

## Zadania do realizacji

1. Zdefiniuj pakiet bez ciała zawierający wartości dwóch stałych fizycznych: prędkość światła w próżni i standardowe przyspieszenie ziemskie. Przetestuj działanie utworzonego pakietu.  
Wskazówka: prędkość światła w próżni to  $c = 299\,792\,458$  m/s, standardowe przyspieszenie ziemskie to  $g = 9,80665$  m/s<sup>2</sup>.
2. a) Utwórz tabelę osoba(id,imie,nazwisko) i dodaj do niej kilka przykładowych rekordów.  
b) Utwórz pakiet o nazwie *statystyki* i zaimplementuj w nim trzy funkcje:  
b1) *LiczbaOsob* – funkcja zwraca liczbę wszystkich osób występujących w tabeli osoba,  
b2) *LiczbaUnikatowychImion* – funkcja zwraca liczbę unikatowych imion występujących w tabeli osoba,  
b3) *LiczbaUnikatowychNazwisk* – funkcja zwraca liczbę unikatowych nazwisk występujących w tabeli osoba,  
c) Przetestuj działanie funkcji z implementowanego pakietu.
3. a) Utwórz tabelę produkt(id\_produkt,nazwa,cena) i dodaj do niej kilka przykładowych rekordów.  
b) Utwórz pakiet o nazwie ceny posiadający dwie przeciążone procedury:  
b1) *podwyzka(p\_procent INTEGER)* – procedura powinna podwyższyć cenę wszystkich produktów w tabeli produkt o zadany procent,  
b2) *podwyzka(p\_procent INTEGER, p\_id\_produkt INTEGER)* – procedura powinna zwiększyć cenę zadanego produktu w tabeli produkt o zadany procent.  
c) Przetestuj działanie utworzonego pakietu.
4. Dana jest specyfikacja pakietu operującego na liczbach zespolonych:  
CREATE OR REPLACE PACKAGE lzesp IS  
TYPE tzesp IS RECORD (re NUMBER, im NUMBER);  
zero tzesp;  
jeden tzesp;  
PROCEDURE wypisz(z tzesp);  
FUNCTION suma(z1 tzesp, z2 tzesp) RETURN tzesp;  
FUNCTION roznica(z1 tzesp, z2 tzesp) RETURN tzesp;  
END lzesp;  
Utwórz ciało pakietu dla zadanej specyfikacji:  
a) Zmienna zespolona *zero* powinna być zainicjowana wartościami re=0 i im=0, a zmienna zespolona *jeden* wartościami re=1 i im=1.

- b) Procedura *wypisz* powinna wypisywać liczbę zespoloną w formacie: (re,im).
- c) Funkcje *suma* i *roznica* powinny obliczać i zwracać odpowiednio sumę i różnicę dwóch liczb zespolonych.
- d) Dla kodu testowego:
- ```
DECLARE
  a lresp.tresp;
  b lresp.tresp;
BEGIN
  a.re:=5; a.im:=6;
  b:=lresp.suma(lresp.roznica(lresp.zero,lresp.jeden),a);
  lresp.wypisz(b);
END;
```
- powinien zostać zwrócony wynik: (4,5).

## Rozwiązania

1. Zdefiniuj pakiet bez ciała zawierający wartości dwóch stałych fizycznych: prędkość światła w próżni i standardowe przyspieszenie ziemskie. Przetestuj działanie utworzonego pakietu.  
Wskazówka: prędkość światła w próżni to  $c = 299\,792\,458$  m/s, standardowe przyspieszenie ziemskie to  $g = 9,80665$  m/s<sup>2</sup>.

```
1 CREATE OR REPLACE PACKAGE stale_fizyczne IS
2   c CONSTANT NUMBER:=299792458;
3   g CONSTANT NUMBER:=9.80665;
4 END stale_fizyczne;
5 /
6 SET SERVEROUTPUT ON
7 /
8 BEGIN
9   dbms_output.put_line('Prędkość światła w próżni wynosi '||stale_fizyczne.c);
10  dbms_output.put_line('Przyspieszenie ziemskie wynosi '||stale_fizyczne.g);
11 END;
```



Script Output x

Task completed in 0,047 seconds

Prędkość światła w próżni wynosi 299792458  
Przyspieszenie ziemskie wynosi 9,80665

2. a) Utwórz tabelę osoba(id,imie,nazwisko) i dodaj do niej kilka przykładowych rekordów.
- b) Utwórz pakiet o nazwie *statystyki* i zaimplementuj w nim trzy funkcje:
  - b1) *LiczbaOsob* – funkcja zwraca liczbę wszystkich osób występujących w tabeli osoba,
  - b2) *LiczbaUnikatowychImion* – funkcja zwraca liczbę unikatowych imion występujących w tabeli osoba,
  - b3) *LiczbaUnikatowychNazwisk* – funkcja zwraca liczbę unikatowych nazwisk występujących w tabeli osoba,
- c) Przetestuj działanie funkcji z implementowanego pakietu.

```
1 DROP TABLE osoba;
2 /
3 CREATE TABLE osoba (
4     id NUMBER(11) PRIMARY KEY,
5     imie VARCHAR2(15) NOT NULL,
6     nazwisko VARCHAR2(20) NOT NULL
7 );
8 /
9 INSERT INTO osoba(id,imie,nazwisko) VALUES (1, 'Jan', 'Kowalski');
10 INSERT INTO osoba(id,imie,nazwisko) VALUES (2, 'Jerzy', 'Kowalski');
11 INSERT INTO osoba(id,imie,nazwisko) VALUES (3, 'Adam', 'Kowalski');
12 INSERT INTO osoba(id,imie,nazwisko) VALUES (4, 'Jerzy', 'Nowak');
13 INSERT INTO osoba(id,imie,nazwisko) VALUES (5, 'Adam', 'Nowak');
14 /
15 CREATE OR REPLACE PACKAGE statystyki IS
16     FUNCTION LiczbaOsob RETURN NUMBER;
17     FUNCTION LiczbaUnikatowychImion RETURN NUMBER;
18     FUNCTION LiczbaUnikatowychNazwisk RETURN NUMBER;
19 END statystyki;
20 /
```

```
21 CREATE OR REPLACE PACKAGE BODY statystyki IS
22   FUNCTION LiczbaOsob RETURN NUMBER IS
23     v_ile NUMBER(11);
24   BEGIN
25     SELECT count(*) INTO v_ile FROM osoba;
26     RETURN v_ile;
27   END LiczbaOsob;
28   FUNCTION LiczbaUnikatowychImion RETURN NUMBER IS
29     v_ile NUMBER(11);
30   BEGIN
31     SELECT count(*) INTO v_ile FROM (SELECT DISTINCT imie FROM osoba) t;
32     RETURN v_ile;
33   END LiczbaUnikatowychImion;
34   FUNCTION LiczbaUnikatowychNazwisk RETURN NUMBER IS
35     v_ile NUMBER(11);
36   BEGIN
37     SELECT count(*) INTO v_ile FROM (SELECT DISTINCT nazwisko FROM osoba) t;
38     RETURN v_ile;
39   END LiczbaUnikatowychNazwisk;
40 END statystyki;
41 /

42 SET SERVEROUTPUT ON
43 /
44 BEGIN
45   dbms_output.put_line('Liczba osób: '||statystyki.LiczbaOsob);
46   dbms_output.put_line('Liczba unikatowych imion: '||
47     statystyki.LiczbaUnikatowychImion);
48   dbms_output.put_line('Liczba unikatowych nazwisk: '||
49     statystyki.LiczbaUnikatowychNazwisk);
50 END;
```



Script Output x

Task completed in 0,14 seconds


Liczba osób: 5  
Liczba unikatowych imion: 3  
Liczba unikatowych nazwisk: 2

3. a) Utwórz tabelę produkt(id\_produkt,nazwa,cena) i dodaj do niej kilka przykładowych rekordów.
- b) Utwórz pakiet o nazwie ceny posiadający dwie przeciążone procedury:
  - b1) podwyzka(p\_procent INTEGER) – procedura powinna podwyższyć cenę wszystkich produktów w tabeli produkt o zadany procent,
  - b2) podwyzka(p\_procent INTEGER, p\_id\_produkt INTEGER) – procedura powinna zwiększyć cenę zadanego produktu w tabeli produkt o zadany procent.
- c) Przetestuj działanie utworzonego pakietu.


```

1 DROP TABLE produkt;
2 /
3 CREATE TABLE produkt(id_produkt INTEGER, nazwa VARCHAR2(50), cena NUMBER(10,2));
4 /
5 INSERT INTO produkt(id_produkt,nazwa,cena) VALUES(1,'Zeszyt A5',4);
6 INSERT INTO produkt(id_produkt,nazwa,cena) VALUES(2,'Gumka',3);
7 INSERT INTO produkt(id_produkt,nazwa,cena) VALUES(3,'Cyrkiel',10);
8 INSERT INTO produkt(id_produkt,nazwa,cena) VALUES(4,'Mazaki',15);
9 /
10 CREATE OR REPLACE PACKAGE ceny IS
11     PROCEDURE podwyzka(p_procent INTEGER);
12     PROCEDURE podwyzka(p_procent INTEGER, p_id_produkt INTEGER);
13 END ceny;
14 /
15 CREATE OR REPLACE PACKAGE BODY ceny IS
16     PROCEDURE podwyzka(p_procent INTEGER) IS
17     BEGIN
18         UPDATE produkt SET cena=cena*(1+p_procent/100);
19     END;
20     PROCEDURE podwyzka(p_procent INTEGER, p_id_produkt INTEGER) IS
21     BEGIN
22         UPDATE produkt SET cena=cena*(1+p_procent/100)
23         WHERE id_produkt=p_id_produkt;
24     END;
25 END ceny;
26 /
27 SELECT * FROM produkt;
28 /
29 BEGIN
30     ceny.podwyzka(10);
31     ceny.podwyzka(100, 1);
32 END;
33 /
34 SELECT * FROM produkt;

```



| ID_PRODUKT | NAZWA       | CENA |
|------------|-------------|------|
| 1          | 1 Zeszyt A5 | 4    |
| 2          | 2 Gumka     | 3    |
| 3          | 3 Cyrkiel   | 10   |
| 4          | 4 Mazaki    | 15   |



| ID_PRODUKT | NAZWA       | CENA |
|------------|-------------|------|
| 1          | 1 Zeszyt A5 | 8,8  |
| 2          | 2 Gumka     | 3,3  |
| 3          | 3 Cyrkiel   | 11   |
| 4          | 4 Mazaki    | 16,5 |

4. Dana jest specyfikacja pakietu operującego na liczbach zespolonych:

```
CREATE OR REPLACE PACKAGE lzesp IS
  TYPE tzesp IS RECORD (re NUMBER, im NUMBER);
  zero tzesp;
  jeden tzesp;
  PROCEDURE wypisz(z tzesp);
  FUNCTION suma(z1 tzesp, z2 tzesp) RETURN tzesp;
  FUNCTION roznica(z1 tzesp, z2 tzesp) RETURN tzesp;
END lzesp;
```

Utwórz ciało pakietu dla zadanej specyfikacji:


- a) Zmienna zespolona *zero* powinna być zainicjowana wartościami re=0 i im=0, a zmienna zespolona *jeden* wartościami re=1 i im=1.
- b) Procedura *wypisz* powinna wypisywać liczbę zespoloną w formacie: (re,im).
- c) Funkcje *suma* i *roznica* powinny obliczać i zwracać odpowiednio sumę i różnicę dwóch liczb zespolonych.
- d) Dla kodu testowego:

```
DECLARE
  a lzesp.tzesp;
  b lzesp.tzesp;
BEGIN
  a.re:=5; a.im:=6;
  b:=lzesp.suma(lzesp.roznica(lzesp.zero,lzesp.jeden),a);
  lzesp.wypisz(b);
END;
```

powinien zostać zwrócony wynik: (4,5).

```
1 CREATE OR REPLACE PACKAGE lzesp IS
2   TYPE tzesp IS RECORD (re NUMBER, im NUMBER);
3   zero tzesp;
4   jeden tzesp;
5   PROCEDURE wypisz(z tzesp);
6   FUNCTION suma(z1 tzesp, z2 tzesp) RETURN tzesp;
7   FUNCTION roznica(z1 tzesp, z2 tzesp) RETURN tzesp;
8 END lzesp;
9 /
```

```
10 CREATE OR REPLACE PACKAGE BODY lzesp IS
11   PROCEDURE wypisz(z tzesp) IS
12     BEGIN
13       DBMS_OUTPUT.PUT_LINE('('||z.re||', '||z.im||')');
14     END;
15   FUNCTION suma( z1 tzesp, z2 tzesp ) RETURN tzesp IS
16     tmp tzesp;
17     BEGIN
18       tmp.re:=z1.re+z2.re;
19       tmp.im:=z1.im+z2.im;
20       RETURN tmp;
21     END;
22   FUNCTION roznica(z1 tzesp,z2 tzesp) RETURN tzesp IS
23     tmp tzesp;
24     BEGIN
25       tmp.re:=z1.re-z2.re;
26       tmp.im:=z1.im-z2.im;
27       RETURN tmp;
28     END;
29 BEGIN
30   zero.re:=0;
31   zero.im:=0;
32   jeden.re:=1;
33   jeden.im:=1;
34 END lzesp;
35 /
36 SET SERVEROUTPUT ON
37 /
38 --kod testujący pakiet z treści zadania
39 DECLARE
40   a lzesp.tzesp;
41   b lzesp.tzesp;
42 BEGIN
43   a.re:=5; a.im:=6;
44   b:=lzesp.suma(lzesp.roznica(lzesp.zero,lzesp.jeden), a);
45   lzesp.wypisz(b);
46 END;
```



Script Output x

Task completed in 0,079 seconds

PACKAGE LZESP compiled  
PACKAGE BODY LZESP compiled  
anonymous block completed  
(4, 5)