

# Scala in Practice

## lab 06

### Acceptance criteria:

Create Scala program with:

- Package *pizzeria* with sub-packages & classes to represent abstractions in pizza-place which:
  - sells 3 types of pizzas: *Margarita* (5\$), *Pepperoni* (6.5\$) & *Funghi* (7\$)
  - in 3 sizes: *small*, *regular* & *large*
  - on 2 crust types: *thin* & *thick* (*same price*)
  - with 2 kinds of toppings: *ketchup* (+0.5\$) & *garlic* (+0.5\$)
  - with 1 kind of meat: *salami* (+1\$)
  - with 1 kind of drink: *lemonade* (2\$)
  - has 2 kinds of discounts: *student* & *senior*
- This should be a valid definition:

```

case class Pizza(
  type:....,
  size:....,
  crust:....,
  extraMeat:...., //optional meat
  extraTopping:... //optional topping
) {

  override def toString() = ??? //pretty print the pizza

  val price: Double = ??? //calculated price for pizza. When
    type=small than price is 90% & if type=large than
    price is 150%
}

```

- Package *orders* with
 

```

class Order(
  name: String,
  address: String,
  phone:..., //mandatory validated phone-number (hint: regex)
  pizzas:....,
  drinks:....,
  discount:..., //optional value

  specialInfo: ..., //optional text, like: "Ring doesnt work,
    please knock"
)

```

# Scala in Practice

## lab 06

) {

```

override def toString() = ??? //pretty print the order

def extraMeatPrice: Option[Double] = ???

def pizzasPrice: Option[Double] = ???

def drinksPrice: Option[Double] = ???

def priceByType(type:...): Option[Double] = ??? //total price of
all pizzas by type (Margarita, Pepperoni & Funghi)

val price: Double = ??? //total price of order. When
discount=student than price for all pizzas is reduced by -5%
& if discount=senior than price for all pizzas & drinks is
reduced by -7%

```

}

- Create *application entry-point* object with some example tests for the above implementation

*Note:* Dont use any **vars & nulls**

Michał Kowalczykiewicz