie jest znana.

18. Jaka funkcje pelnia instrukcje break oraz continue uzyte w instrukcji sterujacej

continue przechodzi do kolejnej iteracji, pomijając tą w której jest ona użyta

break przerywa dzialanie instrukcji sterujacej

19.**Czym różnią się instrukcje iteracyjne while oraz do..while ? Wskaż zasadnicze różnice.**

Do while wykona się zawsze przynajmniej raz

20. Wymien instrukcje sterujace, w ktorych mozliwe jest uzycie instrukcji break

switch, for, while, do while

4.OBIEKTY METODY

1. Wskaz roznice jakie wystepuja pomiedzy zmienna typu prostego, a zmienna typu klasowego

2. Podaj sposob tworzenia obiektow w Javie

new nazwa_klasy(argumenty inicjalizacji obiektu);

np: new Scanner(System.in);

3. **Wyjaśnij, czym jest referencja do obiektu.**

Referencja to wartosc, ktora oznacza lokalizacje (adres) obiektu w pamieci.

4. W jaki sposob utworzone obiekty usuwane sa z pamieci operacyjnej? Kiedy to nastepuje? Jezeli nie sa uzywane, to sa usuwane automatycznie. Zajmuje sie tym specjalny proces. Java okresowo sprawdza referencje do obiektow, te ktora juz nie sa wykorzystywane, sa usuwane.

5. **Podaj okreslenie typu dla metod, ktore nie zwracaja zadnej wartosci**

void

6**. Wskaz roznice pomiedzy metoda instancyjna a klasowa**

metoda klasowa wystepuje jako jedna kompia w programie, a instancyjna jest instancja w kazdym obiekcie ktory taka metode implementuje

7. W jaki sposob kompilator rozpoznaje metody przeciazane

ta sama nazwa, o roznych typach oraz liczbie parametrów

8. Jaka instrukcja konczy dzialanie metody;

return

9. Czym roznia sie poszczegolne modyfikatory dostepu do metod?

Roznia sie zasiegiem

10. Czy ponizszy kod programu to wywolanie metody instancyjnej, czy klasowej.

Metody instancyjne moga byc uruchomione jedynie na rzecz utworzonych obiektow

Instancyjna - tworzysz obiekty

Klasowa - nie tworzysz obiektow

Klasowa - Klasa.metoda(a, b)

11. **W jaki sposób kompilator jest informowany o klasach wykorzystywanych w programie?**

import...

12. Czym różni się double od Double

Double jest typem złożonym oraz posiada więcej możliwości a double to typ prosty

13. Ktore z ponizszy klas naleza do kategorii klas opak.
java.lang.Int, java.lang.Boolean,
java.lang.Long, java.lang.Char

wszystko oprocz voida i stringa z wypisanych

wszystkie typy prymitywne maja klasy opakowujace

14. Poniżej podane zostały przykładowe nagłówki metody przeciążonej wlaczZawor(). Wskaż ewentualne błędy w podanym zapisie.
boolean wlaczZawor(String nazwa, int typ)
int wlaczZawor(String identyfikator, int numer)

odp. błąd polega na tym iż argumenty są tych samych typów a nie powinny być

15. Sprawdź, jakie jest przeznaczenie klasy java.util.Locale. Podaj nazwę pakietu, w skład którego wchodzi podana klasa.

nazwa pakietu java.util, klasa ta wykorzystywana jest do rozwiązywania zagadnień lokalizacyjnych

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

19. Dlaczego metody klasy Math zostały zdefiniowane jako klasowe? Uzasadnij odpowiedź.
Metody te sa często używane, przez ich zdefiniowanie jako klasowe mamy do nich łatwiejszy dostęp. Ponadto jest ich dużo i w klasie łatwiej je uporządkować

ROZDZIAŁ V - Klasy i Obiekty

1. Czym jest klasa oraz jakie posiada właściwości?

Klasa określana jest jako zbiór stanów oraz zachowań opisujących obiekty należące do tej samej kategorii. Jest to pewnego rodzaju wzorzec, na podstawie którego tworzone są unikalne wersje obiektu. W skład każdej klasy wchodzą:
pola (zmienne), zawierające informacje opisujące właściwości (stany) obiektów oraz
metody (podprogramy), modelujące zachowania obiektów.

2. Do czego wykorzystywane są diagramy UML?

Diagramy UML wykorzystywane są do zilustrowania pojęć związanych z programowaniem obiektowym,

reprezentują one związki pomiędzy klasami oraz metody występujące w tych klasach.

3. Czym jest obiekt oraz w jaki sposób jest tworzony?

Obiekt to naturalna reprezentacja klasy. Przechowuje informacje o zmiennych i pozwala się do nich odwoływać, a także za jego pomocą możemy wywoływać metody.

Utworzenie nowego obiektu - czyli instancji klasy dokonujemy za pomocą słowa kluczowego new np. **System s = new System();**

4. Jaka rolę w programowaniu obiektowym spełnia operator new?

Operator new rezerwuje pamięć dla nowego obiektu oraz wywołuje odpowiedni konstruktor. Operator new zwraca także referencję do nowo utworzonego obiektu.

5. Wskaż różnice pomiędzy metodą, a konstruktorem.

Konstruktor jest wywoływany **ZAWSZE** podczas tworzenia obiektu - metoda niekoniecznie

Konstruktor nazywa sie tak samo jak klasa i nic nie zwraca (nawet void)

6. Jaki proces zachodzi podczas wykonywania poniższej instrukcji?
Laptop laptopFirmowy = new Laptop("Lenovo");

Utworzenie obiektu typu Laptop poprzez wywołanie konstruktora klasy Laptop przyjmującego jeden parametr typu String. Obiekt ten przypisywany jest do zmiennej laptopFirmowy typu Laptop.

7. Wymień i scharakteryzuj składowe klasy.

Pola klasy, metody klasy, konstruktor

8. Czy niezainicjalizowane pola posiadają wartość początkową? Jeśli tak, to jaką?

W przypadku braku jawnej inicjalizacji pól klasy, przyjmują one wartości domyślne **(pola numeryczne wartości zerowe**, **pola logiczne wartość false**, natomiast **pola klasowe wartość null).**

9. Czy każda klasa musi posiadać konstruktor?

tak, każda klasa posiada co najmniej jeden konstruktor, jeżeli nie uwzględnimy go w kodzie, to wirtualna maszyna Javy "sama" automatycznie utworzy konstruktor domyślny

10. Poniższy kod programu zawiera definicję klasy wraz z jej składowymi. Wskaż ewentualne błędy.

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

-konstruktor nie może nic zwracać

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

PenDrive mojPendrive = new PenDrive();

konstruktor ma być dwu parametrowy (nazwa, pojemność)

12. Jaką funkcję pełni słowo kluczowe this?

Słowo this oznacza referencję do bieżącego obiektu. Ułatwia to dostęp do jego składowych, jak również umożliwia wywołanie odpowiednich konstruktorów. Często **this jest wykorzystywane, gdy nazwa parametru metody jest identyczna z nazwą pola, umożliwiając rozróżnienie tych identyfikatorów**. Użycie słowa kluczowego this wraz z ewentualną listą parametrów umożliwia wywołanie konstruktora z ciała innego konstruktora

13. Jakie jest główne zastosowanie enkapsulacji?

Enkapsulacja inaczej zwana hermetyzacją polega na ukrywaniu metod i atrybutów dla klas zewnętrznych. Dostęp do nich możliwy jest tylko z wewnątrz klasy, do której należą, z klas zaprzyjaźnionych lub z klas dziedziczących.

14. Poniżej podany został kod programu wykorzystujący klasę PenDrive. Czy odwołanie do pola pojemnosc jest poprawne?
PenDrive mojPendrive = new PenDrive("Toshiba", 4096);
mojPendrive.pojemnosc = 8192;

tak

15. Wskaż różnice pomiędzy statycznymi (klasowymi) i niestatycznymi (instancyjnymi) polami klasy

Pola statyczne (oznaczone w kodzie modyfikatorem static), w przeciwieństwie do pól niestatycznych**,** są wspólne dla wszystkich obiektów danej klasy**. Dostęp do nich jest możliwy bez konieczności tworzenia obiektu klasy**. Dostęp do pól klasy jest możliwy zgodnie z rodzajem użytego w deklaracji modyfikatora dostępu.

16. Dlaczego z metody statycznej nie jest możliwe odwołanie do składowych instancyjnych?

Metoda statyczna nie jest wywoływana na rzecz konkretnego obiektu, dlatego nie może odwoływać się do składowych niestatecznych.

17. Podaj sytuacje, w których wykorzystywane są metody prywatne.

Metody prywatne są wykorzystywane w programie gdy nie chcemy aby użytkownik ingerował w nasze zmienne.

18. Wskaż przykład zastosowania metod dostępowych oraz modyfikujacych dla klasy PenDrive.

Metody dostępowe są używane do odczytu wartości pól (get), a modyfikujące do zmiany pól (set)

19. Jaka informacja zostanie wyświetlona na konsoli po wykonaniu poniższego kodu programu?
```
Student[] studenci = new Student[10];
System.out.println(studenci[4]);
```

null

20. Wskaż błąd w poniższym kodzie programu, wykorzystującym klasę Telefon.

```
int n = 10;
Telefon[] telefony = new Telefon[n];
for (int i = 0; i < n; i++) {
telefony[i].zadzwon("112");
}
```

brakuje konstruktora

1. Czym jest kompozycja w programowaniu obiektowym?

- **Kompozycja –** wyraża relacje „składa się z” lub „posiada” . w skład tworzonej klasy może wchodzić dowolna liczba obiektów tworzona na podstawie istniejących klas

2. Wskaż różnice pomiędzy pojęciami klasa bazowa, a podklasa.

- Różnica pomiędzy pojęciami klasa bazowa (ta z której dokonano dziedziczenia) a podklasa (jest to też dziecko o_O; ta która dziedziczy).

3. Jakie składowe klasy bazowej dziedziczy podklasa?

- Podklasa dziedziczy składowe klasy bazowej - wszelkie cechy i zachowania(pola i metody), dodając bądź modyfikując je by były bardziej wyspecjalizowane, konstruktory **nie** są dziedziczone.

4. Czy każda klasa w Javie dziedziczy z innej klasy? Czy możliwe jest dziedziczenie z wielu klas?

- Klasy które nie posiadają zadeklarowanego dziedziczenia, domyślnie dziedziczą z klasy Object, nie jest możliwe dziedziczenie z wielu klas.

5. Wyjaśnij pojęcie polimorfizmu.

- **Polimorfizm –**Możliwość traktowania obiektów różnych podtypów pewnego wspólnego typu, w ten sam sposobik . inaczej wielopostaciowość poleg na tym, że do ogólnej wspólnej referencji można przypisać obiekty różnych typów, które po typie tej referencji dziedziczą.

6. Jakie zadania spełnia klasa `Object`? Jakimi metodami dysponuje?

- **Klasa Object** jest główną klasą z której pośrednio lub bezpośrednio dziedziczą wszystkie klasy. Zawiera szereg metod zawierających operacje na obiektach. Najważniejsze z nich to:
  clone() – tworzy kopię obiektu,
  toString() – zwraca reprezentację obiektu w formie łańcucha tekstowego,
  equals(Object) – porównuje dwa obiekty,
  getClass() – zwraca nazwę klasy na podstawie, której powstał obiekt.

7. Podaj przykłady użycia metody `toString()` z klasy `Object`?

- toString() – zwraca reprezentację obiektu w formie łańcucha tekstowego,

8. Jaką funkcję pełni operator `instanceof`

- Operator instanceof służy do sprawdzenia czy dany obiekt należy do wskazanej klasy. Przyjmuje on wartość true lub false.

9. Wskaż różnice pomiędzy słowami kluczowymi `this` i `super`.

- this znajdujaca sie w konstruktorze, wywołuje inny konstruktor z tej samej klasy
  super - wywołuje konstruktor z klasy bazowej

10.Określ zalety używania pakietów.

- pozwala na szybsze odnalezienie klas i interfejsów, wpływa korzystnie na przejrzystość pisanych aplisracji, jasne określenie związków między klasami, unikanie konfliktów nazw, możliwość z skorzystania dodatkowych modyfikatorów dostępu

11. W jaki sposób możliwe jest użycie klas znajdujących się we wskazanym pakiecie?

- umożliwia kluczowe słowo *import* a potem nazwa pakietu oraz nazwa klassy albo * na wszyyyyystkie klasy należące do pakietu

12. Czy możliwe jest przesłonięcie metody `equals(Object)`? Jeśli tak, to jakie zadanie będzie spełniać ta metoda?

- Jest możliwe przesłonięcie metody equals, będzie porównywa obiekt który ją wywołuje oraz obiekt podany w parametrze metody

Rozdział 7.

1. Określ cel i zastosowanie interfejsów.

wymusza implementacje metod, które są w nim zawarte prze klasy które go implementują.

2. Czy specyfikacja języka Java narzuca ograniczenia na liczbę interfejsów, jakie może implementować tworzona klasa?

- Można implementować wiele interfejsów

3. Czy możliwe jest tworzenie interfejsów pochodnych?

- Dopuszczalne jest dziedziczenie interfejsów

4. W jakim przypadku klasa implementująca interfejs nie musi tworzyć wszystkich metod występujących w tym interfejsie?

- Nie musi implementować metody jak już ją kurde posiada

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

- Class nie może extends kilku klas (dziedziczenie z 1)
  interfejs nie może implementować interfejsu (ale dziedziczyć może i to z wielu np. interface InterfejsC extends InterfaceA, InterfaceB)

7. Czy poprawny jest poniższy kod?

```
public interface Tester { }
```

- Może być interfejs bez ciała

8. Podaj definicję klasy abstrakcyjnej.

- Klasa abstrakcyjne – wykorzystywane wprocesie modelowania zawierając jedynie ogólne cechy, deklaracja odbywa się zza pomocą słowa kluczowego abstrakt, nie może stanowić podstawy do utworzenia obiektu, na jej podstawie możliwe jest tworzenie klas pochodnych które muszą zaimplementować wszystkie metody abstrakcyjne występujące w klasie bazowej.

9. Wskaż różnice między klasą abstrakcyjną, a interfejsem.

- Przede wszystkim w klasie abstrakcyjnej możemy już umieścić jakąś implementację, natomiast klasy rozszerzające będą miały jedynie obowiązek rozbudować funkcjonalność poprzez zaimplementowanie metod abstrakcyjnych. Klasy abstrakcyjne wymuszają w klasach pochodnych implementacje wszystkich metod abstrakcyjnych.

Poniższy kod programu zawiera definicję klasy abstrakcyjnej wraz z jej składowymi. Wskaż ewentualne błędy.

- Metoda abstrakcyjna nie może mieć ciała

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

- Nie można tworzyć na jej bazie obiektu (tylko dziedziczenie)

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

- Klasa która dziedziczy z klasy abstrakcyjnej i nie dostarcza implementacji metod nadklasy musi być poprzedzona identyfikatorem *abstact.* (nie jest abstrakt jak już ma ciało metody abstrakcyjnej)

13. Wskaż błąd w poniższym kodzie programu

- Nie można dziedziczyć z klas finalnych *final*

Jakie własności posiadają metody finalne?

- Metody finalne – nie możliwe jest ich przesłonięcie w klasach dziedziczących

15. Czym są klasy anonimowe? Wskaż ich zastosowanie.

- Klasy anonimowe to klasy wewnętrzne, nie posiadające nazwy, wykorzystywane do obsługi zdarzeń w programowaniu aplikacji korzystających z GUI SRUI

Rozdział 8.

1. Wymień zalety stosowania procedury obsługi wyjątków.
program nie kończy działania po wystąpieniu wyjątku, po wystąpieniu błędu pozwala na podjęcie odpowiednich działań,

2. Co oznaczają następujące wyjątki:
ArrayIndexOutOfBoundsException – odwołujemy się do nieistniejącego elementu tablicy,

NumberFormatException – gdy chcemy przekonwertować string do typu numerycznego, mając niewłaściwy format,

NullPointerException – gdy aplikacja próbuje użyć NULL w sytuacji gdy potrzebny jest obiekt

3. CharAt wyrzuca błąd – IndexOutOfBoundsException

4. Czym charakteryzują się wyjątki kontrolowane? Z jakich klas się wywodzą?
– muszą zostać obsłużone bezpośrednio w kodzie programu, brak obsługi spowoduje powstanie błędu kompilacji, wywodzą się z klasy exception oprócz RuntimeException

5. Do jakiej kategorii należy wyjątek `FileNotFoundException`?
– wyjątek kontrolowany (próba dostępu do nieistniejącego pliku)

6. Scharakteryzuj składnię instrukcji `try..catch`.
Jeśli którakolwiek z instrukcji umieszczonych wewnątrz bloku try spowoduje błąd, wykonanie programu zostanie przerwane, a sterowanie przekazane do bloku instrukcji catch. Jeśli typ powstałego wyjątku jest zgodny z wymienionym typem w instrukcji catch, zostaną wykonane wszystkie instrukcje znajdujące się wewnątrz bloku catch. Następnie sterowanie przebiegiem wykonania programu zostanie przekazane do kolejnych instrukcji występujących po bloku catch

7. Wskaż zastosowania, jakie pełni blok instrukcji `finalny`.
Blok instrukcji finaly, można umieścić po catch, jego instrukcje zostaną wykonane niezależnie odpowstałych w bloku try błedów, w tym bloku zazwyczaj umieszczone są polecenia, których zadaniem jest zarządzanie zasobami systemu.

8. Jaką funkcję pełni klauzula `throws` w nagłówku metody?
Klauzula throws w nagłówku metody pełni funkcję, jeżeli wywołana metoda zgłasza wyjątek kontrolowany, i jego obsługa polega na przekazaniu go do metody nadrzędnej to wtedy w nagłówku tej

metody nadrzędnej trzeba użyć klauzuli throws, która zawierać musi nazwy wszystkich zgłaszanych wyjątków.

9. Podaj nazwy klas, z których dziedziczyć mogą nowo tworzone wyjątki.
Error, Exception

10. Omów składnię instrukcji `throw`. Jaką funkcję pełni ta instrukcja? Podaj przykłady użycia. składnia: throw objektWyjątku

funkcja: nie da się obsłużyć wyjątków, ich obsługa nie ma sensu, lub po prostu ich wystąpienie może spowodować nieprawidłową pracę w dalszych etapach programu

przykład: throw new WyjatekLiczbaNaturalna()

11. Wskaż różnice pomiędzy klasami `Reader` oraz `InputStream`.
Reader jest znakowy, a InputStram binarny a oba są wejściowymi strumieniami do obsługi ich źródeł.

12. Język Java zawiera parę klas: `InputStreamReader` oraz `OutputStreamWriter`. Jaki jest cel stosowania tych klas?
InputStreamReader – odczytuje strumień bajtowy i zapisuje go w znaki

OutputStreamWriter - zamienia strumień znaków w strumień bajtów

13. Jaką funkcję pełni metoda `mkdirs()` klasy `java.io.File`?
mkdirs() słóży do stworzenia katalogu, zawiera wszystkie niezbędne ale nieistniejące kataloginadrzędne

15. Jaką funkcję pełni pole statyczne `File.separator`? Podaj cel jego stosowania w kontekście tworzenia programów uruchamianych w otoczeniu różnych systemów operacyjnych.
Jest to domyslny separator nazwy zależny od systemu/ platformy (nie zawsze wszystkie systemy mają ten sam separator, a jeśli program ma być uruchamiany w różnych systemach,, powinniśmy użyć File.separator

16. W jaki sposób można uzyskać znak końca linii, niezależny od stosowanego systemu operacyjnego?
\r ^ Line.separate znak końca linii

17. **URL** Uniform Resource Locator, http://www.wikipedia.com/wiki/URL (protokół, host, ścieżka dostępu do zasobu)

18. W jaki sposób realizowane jest dopisywanie danych na końcu pliku? Co stanie się, gdy plik do zapisu zostanie otwarty z wartością parametru `append` równą `false`?
Za pomocą FileWriter można dopisywać coś na końcu pliku, wartość false do parametru append w FileWriter nie pozwala na dopisywane na końcu pliku.

19. Operacja buforowania dostępu do danych pozwala na zwiększenie wydajności odczytu oraz zapisu informacji. W jaki sposób jest realizowana?
realizowana jest poprzez użycie klas BufferedInputStream, BufferedOutputStream, BufferedReader, BufferedWriter.

20. Co oznacza wyjątek: `MalformedURLException`? W jakiej sytuacji jest generowany?
źle sformuowany adres URL, zła składnia lub protokół

Rozdział 9.

1. Czym jest aplet? W jakim środowisku jest uruchamiany?
Aplet stanowi aplikację graficzną utworzoną w języku programowania Java, uruchamianą w środowisku przeglądarki internetowej.

Podstawowe właściwości dotyczące apletu zawarte są w elemencie APPLET, umieszczonym w treści dokumentu HTML. Wartością atrybutu CODE jest nazwa pliku (.class) zawierającego kod programu do wykonania. Atrybuty WIDTH i HEIGHT określają szerokość oraz wysokość apletu w pikselach. Na szczególną uwagę zasługują elementy PARAM umożliwiające przekazywanie wartości z dokumentu HTML bezpośrednio do apletu.

2. Jaka jest różnica pomiędzy apletem, a aplikacją komunikującą się poprzez konsolę?
Aby uruchomić aplet konieczne jest nie tylko utworzenie kodu programu, kompilacja utworzonych plików źródłowych do postaci kodu bajtowego, ale także utworzenie dokumentu HTML, zawierającego wywołanie aplikacji w języku Java (plik *.class).

3. Wymień zasadnicze elementy struktury dokumentu HTML.
<html>

<head>

<title>tytuł dokumentu</title>

</head>

<body>

treść dokumentu ...

<applet

codebase = codebaseURL (lokalizacja plików klas)

archive = archiveList (archiwum plików JAR)

code = appletFile (nazwa pliku kasy)

alt = alternateText (tekst alternatywny)

name = appletInstanceName (nazwa instancji apletu)

width = pixels (wysokość apletu w pikselach)

height = pixels (szerokość apletu w pikselach)

align = alignment (wyrównanie apletu)

vspace = pixels HSPACE = pixels (dodatkowy margines wokół apletu)

>

<param name = appletAttribute1 VALUE = value>

<param name = appletAttribute2 VALUE = value>

. . .

alternatywny HTML

</applet>

treść dokumentu ...

</body>

</html>

4. Jaką funkcję pełni opcjonalny element PARAM występujący w strukturze dokumentu HTML

PARAM umożliwia przekazywanie wartości z dokumentu HTML bezpośrednio do apletu.

5. Opisz sposób przekazywania do apletu wartości numerycznych.
getParameter() zwraca wartość parametru jako String. Jeśli chcemy przekazać wartości numeryczne, jak np. integer, wtedy musimy przekonwerować String w typ, który chcemy, np. int mySize = Integer.parseInt(getParameter("size"));

6. Wskaż różnice występujące pomiędzy klasą `java.awt.Applet`, a `javax.swing.JApplet`? Kiedy stosowana jest każda z nich?
Aplety mogą dziedziczyć po klasie javax.swing.JApplet (jeśli wykorzystywany jest w apletachgraficzny interfejs użytkownika GUI - Swing) lub java.applet.Applet (jeśli stosowane są własne procedury rysujące przy wykorzystaniu dostępnych metod klasy Graphics).

7. Wymiary apletu są wyrażane w pikselach.

8. **RGB** – jeden z modeli przestrzeni barw, opisywanej współrzędnymi RGB. Jego nazwa powstała ze złożenia pierwszych liter angielskich nazw barw: R – red (czerwonej), G – green (zielonej) i B – blue (niebieskiej), z których model ten się składa.

9. Wymień metody umożliwiające rysowanie figur na płaszczyźnie. W jakim pakiecie się one znajdują?
metody umożliwiające rysowanie figur na płaszczyźnie znajdują się w pakiecie java.awt.Graphics, są to np. drawOval(int x, int y, int width, int height), drawRectangle(), drawPolygon(), drawString().

10. **Metoda repaint()** została zdefiniowana w klasie java.awt.Component. Ta metoda jest używana, kiedy komponent musi być przemalowany, np. przy poruszeniu oknem, zmienieniu rozmiaru, itp.

Rozdział 10

1. Wskaż cel stosowania pakietu Swing?
zbiór klas bibliotecznych Swing jest rozwinięciem istniejącej biblioteki AWT (ang. AbstractWindow Toolkit), jest stosowany do tworzenia interfejsów graficznych.

2. Do czego wykorzystywane są podstawowe kontenery (okna)?
JPanel Grupowanie komponentów

JPanel Grupowanie komponentów

JScrollPane Dodawanie pasków przesuwnych

JSplitPane Podział kontenera (w pionie bądź w poziomie)

JTabbedPane Tworzenie zakładek

JToolBar Obsługa paska narzędziowego

3. Scharakteryzuj stosowaną konwencję nazewnicwa klas w pakiecie Swing.

Nazwy wszystkich klas biblioteki Swing zaczynają się od litery J, po czym następuje właściwa

nazwa komponentu, np. JButton.

4. Wymień przykładowe nazwy kontenerów umożliwiające grupowanie obiektów.

JPanel

5. Podaj różnice występujące pomiedzy kontenerami `JPanel` oraz `JScrollPane`.
różnice występujące pomiedzy kontenerami JPanel oraz JScrollPane:

JPanel służy do grupowania komponentów, a JScrollPane do dodawania pasków przesuwnych

6. Wymień komponenty pakietu Swing stosowane przy tworzeniu graficznego interfejsu użytkownika.
JButton Przycisk

JComboBox Lista rozwijana

JList Lista

JCheckBox Pole wyboru

JMenu Menu

JRadioButton Pole opcji

JSlider Suwak

JTextField, JTextArea Pole tekstowe

JLabel Etykieta tekstowa

JProgressBar Wskaźnik postępu

JToolTip Podpowiedź

JColorChooser Panel wyboru koloru

JTable Tabela danych

JTree Układ hierarchiczny danych (struktura drzewa)

7. Jaki komponent odpowiada za etykiety tekstowe? Czy mogą one posiadać ikony?

Za etykiety tekstowe odpowiada komponent JLabel, mogą one posiadać ikony

8. Czym są oraz jaką funkcję pełnią menadżery rozkładu? Wymień ich nazwy.
Menadżery rozkładu umożliwiają rozmieszczenie poszczególnych elementów (kontenerów, czy też komponentów) w oknie aplikacji. Wpływają one zarówno na pozycję elementów względem siebie, jak również na ich rozmiar i sposób wyrównywania. Ich nazwy to:

BorderLayout, BoxLayout, FlowLayout, GridLayout, GridBagLayout, GroupLayout,CardLayout

9. Jaka metoda służy do zmiany domyślnego sposobu rozmieszczenia elementów?

metoda setLayout() służy do zmiany domyślnego sposobu rozmieszczenia elementów

10. Czy pakiet Swing umożliwia obsługę zdarzeń?
tak

11. Opisz zadanie interfejsów `ActionListener` oraz `MouseListener`.
ActionListener służy do zmiany stanu komponentu, a MouseListener do zmiany stanu przycisków myszy

12. Jak powinna zostać zbudowana klasa obsługująca występujące w programie zdarzenia?
Klasa obsługująca zdarzenia implementuje odpowiedni interfejs (interfejsy), np. ActionListener, powinna także posiadać metodę actionPerformed(ActionEvent e)

13. W jaki sposób odbywa się powiązanie komponentu ze zdarzeniem i jego obsługą?
Powiązanie komponentu z obiektem obsługującym zdarzenie realizowane jest za pomocą metody addNazwaInterfejsu(). NazwaInterfejsu oznacza konkretny typ zdarzenia.

14. Która metoda powinna zostać przesłonięta, aby możliwe było wykonanie operacji rysowania?

paintComponent()

15. klasy stosowane w aplikacjach korzystających z grafiki:

java.awt.geom, java.awt.Color, java.awt.Font