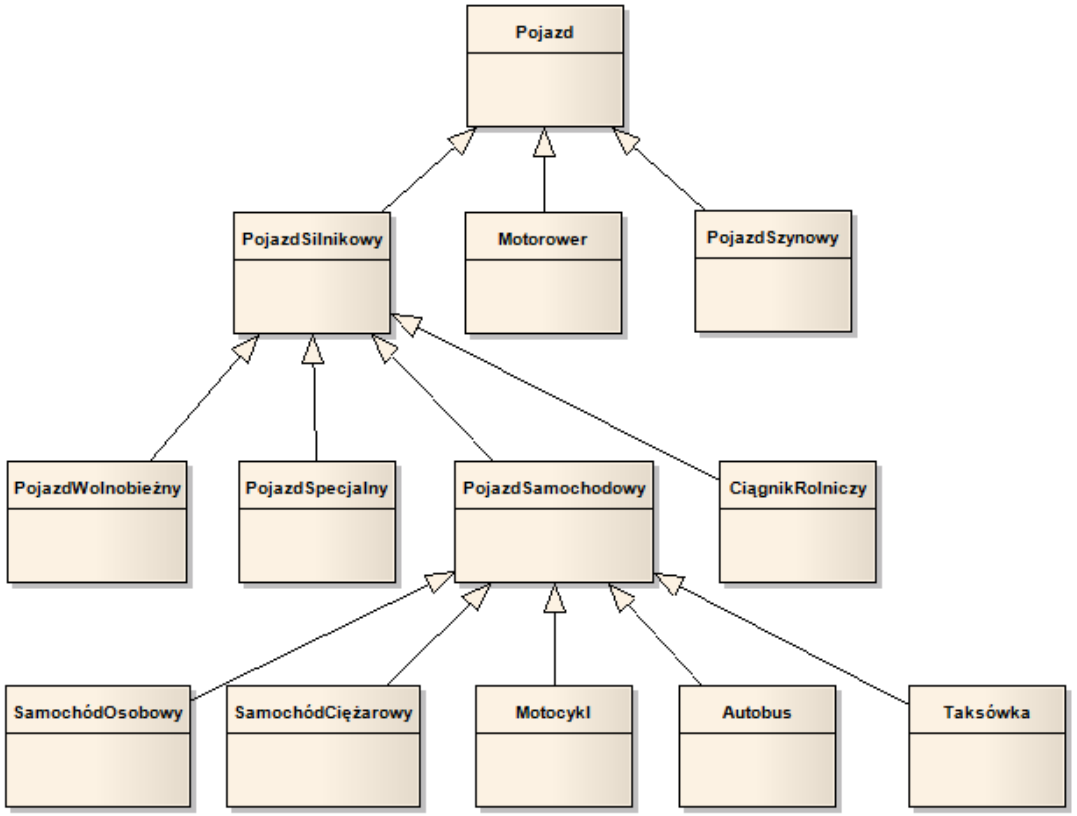
7. Lekcja

Programowanie obiektowe

Programować obiektowo czyli jak?



- **programowanie** gdzie definiuje się obiekty – elementy łączące stan (czyli dane, nazywane najczęściej polami) i zachowanie (czyli procedury, tu: metody). Programowanie obiektowe ma ułatwić pisanie, konserwację i wielokrotne użycie programów lub ich fragmentów.

Klasa a obiekt

- **Klasa –** definiuje zawartość i funkcjonalność obiektów. Zawiera również informację o tym czego potrzeba i jak stworzyć obiekt.
- **Obiekt –** instancja klasy, reprezentująca rzeczywisty obiekt w modelu. Dostarcza pola i metody do użycia w kodzie programu.

Modyfikatory dostępu

```
public class ExampleClass {

    private int fieldOnlyForClass;
    protected int fieldForChildren;
    public int fieldForAll;

    private void methodeOnlyForClass(){};
    protected void methodeForChildren(){};
    public void methodeForAll(){};

}
```

Służą do określenia poziomu dostępu do elementu klasy.

- private – dostęp tylko z poziomu klasy
- proteced – dostęp z poziomu klas dziedziczących
- public – dostęp z klasy nadrzędnej

Pole, konstruktor, metoda klasy

- Pole – zmienna obiektu.
- Konstruktor – metoda inicjalizująca obiekt do życia bazując na klasie.
- Metoda – specyficzna funkcjonalność dla obiektu na bazie klasy

```
public class Point {

    private int x;
    private int y;

    public Point(int x, int y) {
        this.x = x;
        this.y = y;
    }

    public int getX() {
        return x;
    }

    public void setX(int x) {
        this.x = x;
    }

    public int getY() {
        return y;
    }

    public void setY(int y) {
        this.y = y;
    }
}
```

Konstruktor

Specjalna metoda do tworzenia obiektów danej klasy. Służy do początkowej inicjalizacji elementów klasy.

- Każda klasa bez konstruktora posiada niejawny konstruktor bezargumentowy.
- Do użycia konstruktora wykorzystywane jest słówko new
- Do użycia publicznych elementów obiektu używany jest operator „ . ”

```
public Point(int x, int y) {
    this.x = x;
    this.y = y;
}
```

```
public class Main {

    public static void main(String[] args) {

        Main myMain = new Main();

        Point point = new Point( x: 10,  y: 5);

        System.out.println("My point is P(" + point.getX() + "," + point.getY() + ")");

    }


}
```

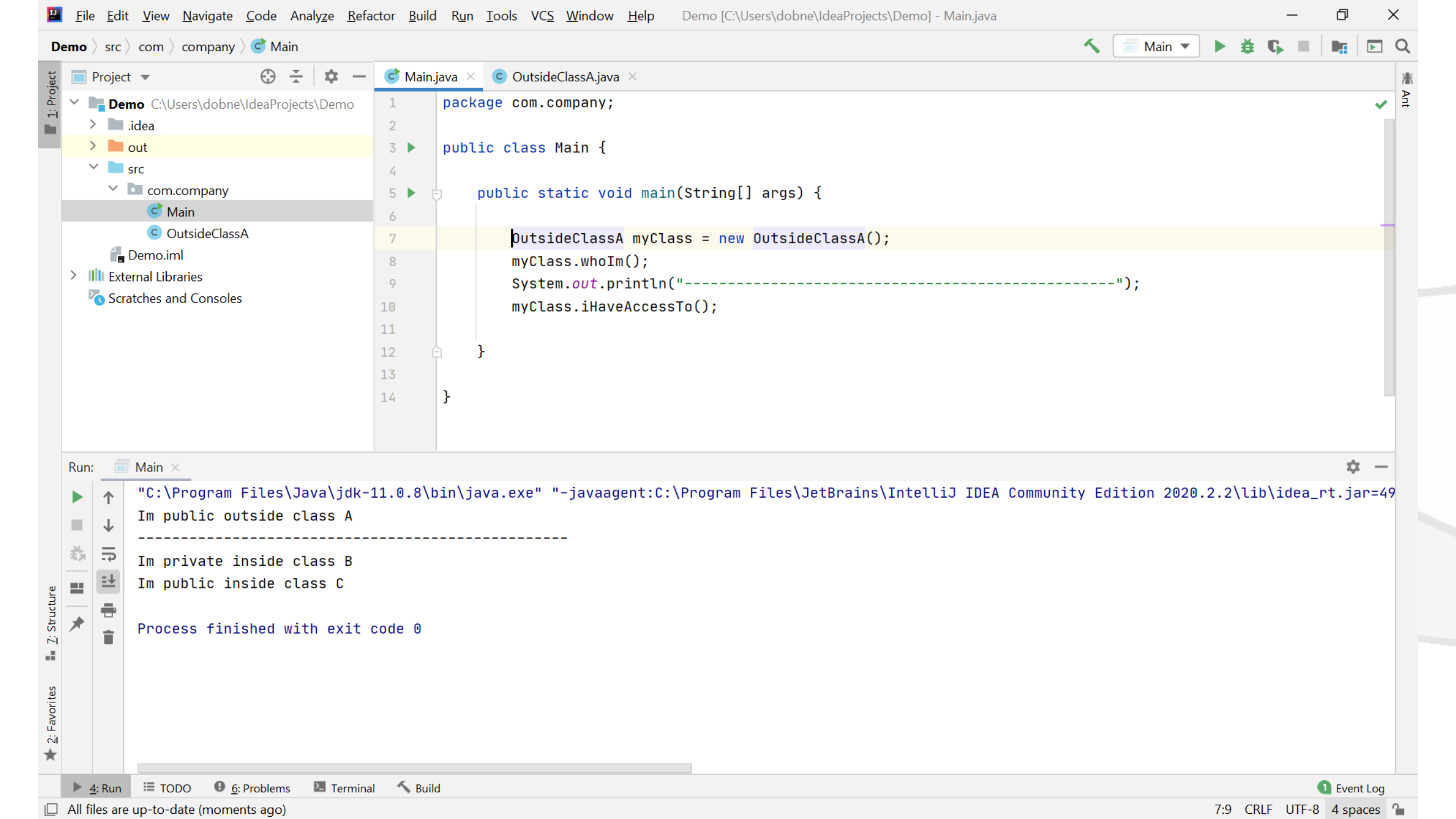
```java
public class OutsideClassA {

    public void whoIm() {
        System.out.println("Im public outside class A");
    }

    public void iHaveAccessTo() {
        new InsideClassB().whoIm();
        new InsideClassC().whoIm();
    }

    private class InsideClassB {

        public void whoIm() {
            System.out.println("Im private inside class B");
        }

        public void iHaveAccessTo() {
            new OutsideClassA().whoIm();
            new InsideClassC().whoIm();
        }

    }

    public class InsideClassC {

        public void whoIm() {
            System.out.println("Im public inside class C");
        }

        public void iHaveAccessTo() {
            new OutsideClassA().whoIm();
            new InsideClassB().whoIm();
        }
    }

}
```

Klasa wewnętrzna i zewnętrzna

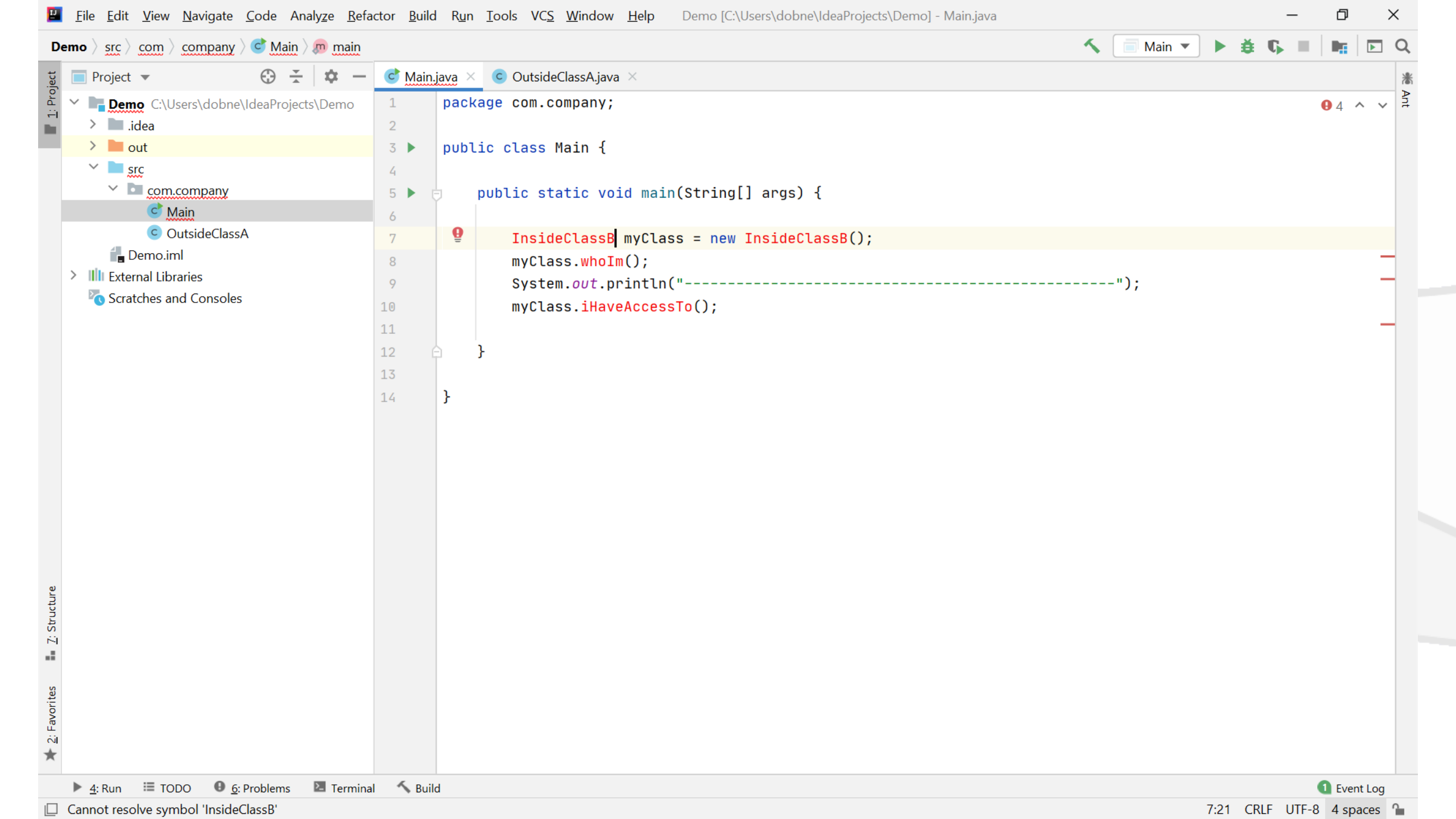
Publiczna zewnętrzna klasa A i metody klasy A

Prywatna wewnętrzna klasa B i metody klasy B

Publiczna wewnętrzna klasa C i metody klasy C


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Demo [C:\Users\dobne\IdeaProjects\Demo] - Main.java
Demo src com company Main
Main
Project
Demo C:\Users\dobne\IdeaProjects\Demo
.idea
out
src
com.company
Main
OutsideClassA
Demo.iml
External Libraries
Scratches and Consoles
Main.java
OutsideClassA.java
1 package com.company;
2
3 public class Main {
4
5 public static void main(String[] args) {
6
7 OutsideClassA myClass = new OutsideClassA();
8 myClass.whoIm();
9 System.out.println("----------");
10 myClass.iHaveAccessTo();
11
12 }
13
14 }
Run: Main
"C:\Program Files\Java\jdk-11.0.8\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2020.2.2\lib\idea_rt.jar=49
Im public outside class A

Im private inside class B
Im public inside class C
Process finished with exit code 0
1: Project
2: Favorites
7: Structure
Ant
4: Run
TODO
6: Problems
Terminal
Build
Event Log
All files are up-to-date (moments ago)
7:9 CRLF UTF-8 4 spaces


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Demo [C:\Users\dobne\IdeaProjects\Demo] - Main.java
Demo src com company Main main
Main
Project
Demo C:\Users\dobne\IdeaProjects\Demo
.idea
out
src
com.company
Main
OutsideClassA
Demo.iml
External Libraries
Scratches and Consoles
Main.java
OutsideClassA.java
1 package com.company;
2
3 public class Main {
4
5 public static void main(String[] args) {
6
7 InsideClassB myClass = new InsideClassB();
8 myClass.whoIm();
9 System.out.println("----------");
10 myClass.iHaveAccessTo();
11
12 }
13
14 }
4
Ant
1: Project
7: Structure
2: Favorites
4: Run
TODO
6: Problems
Terminal
Build
Event Log
Cannot resolve symbol 'InsideClassB'
7:21
CRLF
UTF-8
4 spaces

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help Demo [C:\Users\dobne\IdeaProjects\Demo] - Main.java

Demo > src > com > company > Main

Main

Project

Demo C:\Users\dobne\IdeaProjects\Demo
.idea
out
src
com.company
Main
OutsideClassA
Demo.iml
External Libraries
Scratches and Consoles

Main.java OutsideClassA.java

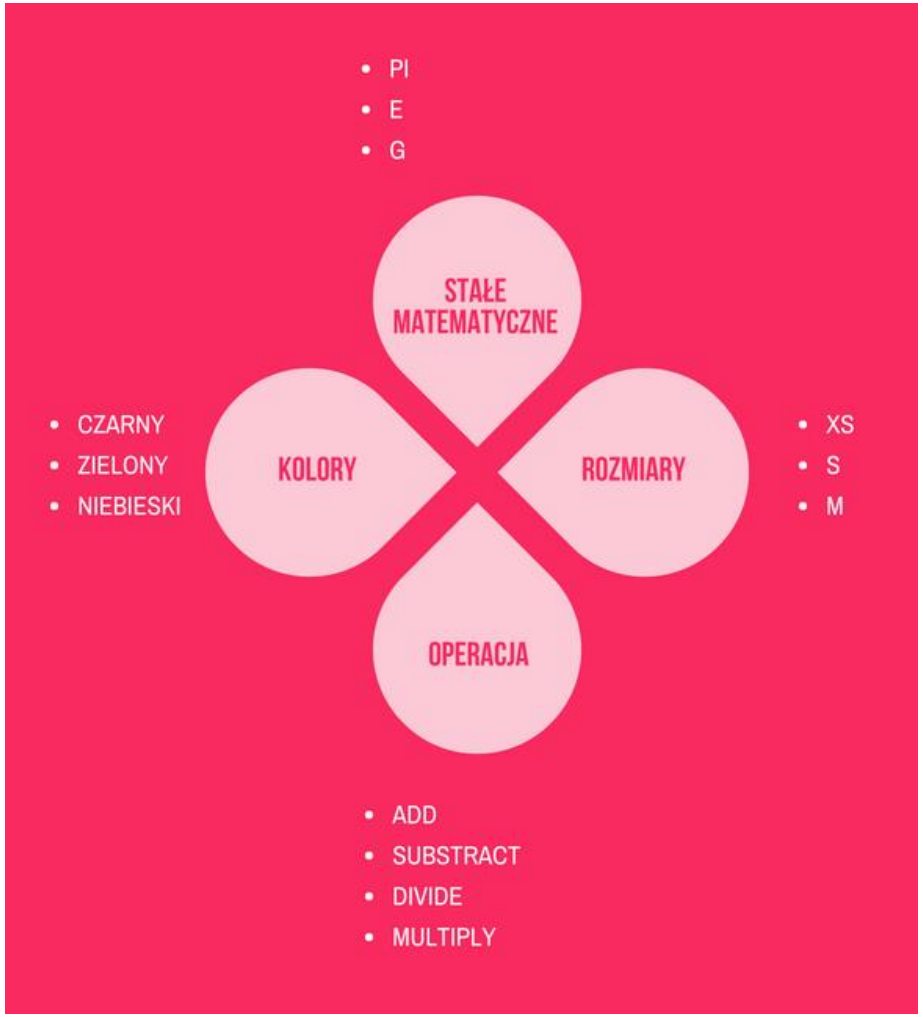
```
1  package com.company;
2
3  public class Main {
4
5      public static void main(String[] args) {
6
7          OutsideClassA.InsideClassC myClass = new OutsideClassA.InsideClassC();
8          myClass.whoIm();
9          System.out.println("----------------------------------------------------");
10         myClass.iHaveAccessTo();
11
12     }
13
14 }
```

1: Project 2: Favorites 7: Structure Ant

4: Run TODO 6: Problems Terminal Build Event Log

All files are up-to-date (6 minutes ago) 13:1 CRLF UTF-8 4 spaces

Enum



Tak zwany **typ wyliczeniowy.** Służy do definiowania własnych typów.

Np.

- Miesiąc(JAN, FEB, MAR…, DEC)
- Kolor (RED, YELLOW, … GREEN)
- Rozmiar (XS, S, M… ,XXL)

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help Demo [C:\Users\dobne\IdeaProjects\Demo] - Main.java

Demo > src > com > company > Main

Main

Project

Demo C:\Users\dobne\IdeaProjects\Demo
.idea
out
src
com.company
Main
Demo.iml
External Libraries
Scratches and Consoles

Main.java

```
1  package com.company;
2
3  public class Main {
4
5      public enum ClotheSize {
6          XSS, XS, S, M, L, XL, XXL
7      }
8
9      public enum Colour {
10         RED, YELLOW, BLUE, GREEN, ORANGE, VIOLET, BLACK
11     }
12
13     public enum Month {
14         JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, DEC
15     }
16
17     public static void main(String[] args) {
18
19         Month myBirthday = Month.JAN;
20         ClotheSize myClothesSize = ClotheSize.L;
21         Colour myFavoriteColour = Colour.GREEN;
22
23         System.out.println("My birthday is in " + myBirthday);
24         System.out.println("My clothes size is " + myClothesSize);
25         System.out.println("My favorite color is " + myFavoriteColour);
```

23

Run: Main

```
My birthday is in JAN
My clothes size is L
My favorite color is GREEN
```

4: Run TODO 6: Problems Terminal Build Event Log

Build completed successfully in 15 s 16 ms (2 minutes ago) 20:47 CRLF UTF-8 4 spaces

Tyle z teorii! Zadania