

Programowanie obiektowe

Wykład 4: Ochrona implementacji.

3/17/2013

S.Denziak: Programowanie obiektowe - Java

1

Problem udostępniania kodu

- Jak zorganizować zbiory klas?
 - Konflikt nazw klas
 - Organizacja plików na dysku
- Jak zabezpieczyć się przed nieprawidłowym użyciem?

```
Np. class Data {
    byte dzien, miesiac;
    int rok;
    ...
    Data(byte d, byte m,
        int r) {
        ...
    }
}
```

```
Data d;
d= new Data(30,3,2009);
...
d.miesiac = 2;   ???
```

3/17/2013

S.Denziak: Programowanie obiektowe - Java

2

Pakiety

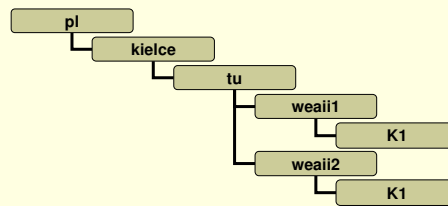
Pakiet: jednostka biblioteczna

```
package nazwa_pakietu;  
...  
public class MojaKlasa { ...}
```

```
import nazwa_pakietu;  
.....  
MojaKlasa m= new MojaKlasa();
```

```
CLASSPATH= c:\java\pakiety;d:\java2\pakiety;d:\java2\bib.jar
```

```
package pl.kielce.tu.weaii1;  
public class K1 { ...}  
  
package pl.kielce.tu.weaii2;  
public class K1 { ...}
```



3/17/2013

S.Denziak: Programowanie obiektowe - Java

3

Korzystanie z pakietów

Importowanie klas z bibliotek (pakietów):

```
import pakiet.klasa;  
import pakiet.*;
```

Pakiety standardowe:

```
java.lang  
import java.util.*;  
import java.math.*;
```

3/17/2013

S.Denziak: Programowanie obiektowe - Java

4

Modyfikatory dostępu (1)

public – dostęp publiczny (dla wszystkich)

```
package p1;
public class K1 {
    ....
    public int i1;
    public int m1(){
    }

    class K2{
        int x;
        K1 o= new K1();
        o.i1=5;
        x=o1.m1();
        ...
    }
}
```

3/17/2013

S.Deniziak: Programowanie obiektowe - Java

5

```
package p2;
import p1;
public class K3 {
    int y;
    K1 o1= new K1();
    o1.i1=5;
    y=o1.m1();
    ...
}
```

Modyfikatory dostępu (2)

private – dostęp prywatny (tylko w ramach klasy)

```
package p1;
public class K1 {
    ....
    private int i1;
    private int m1(){
    }

    class K2{
        int x;
        K1 o= new K1();
        o.i1=5;
        x=o.m1();
        ...
    }
}
```

3/17/2013

S.Deniziak: Programowanie obiektowe - Java

6

```
package p2;
import p1;
public class K3 {
    int y;
    K1 o1= new K1();
    o1.i1=5;
    y=o1.m1();
    ...
}
```

Po co private?

```
Np. class Data {
    private byte d, m;
    private int r;
    public void zmienDate(byte d, byte m,
        int r) {
        ...
    }
    public int rok() {return r;}
    public byte dzien() {return d;}
    public byte miesiac() {return m;}
    Data(byte d, byte m,
        int r) {
        ...
    }
}
```

Kontrola dostępu do pól

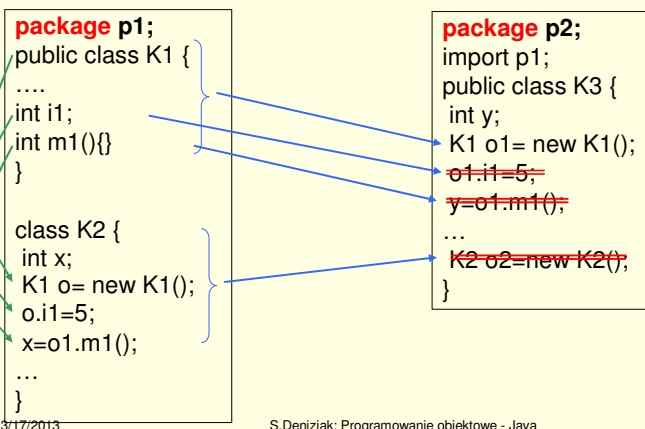
3/17/2013

S.Denziak: Programowanie obiektowe - Java

7

Modyfikatory dostępu (3)

dostęp „przyjazny” (tylko w ramach pakietu)



3/17/2013

S.Denziak: Programowanie obiektowe - Java

8

Interfejs a implementacja

- Interfejs:
 - Tylko elementy publiczne!
 - Interfejs do elementów chronionych!
- Implementacja
 - Elementy publiczne: implementacja może być zmieniona przez użytkownika klasy!
 - Elementy chronione: ochrona implementacji (po co?)
 - Elementy prywatne: lokalne deklaracje w klasie
 - Elementy „przyjazne”: lokalne deklaracje w pakiecie

3/17/2013

S.Denziak: Programowanie obiektowe - Java

9

Przykład 1

```
class Zupa {
    private Zupa() {}
    public static Zupa robZupe() {
        return new Zupa();
    }
    private static Zupa ps1 = new Zupa();
    public static Zupa dostep() {
        return ps1;
    }
    public void f() {}
}

class Kanapka {
    void f() { new Lunch(); }
}
```

```
public class Lunch {
    void test() {
Zupa priv1 = new Zupa();
        Zupa priv2 = Zupa.robZupe();
        Kanapka f1 = new Kanapka();
        Zupa.dostep().f();
    }
}
```

3/17/2013

S.Denziak: Programowanie obiektowe - Java

10

Przykład 2

```
public class Klasa1 {
    private int ro;
    Klasa1(int p) {
        ro=p;
    }
    public int dostep() {
        return ro;
    }
}
```

3/17/2013

S.Deniziak: Programowanie obiektowe - Java

11

Podsumowanie

- Ukrywanie implementacji
 - Tylko twórcy mają dostęp do wszystkiego
 - Pozostali w takim zakresie na jaki pozwoli autor
- Rozsądne używanie kwalifikatorów dostępu
 - Tak, aby ograniczyć nieprawidłowe korzystanie
 - Nie nadużywać!
- Aktualna kartoteka w CLASSPATH!!
 - **Np. CLASSPATH=.;c:\java\lib;**

3/17/2013

S.Deniziak: Programowanie obiektowe - Java

12

Pytania

1. Do czego służą kwalifikatory dostępu?
2. Znaczenie i zastosowanie kwalifikatora `private`.
3. Znaczenie i zastosowanie kwalifikatora `public`.
4. Dostępność klas, pól i metod bez specyfikacji dostępu.
5. Do czego służą pakiety.

Koniec