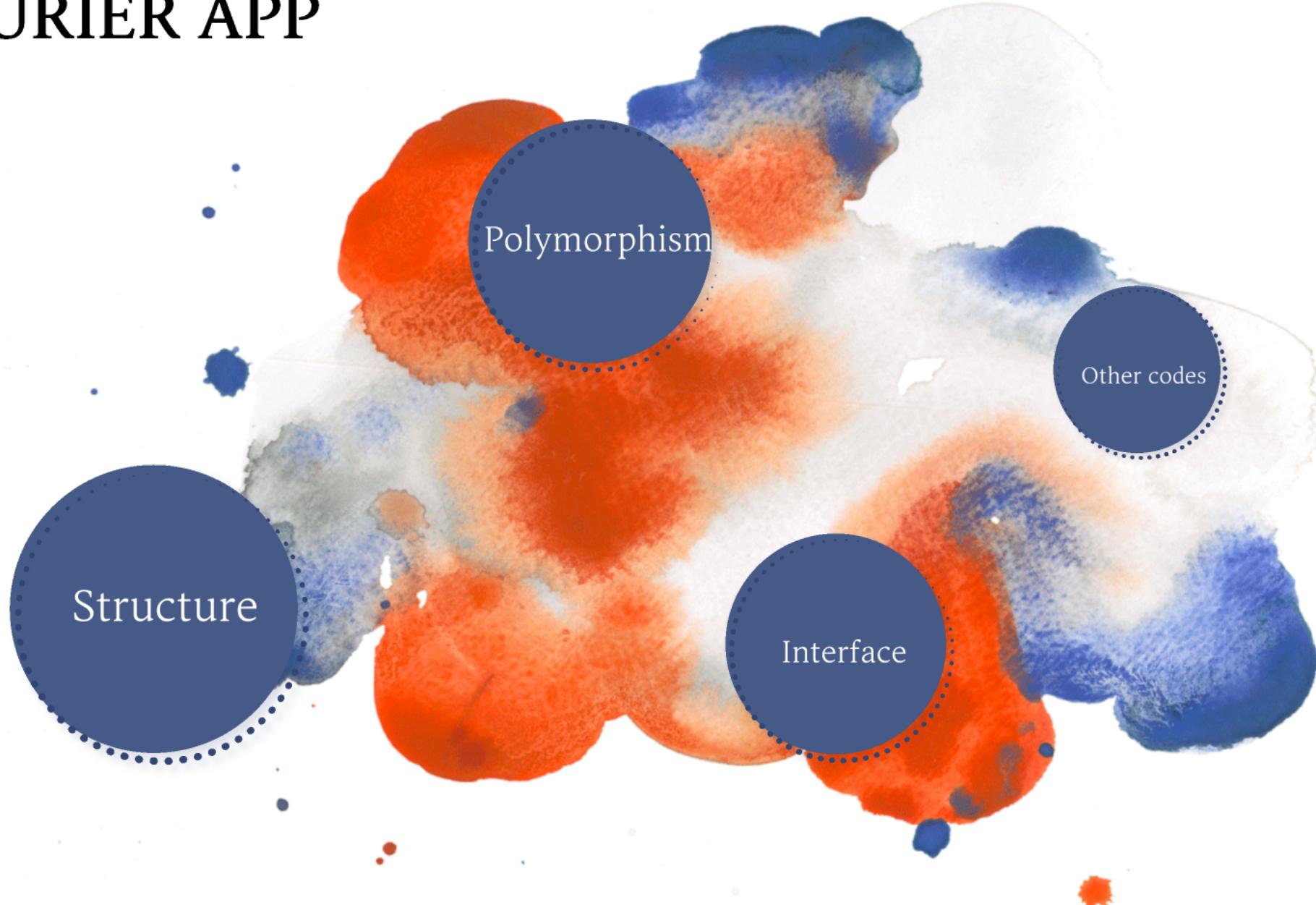
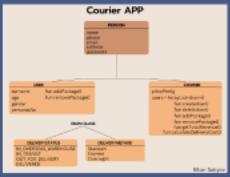
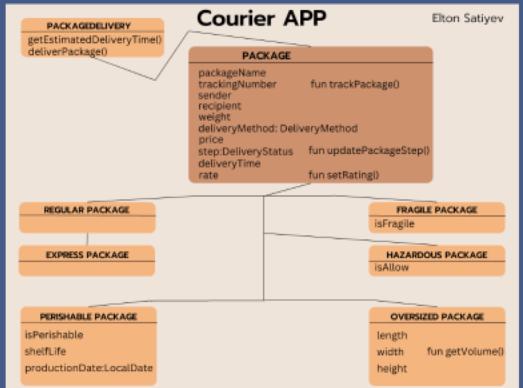


COURIER APP



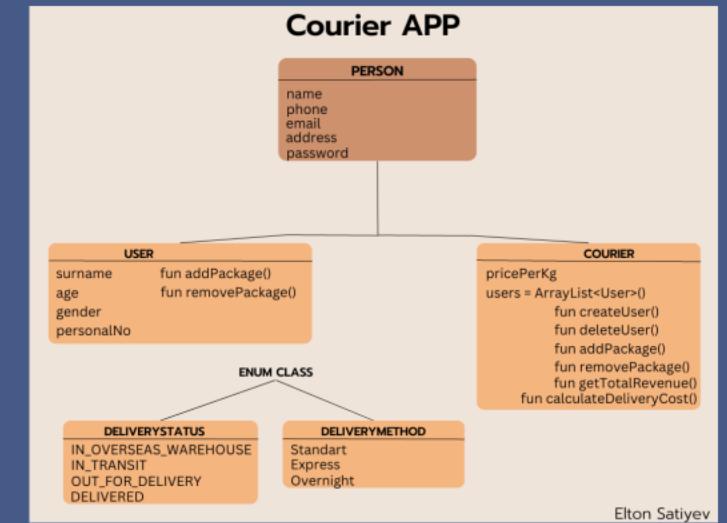
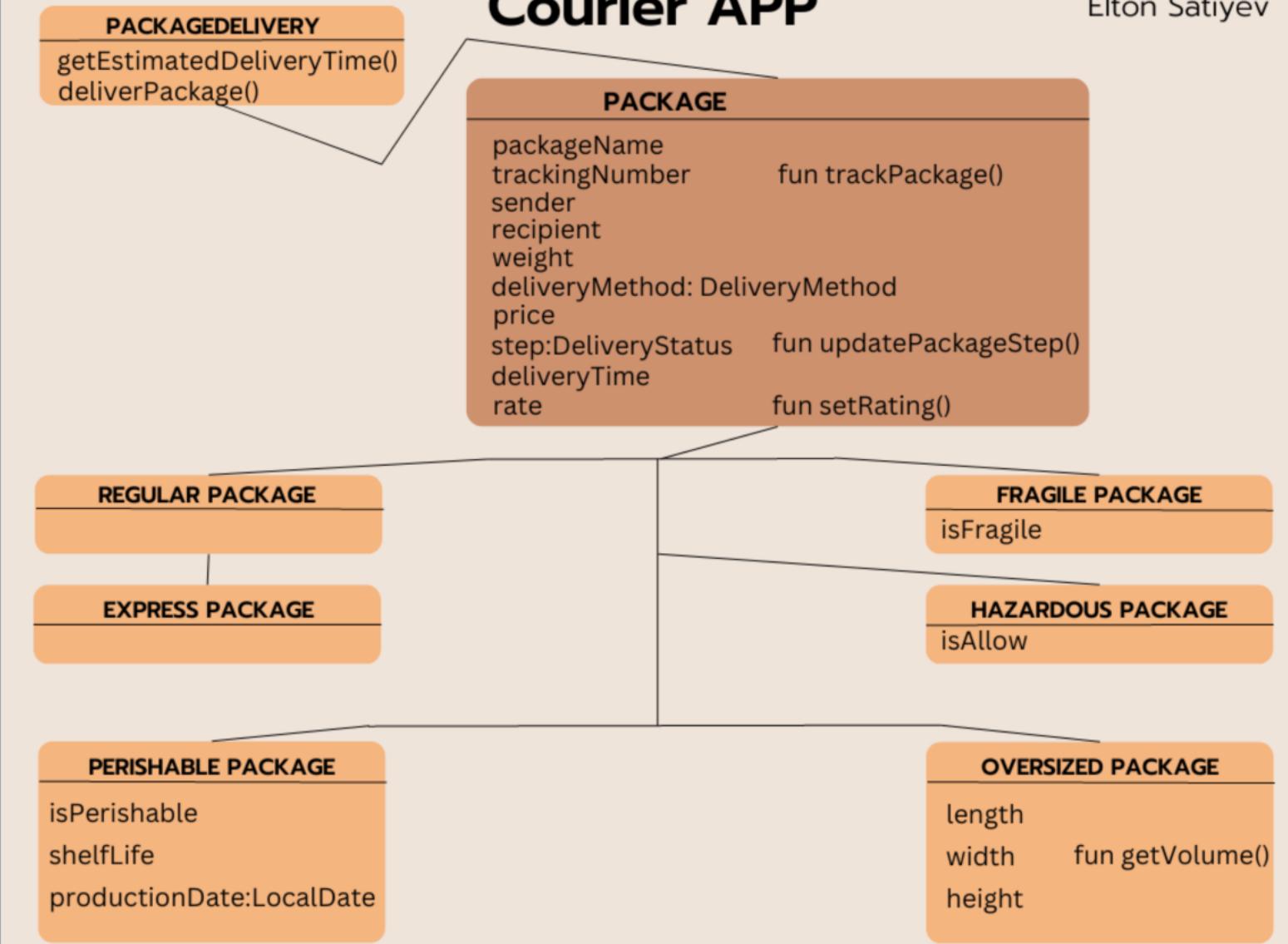
엘톤



App Interface

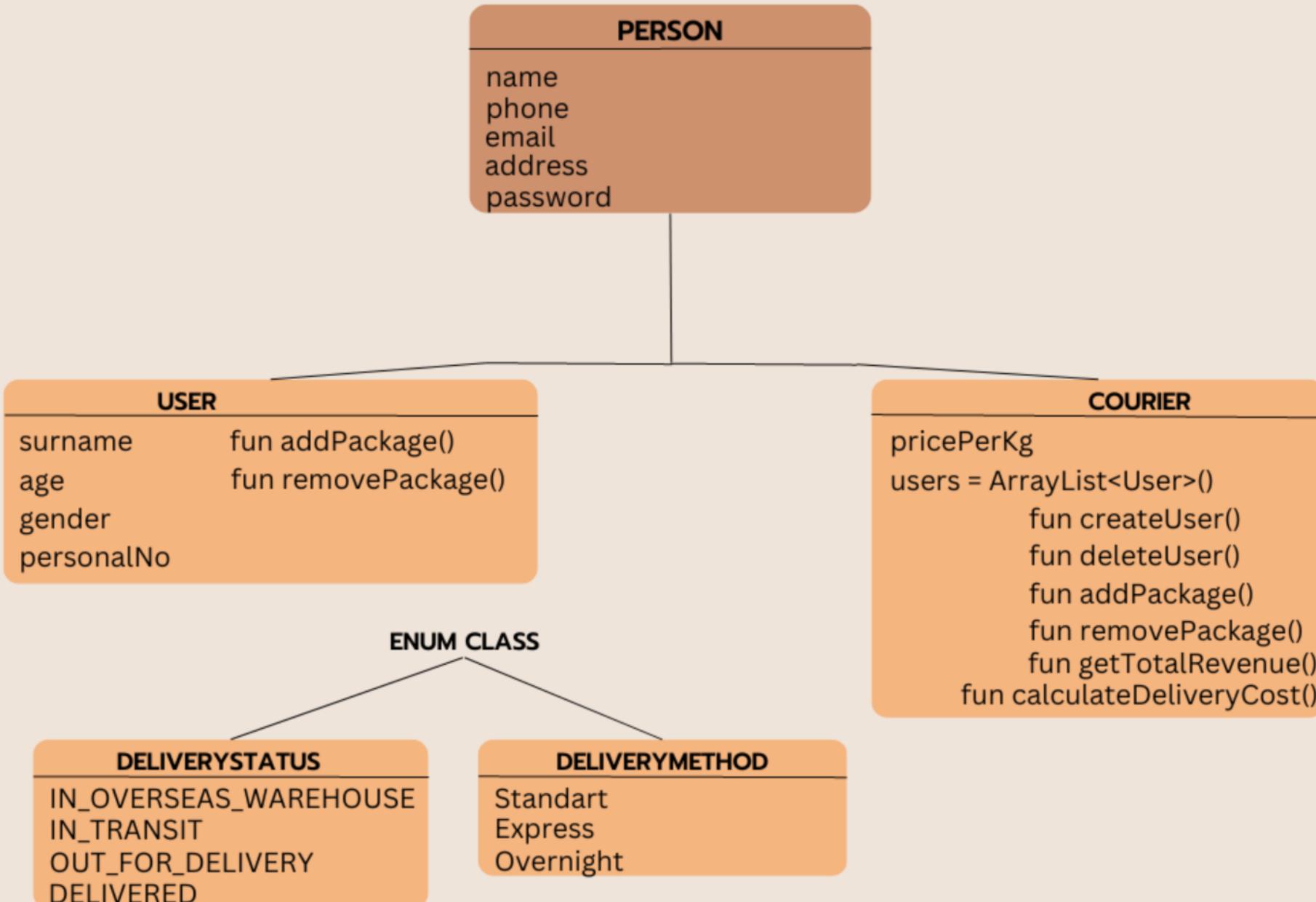
Courier APP

Elton Satiyev

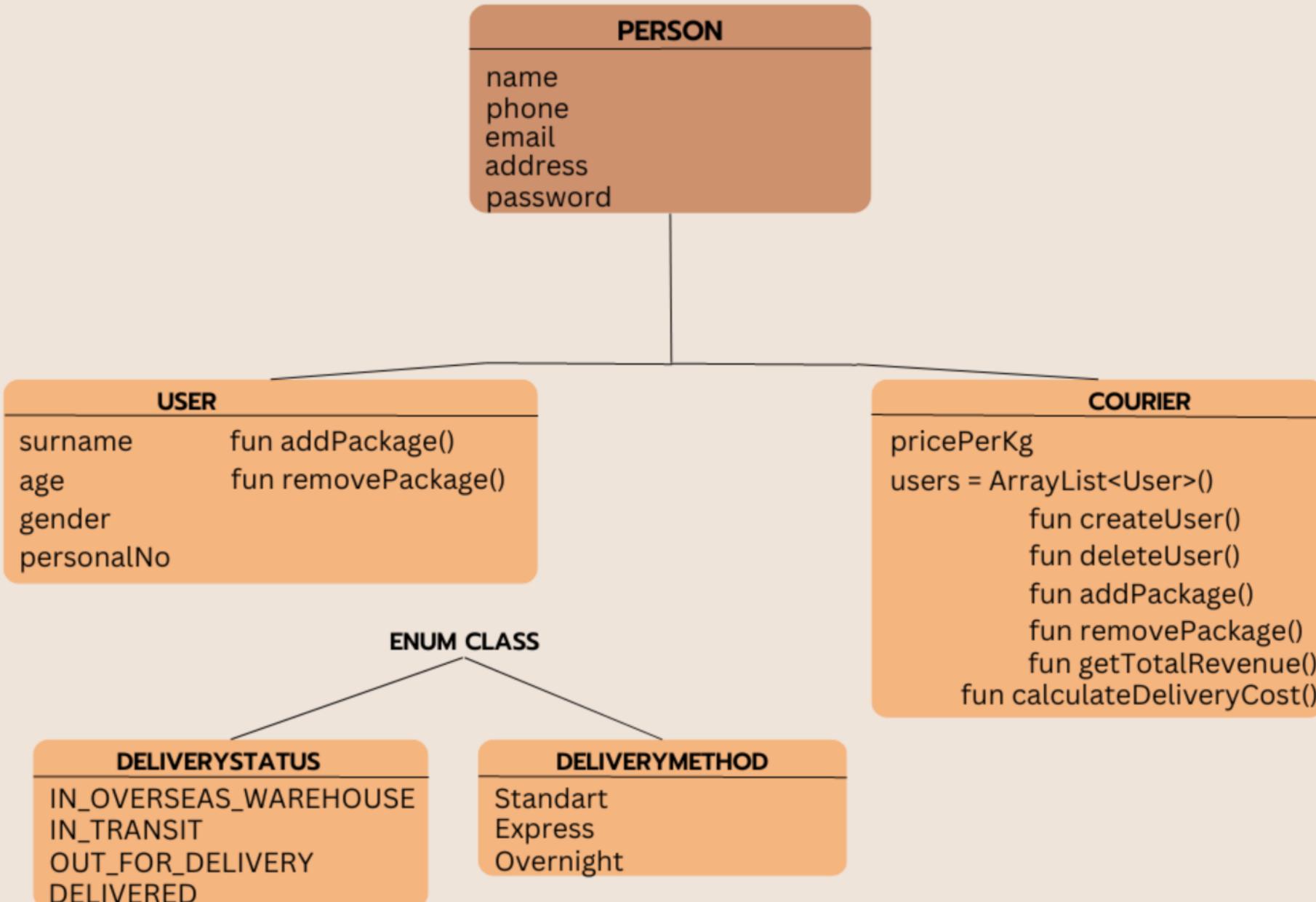


Elton Satiyev

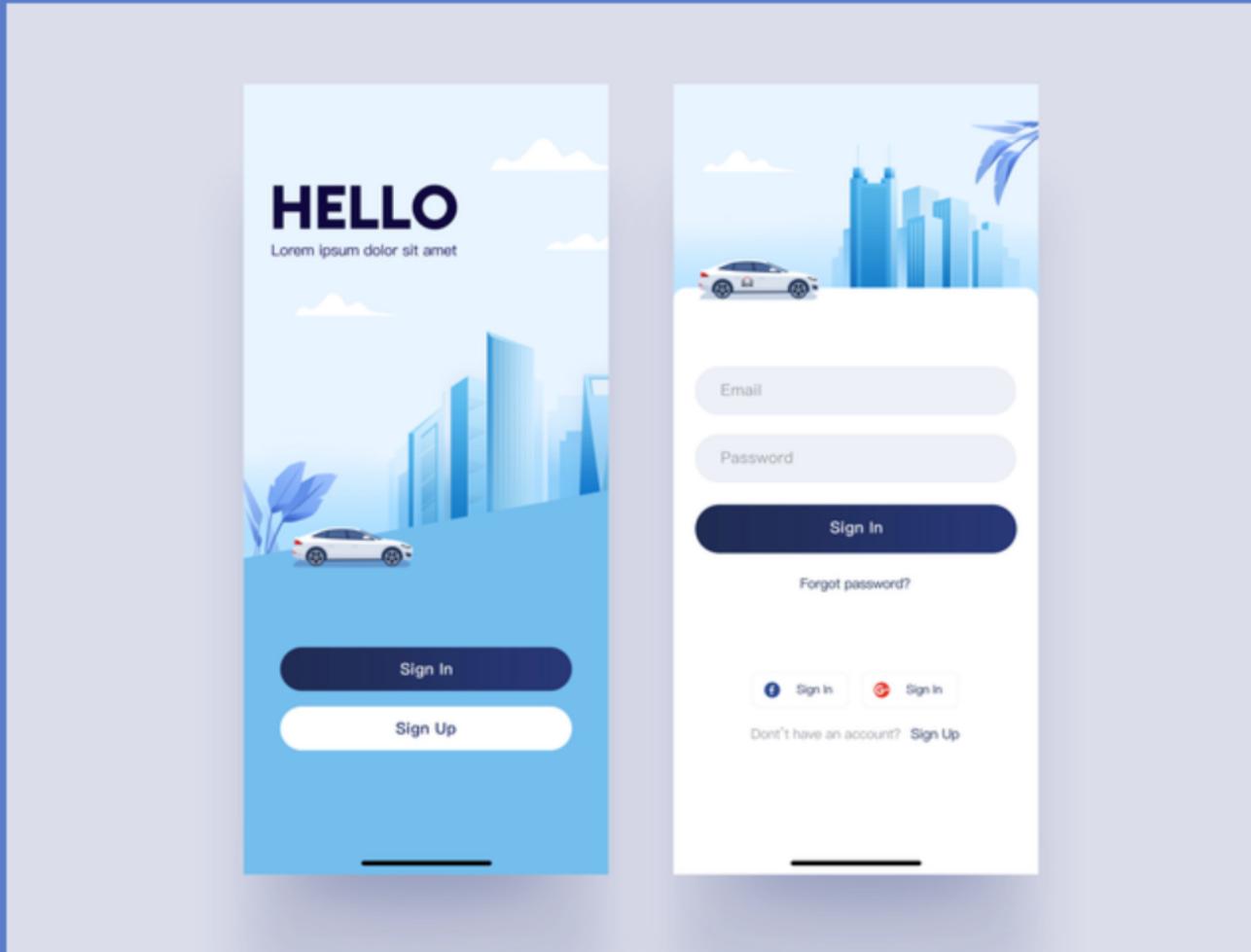
Courier APP



Courier APP

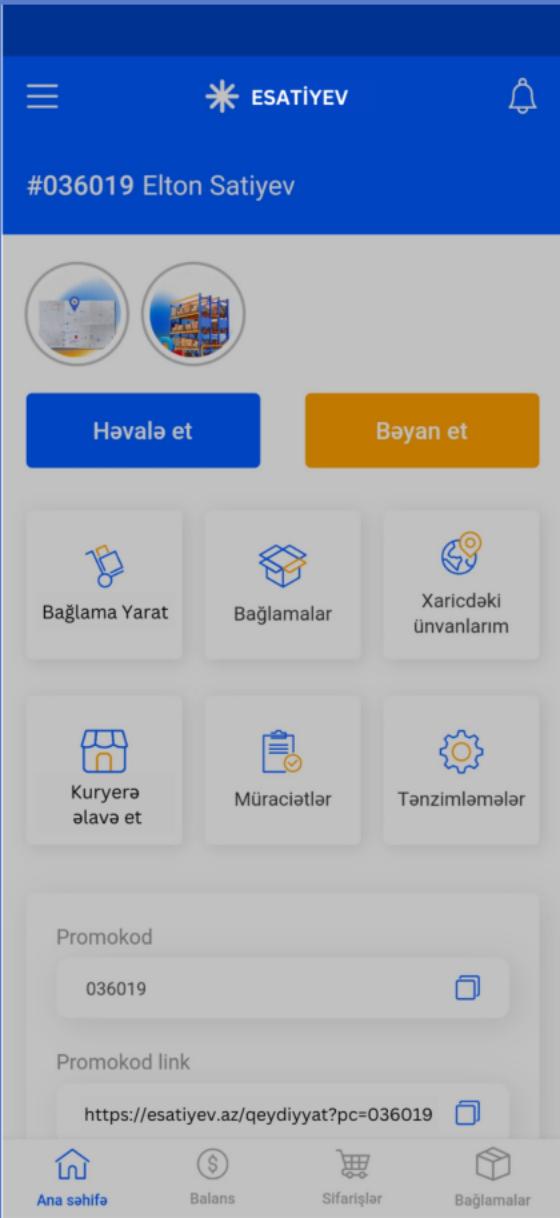


Login & Register Interface



```
C:\Users\HP\.jdks\openjdk-20.0.1\bin\  
1. Log into app as user  
2. Log into app as admin  
3. Log into app as courier  
Enter a number: 1  
1. Log in  
2. Register  
0. Back  
Enter a number: 1
```

User Main Menu



My Packages

1. Add package to specific courier
 2. Remove the package from current courier
 3. Delete the package permanently
 4. Show package information
 0. Back
- Enter a number:

Hi, Elton. You are already login.

1. Create New Package
2. My Packages
3. Profile
4. Calculator & Tariffs
0. Sign out

Enter a number: 1

Profile

Name: Elton
Surname: Satiyev
E-mail: a
Phone: 0100
Gender: Male
Address: a
Personal No: 82tt

1. Change the password
2. Change the phone number
3. Change the address
0. Back

Enter a number:

Courier Menu

Main Menu

Name: *usps*

Password: *1*

Hi, *usps*. You are already login.

1. Package Information

2. Package Operations

3. Get Total Revenue

0. Sign Out

Enter a number:

Package Information

1. Show all packages:

2. Show a package

0. Back

Enter a number:

Package Operations

1. Update Package Step

2. Deliver the package

0. Back

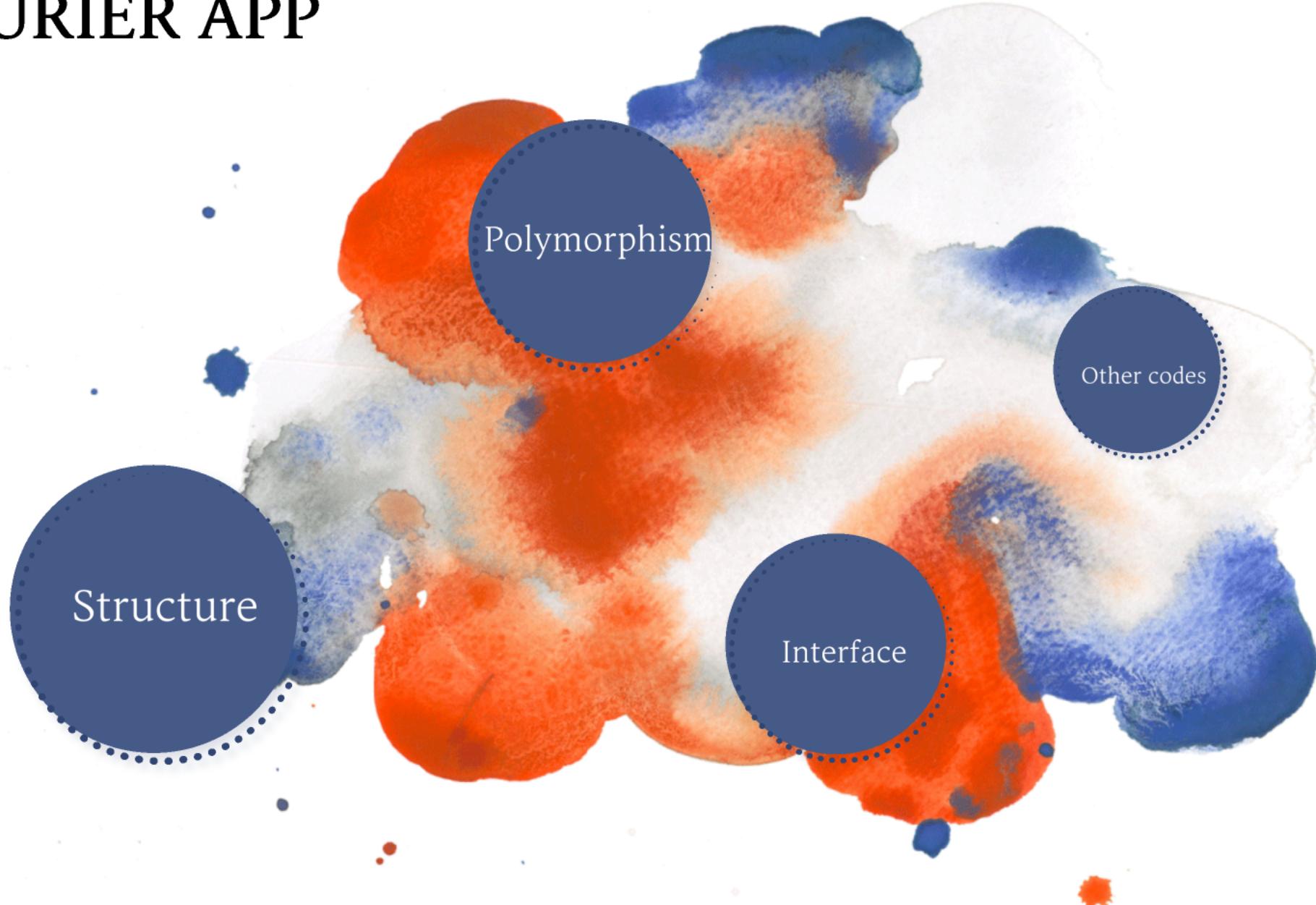
Enter a number:

Admin Menu

Main Menu

- 1. Create a new courier
 - 2. Delete the courier permanently
 - 3. Get total revenue of courier(s)
 - 4. Package Information
 - 0. Sign out
- Enter a number:

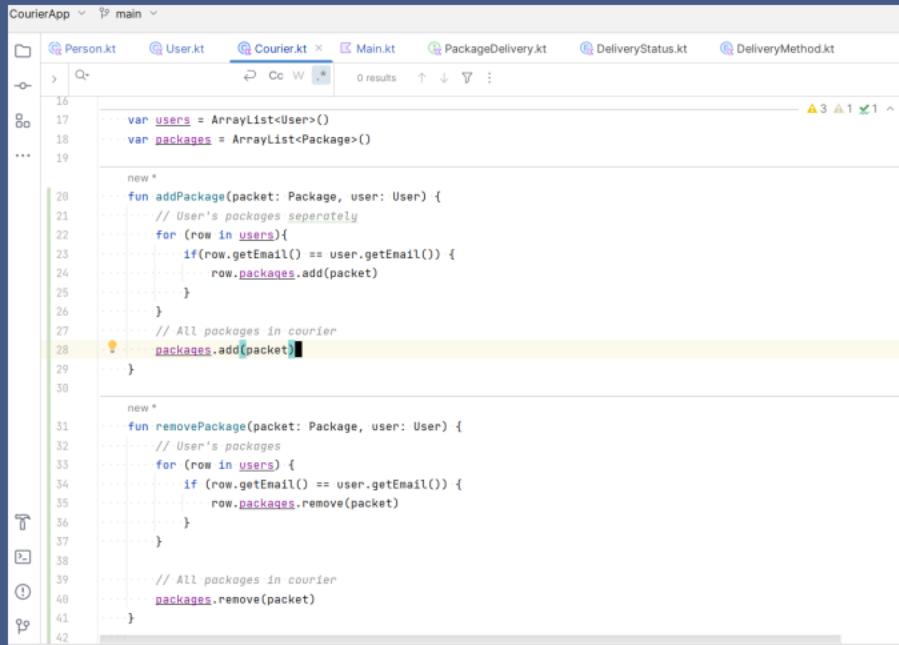
COURIER APP



엘톤

In Courier Class

In User Class



The screenshot shows a code editor window with the following code:

```
var users = ArrayList<User>()
var packages = ArrayList<Package>()

new *
fun addPackage(packet: Package, user: User) {
    // User's packages separately
    for (row in users) {
        if (row.getEmail() == user.getEmail()) {
            row.packages.add(packet)
        }
    }
    // All packages in courier
    packages.add(packet)
}

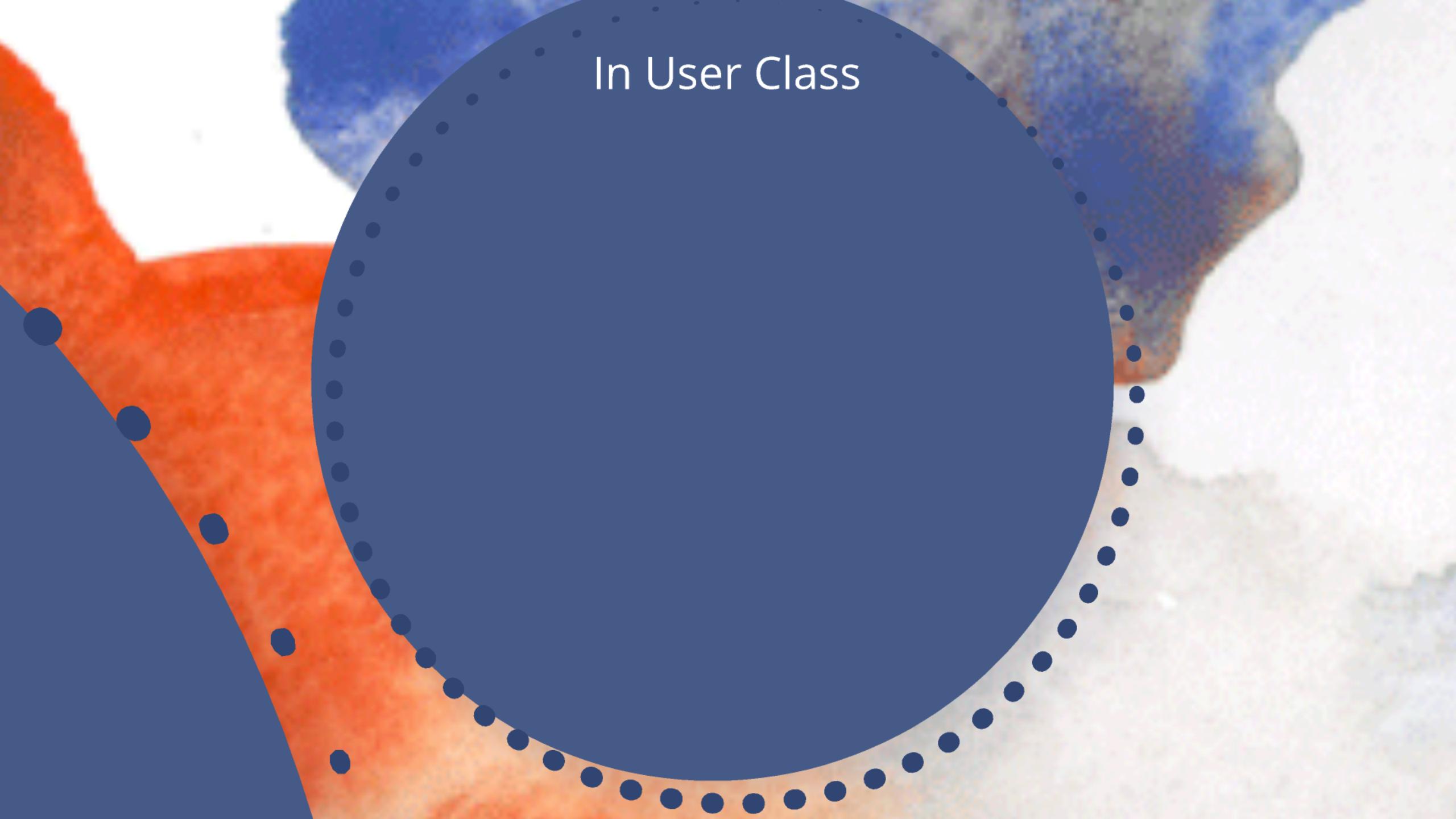
new *
fun removePackage(packet: Package, user: User) {
    // User's packages
    for (row in users) {
        if (row.getEmail() == user.getEmail()) {
            row.packages.remove(packet)
        }
    }
    // All packages in courier
    packages.remove(packet)
}
```

The code defines two methods: `addPackage` and `removePackage`. Both methods iterate through a list of `User` objects. If a user's email matches the provided `user`, the method adds or removes the `packet` from the user's list of `packages`. After processing all users, it adds or removes the `packet` from a global list of `packages`.

The screenshot shows a Java IDE interface with the following details:

- Project Structure:** On the left, there's a tree view showing files: Person.kt, User.kt, Courier.kt (which is the active tab), Main.kt, PackageDelivery.kt, DeliveryStatus.kt, and DeliveryMethod.kt.
- Search Bar:** At the top center, there's a search bar with a magnifying glass icon and a dropdown arrow, followed by buttons for Cc, W, and *.
- Status Bar:** At the top right, it says "0 results" and has navigation icons (up, down, filter, etc.).
- Code Editor:** The main area contains the following Kotlin code:

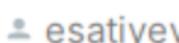
```
16  
17     var users = ArrayList<User>()  
18     var packages = ArrayList<Package>()  
19  
20     new *  
21     fun addPackage(packet: Package, user: User) {  
22         // User's packages separately  
23         for (row in users){  
24             if(row.getEmail() == user.getEmail()){  
25                 row.packages.add(packet)  
26             }  
27         }  
28         // All packages in courier  
29         packages.add(packet)  
30     }  
31  
32     new *  
33     fun removePackage(packet: Package, user: User) {  
34         // User's packages  
35         for (row in users) {  
36             if (row.getEmail() == user.getEmail()) {  
37                 row.packages.remove(packet)  
38             }  
39         }  
40         // All packages in courier  
41         packages.remove(packet)
```
- Annotations:** A yellow lightbulb icon is positioned above the line "packages.add(packet)" at line 28.
- Toolbars:** On the far left, there are three small icons: a wrench, a square with a minus sign, and a circle with an exclamation mark.
- Right Margin:** At the top right, there are three yellow warning icons with the number "3".



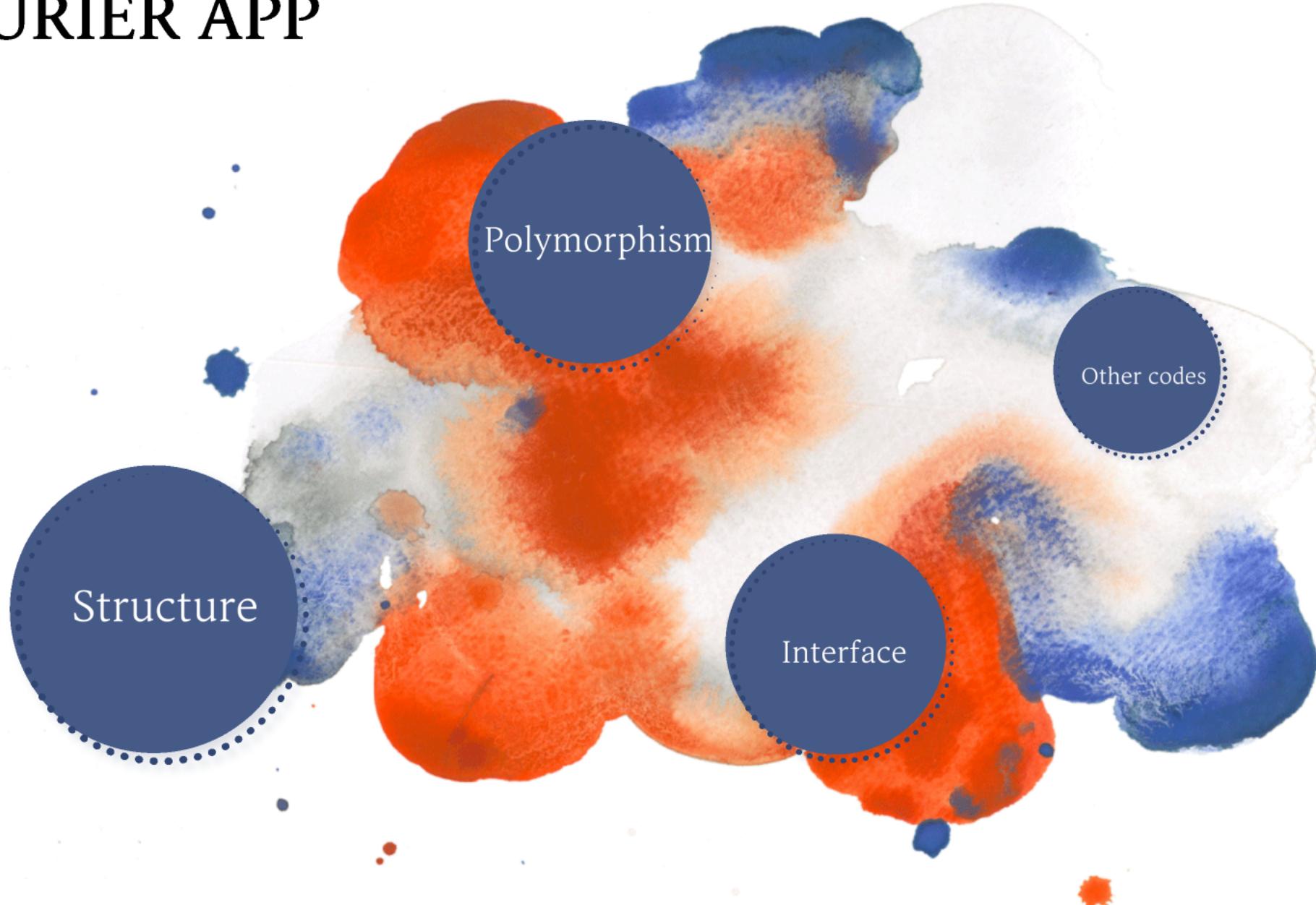
In User Class

In User Class

```
 1  class User(
 2    name: String,
 3    private var surname: String,
 4    private var age: Int,
 5    private var gender: String,
 6    phone: String,
 7    email: String,
 8    address: String,
 9    private var personalNo: String,
10    password: String
11  ) : Person(name, phone, email, password, address) {
12
13    var packages = ArrayList<Package>()
14
15    fun addPackage(packet: Package) {
16        packages.add(packet)
17    }
18
19    fun removePackage(packet: Package) {
20        packages.remove(packet)
21    }
```

```
1 class User{  
2     ... name: String,  
3     ... private var surname: String,  
4     ... private var age: Int,  
5     ... private var gender: String,  
6     ... phone: String,  
7     ... email: String,  
8     ... address: String,  
9     ... private var personalNo: String,  
10    ... password: String  
11    ) : Person(name, phone, email, password, address) {  
12  
13    ... var packages = ArrayList<Package>()  
14    ...   
15    ... fun addPackage(packet: Package) {  
16        ... packages.add(packet)  
17        ...   
18    ... fun removePackage(packet: Package) {  
19        ... packages.remove(packet)  
20    ... }
```

COURIER APP



엘톤

Interface

```
interface PackageDelivery {  
    ... esatiyev  
    fun getEstimatedDeliveryTime(): String  
    ... esatiyev  
    fun deliverPackage()  
}
```

PackageDelivery

```
interface PackageDelivery {  
    ... esatiyev  
    ...  fun getEstimatedDeliveryTime(): String  
    ... esatiyev  
    ... fun deliverPackage()  
}
```

PackageDelivery

```
override fun getEstimatedDeliveryTime(): String{
    if (this.deliveryTime == "0"){
        val distance: Int = Random.nextInt( from: 1, until: 1000)
        val averageSpeed: Int = this.deliveryMethod.speed
        this.deliveryTime = if(distance / averageSpeed == 0 || distance / averageSpeed == 1) "1 day" else "${distance / averageSpeed} days"
    }

    var days: Long = 0
    var s: String = ""
    for (i in deliveryTime) {
        if(i.isDigit()) {
            s += i
        }
    }

    / if (i != deliveryTime[deliveryTime.length - 1]) .../
    days = s.toLong()
    return LocalDate.now().plusDays(days).toString()
}
```

```
open class RegularPackage(
    packageName: String,
    trackingNumber: String,
    sender: String,
    recipient: String,
    weight: Float,
    price: Float,
    deliveryMethod: DeliveryMethod,
    step: DeliveryStatus = DeliveryStatus.IN_OVERSEAS_WAREHOUSE
) : Package(packageName, trackingNumber, sender, recipient, weight, price, deliveryMethod, step){}

@esatiyev
override fun deliverPackage() {
    step = DeliveryStatus.DELIVERED
    println("Package delivered on ${LocalDate.now()}")
}
```

Average Delivery

```
override fun getEstimatedDeliveryTime(): String{
    if (this.deliveryTime == "0"){
        val distance: Int = Random.nextInt( from: 1, until: 1000)
        val averageSpeed: Int = this.deliveryMethod.speed
        this.deliveryTime = if(distance / averageSpeed == 0 || distance / averageSpeed == 1) "1 day" else "${distance / averageSpeed} days"
    }

    var days: Long = 0
    var s: String = ""
    for (i in deliveryTime) {
        if(i.isDigit()) {
            s += i
        }
    }

    / if (i != deliveryTime[deliveryTime.length - 1]) .../
}

days = s.toLong()
return LocalDate.now().plusDays(days).toString()
```

```
open class RegularPackage(
```

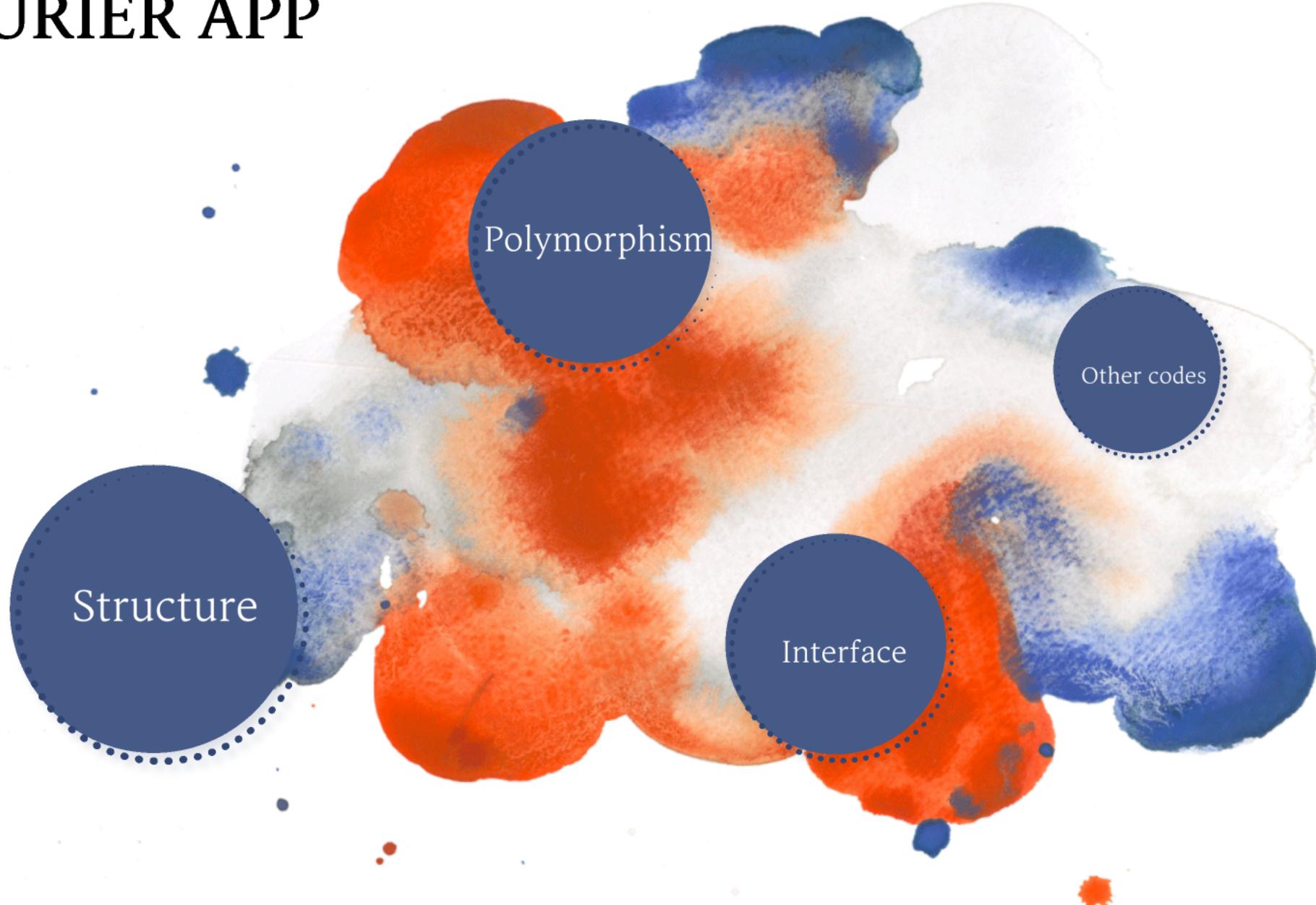
```
open class RegularPackage(  
    packageName: String,  
    trackingNumber: String,  
    sender: String,  
    recipient: String,  
    weight: Float,  
    price: Float,  
    deliveryMethod: DeliveryMethod,  
    step: DeliveryStatus = DeliveryStatus.IN_OVERSEAS_WAREHOUSE  
) : Package(packageName, trackingNumber, sender, recipient, weight, price, deliveryMethod, step){
```

✉ esatiyev

```
    override fun deliverPackage() {  
        step = DeliveryStatus.DELIVERED  
        println("Package delivered on ${LocalDate.now()}")  
    }
```

}

COURIER APP



엘톤

Deliver Package

Delivery Time

```
// Update Package Step
1 -> {
    i = 1

    for(col in COURIER.packages) {
        if(packageOption == i) {
            // check that previous step is DELIVERED or not
            condition = true
            if (col.getStepp().name == "DELIVERED") {
                condition = false
            }
            //
            else {
                println("Package step is updated.")
            }
            val step = col.getStepp().ordinal
            col.updatePackageStep( value: step + 1)
            println("Current step: ${col.getStepp().name}")

            // update delivery time
            if (col.getStepp().name == "DELIVERED" && condition) {
                col.setDeliveryTimeFormatted(LocalDate.now().toString())
            }
            break
        }
        i++
    }
} // Update Package Step
```

```
// Deliver Package
2 -> {
    i = 1

    for(col in COURIER.packages) {
        if(packageOption == i) {
            // check that previous step is DELIVERED or not
            if (col.getStepp().name == "DELIVERED") {
                println("This package is already delivered!")
            }
            //
            else {
                col.deliverPackage()
                println("Package is delivered.")
                println("Current step: ${col.getStepp().name}")
                // update delivery time
                col.setDeliveryTimeFormatted(LocalDate.now().toString())
            }
            break
        }
        i++
    }
} // Deliver Package
```

```
A esufye
fun updatePackageStep(value: Int){
    val statusValues = DeliveryStatus.values()
    if (value < statusValues.size) {
        this.step = statusValues.elementAt(value)
    }
    else {
        println("This package is already delivered.")
    }
}

A esufye
fun trackPackage() : Unit = println("Package with tracking number $trackingNumber is currently $step.")
```

```
override fun deliverPackage() {
    step = DeliveryStatus.DELIVERED
    println("Package delivered on ${LocalDate.now()}")
}
```

ENUM

Rate

Enum Class

```
enum class DeliveryMethod(var speed: Int) {  
    Standart(speed: 100),  
    Express(speed: 300),  
    Overnight(speed: 500)  
}
```

```
num class DeliveryStatus(val status: String, val value: Int) {  
    IN_OVERSEAS_WAREHOUSE(status: "The package is at the warehouse.", value: 1)  
    IN_TRANSIT(status: "The package is in transit.", value: 2),  
    OUT_FOR_DELIVERY(status: "The package is out for delivery.", value: 3),  
    DELIVERED(status: "The package is delivered.", value: 4)  
}
```

```
enum class DeliveryMethod(var speed: Int) {  
     Standard(speed: 100),  
    Express(speed: 300),  
    Overnight(speed: 500)  
}
```

```
enum class DeliveryStatus(val status: String, val value: Int) {  
    IN_OVERSEAS_WAREHOUSE(status: "The package is at the warehouse.", value: 1),  
    IN_TRANSIT(status: "The package is in transit.", value: 2),  
    OUT_FOR_DELIVERY(status: "The package is out for delivery.", value: 3),  
    DELIVERED(status: "The package is delivered.", value: 4)
```

Rate

```
0070 - fun rateDelivery(trackingNumber: String, email: String, couriers: ArrayList<Courier>): ArrayList<Courier> {
0071     var option: Int = 0
0072     for (item in couriers) {
0073         // couriers.users.packages rate
0074         for (col in item.users) {
0075             if (col.getEmail() == email) {
0076                 for (th in col.packages) {
0077                     if (th.getTrackingNumber() == trackingNumber) {
0078                         if (th.getPackageName() == "A") {
0079                             println("Package: " + th.getPackageName() + " - " + th.getTrackingNumber())
0080                             println("Ratings as far better service!")
0081                             println("1. 1 points 🌟")
0082                             println("2. 2 points 🌟")
0083                             println("3. 3 points 🌟")
0084                             println("4. 4 points 🌟")
0085                             println("5. 5 points 🌟")
0086                         }
0087                         rating = ratingNumber(1, 5)
0088                         println("Thanks for rating!")
0089                         th.setRating(rating)
0090                     }
0091                 }
0092             // couriers.packages rate
0093             for (th in item.packages) {
0094                 if (th.getTrackingNumber() == trackingNumber) {
0095                     col.setRating(option)
0096                 }
0097             }
0098         }
0099     }
0100 }
```

```
// User Interface
1 -> {
    USER = users[uNumber]

    var userInterface = true
    while (userInterface) {
        for (row in couriers) {
            for (col in row.users) {
                if (col.getEmail() == USER.getEmail()) {
                    for (th in col.packages) {
                        if (th.getStepp().name == "DELIVERED" && th.getRating() == 0) {
                            couriers = rateDelivery(th.getTrackingNumber(), USER.getEmail(), couriers)
                            for (i in USER.packages) {
                                if (i.getTrackingNumber() == th.getTrackingNumber()) {
                                    i.setRating(th.getRating())
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
```

```
// User Interface  
1 -> {  
    ..USER = users[uNumber]  
  
    var userInterface = true  
    while (userInterface) {  
        for (row in couriers) {  
            for (col in row.users) {  
                if (col.getEmail() == USER.getEmail()) {  
                    for (th in col.packages) {  
                        if (th.getStep().name == "DELIVERED" && th.getRating() == 0) {  
                            couriers = rateDelivery(th.getTrackingNumber(), USER.getEmail(), couriers)  
                            for (i in USER.packages) {  
                                if (i.getTrackingNumber() == th.getTrackingNumber()) {  
                                    i.setRating(th.getRating())  
                                }  
                            }  
                        }  
                    }  
                }  
            }  
        }  
    }  
}
```

```
1070    fun rateDelivery(trackingNumber: String, email: String, couriers: ArrayList<Courier>): ArrayList<Courier> {
1071        var option: Int = 0
1072        for (row in couriers) {
1073            // couriers.users.packages.rate
1074            for (col in row.users) {
1075                if(col.getEmail() == email) {
1076                    for (th in col.packages) {
1077                        if(th.getTrackingNumber() == trackingNumber) {
1078                            println("Package: " + th.getPackageName() + " - " + th.getTrackingNumber())
1079                            println("Rate us for better service:")
1080                            println("1. 1 point 😐")
1081                            println("2. 2 points 😐")
1082                            println("3. 3 points 😒")
1083                            println("4. 4 points 😊")
1084                            println("5. 5 points 😃")
1085
1086                            option = enterNumber( x: 1, y: 5)
1087                            println("Thanks for rating!")
1088                            th.setRating(option)
1089                        }
1090                    }
1091                }
1092            }
1093            // couriers.packages.rate
1094            for (col in row.packages) {
1095                if(col.getTrackingNumber() == trackingNumber) {
1096                    col.setRating(option)
1097                }
1098            }
1099        }
1100    }
```

Delivery Time and others

```
seconder@for (row in couriers) {  
    for (col in row.users)  
        if (col.getEmail() == USER.getEmail()) {  
            for (th in col.packages) {  
                if (th.getTrackingNumber() == USER.packages[option-1].getTrackingNumber()) {  
                    println("Package : ${th.getPackageName()}")  
                    println("Recipient: ${th.getRecipient()}")  
                    println("Sender: ${th.getSender()}")  
                    println("Price: ${th.getPrice()}")  
                    println("Weight: ${th.getWeight()}")  
                    println("Delivery cost: ${row.calculateDeliveryCost(th.getTrackingNumber())}")  
                    println("Courier: ${row.getName()}")  
                    println("Tracking number: ${th.getTrackingNumber()}")  
                    if (th.getStep().name == "DELIVERED")  
                        println("Delivery Time: ${th.getDeliveryTimeFormatted()}")  
                    else  
                        println("Estimated Delivery Time: ${th.getEstimatedDeliveryTime()}")  
                    println()  
                    println("Delivery Status: " + th.getStep().name)  
  
                    if (th is FragilePackage) {  
                        println("Is it Fragile: " + th.getIsFragile())  
                    }  
                    else if (th is PerishablePackage) {  
                        println("Is it Perishable: " + th.getIsPerishable())  
                        println("Production date: " + th.getProductionDate())  
                        println("Shelf life: " + th.getShelfLife())  
                    }  
                    else if (th is HazardousPackage) {  
                        println("Is it Hazardous: " + th.getIsHazardous())  
                        println("Hazardous type: " + th.getHazardousType())  
                    }  
                }  
            }  
        }  
    }  
}
```

```
for (col in row.users) {
    if (col.getEmail() == USER.getEmail()) {
        for (th in col.packages) {
            if (th.getTrackingNumber() == USER.packages[option-1].getTrackingNumber()) {
                println("Package : ${th.getPackageName()}")
                println("Recipient: ${th.getRecipient()}")
                println("Sender: ${th.getSender()}")
                println("Price: ${th.getPrice()}")
                println("Weight: ${th.getWeight()}")
                println("Delivery cost: ${row.calculateDeliveryCost(th.getTrackingNumber())}")
                println("Courier: ${row.getName()}")
                println("Tracking number: ${th.getTrackingNumber()}")
                if (th.getStepp().name == "DELIVERED")
                    println("Delivery Time: ${th.getDeliveryTimeFormatted()}")
                else
                    println("Estimated Delivery Time: ${th.getEstimatedDeliveryTime()}")
                println()
                println("Delivery Status: " + th.getStepp().name)
            }
            if (th is FragilePackage) {
                println("Is it Fragile: " + th.getIsFragile())
            }
            else if (th is PerishablePackage) {
                println("Is it Perishable: " + th.getIsPerishable())
                println("Production date: " + th.getProductionDate())
                println("Shelf life: " + th.getShelfLife())
            }
        }
    }
}
```

esatiyev

```
fun updatePackageStep(value: Int){  
    val statusValues = DeliveryStatus.values()  
    if (value < statusValues.size) {  
        this.step = statusValues.elementAt(value)  
    }  
    else {  
        println("This package is already delivered.")  
    }  
}
```

esatiyev

```
fun trackPackage(): Unit = println("Package with tracking number $trackingNumber is currently $step.")
```

```
// Update Package Step
1 -> {
    i = 1

    for(col in COURIER.packages) {
        if(packageOption == i) {
            // check that previous step is DELIVERED or not
            condition = true
            if (col.getStepp().name == "DELIVERED") {
                condition = false
            }
            //
            else {
                println("Package step is updated.")
            }
            val step = col.getStepp().ordinal
            col.updatePackageStep( value: step + 1)
            println("Current step: ${col.getStepp().name}")

            // update delivery time
            if (col.getStepp().name == "DELIVERED" && condition) {
                col.setDeliveryTimeFormatted(LocalDate.now().toString())
            }
            break
        }
        i++
    }
} // Update Package Step
```

```
// Deliver Package
2 -> {
    i = 1

    for(col in COURIER.packages) {
        if(packageOption == i) {
            // check that previous step is DELIVERED or not
            if (col.getStepp().name == "DELIVERED") {
                println("Delivery successful")
            }
            else {
                col.deliveryStatus = "PENDING"
                println("Delivery pending")
                println("Reason: ${col.getStepp().name}")
                // update delivery time
                col.setDeliveryTimeFormatted(LocalDate.now().toString())
            }
            break
        }
        i++
    }
} // Deliver Package
```

Deliver Package

```
override fun deliverPackage() {
    step = DeliveryStatus.DELIVERED
    println("Package delivered on ${LocalDate.now()}")
}
```

or not
{
name}")

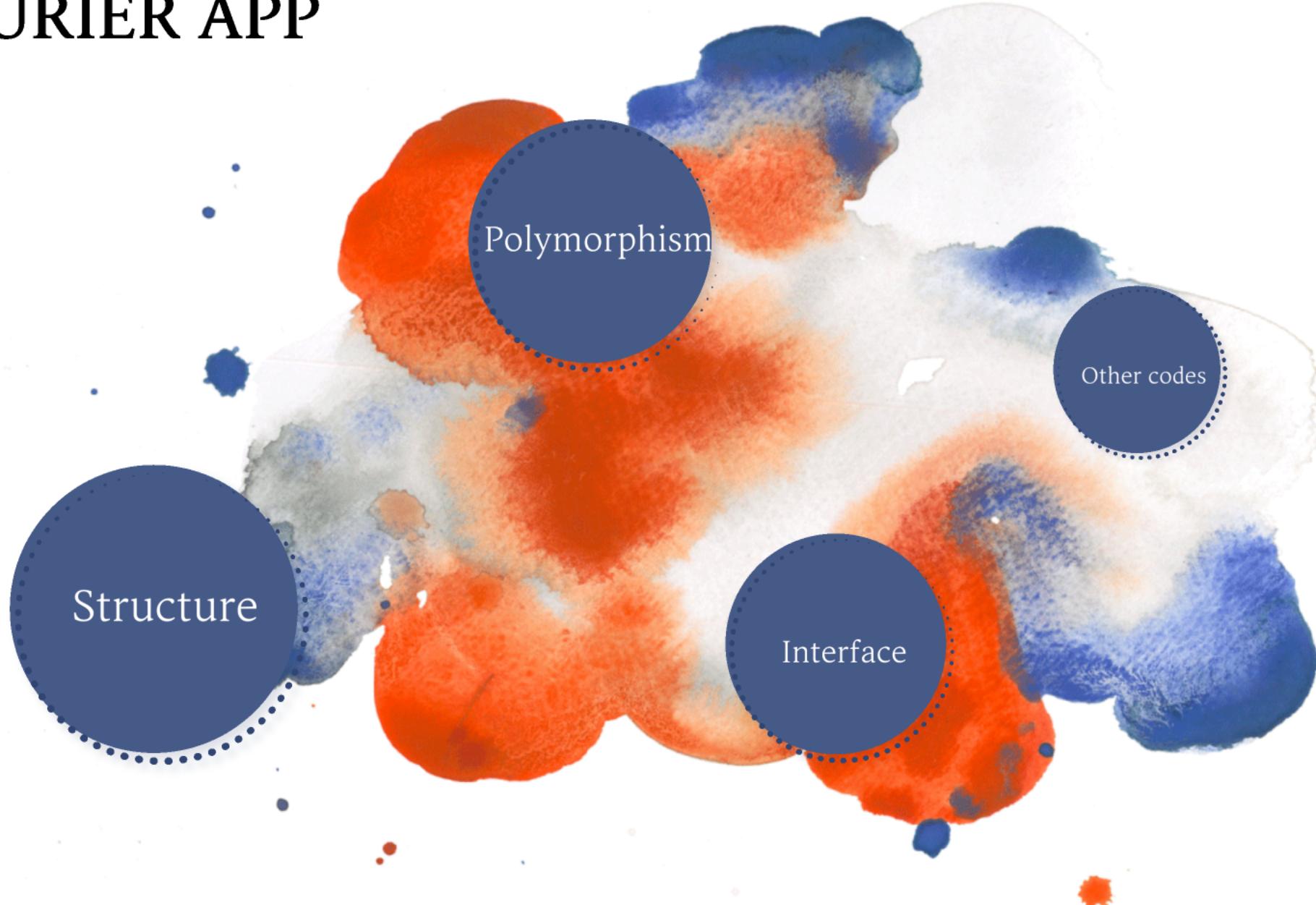
& condition) {
te.now().toString())

```
// Deliver Package
2 -> {
    i = 1

    for(col in COURIER.packages) {
        if(packageOption == i) {
            // check that previous step is DELIVERED or not
            if (col.getStepp().name == "DELIVERED") {
                println("This package is already delivered!")
            }
            //

            else {
                col.deliverPackage()
                println("Package is delivered.")
                println("Current step: ${col.getStepp().name}")
                // update delivery time
                col.setDeliveryTimeFormatted(LocalDate.now().toString())
            }
            break
        }
        i++
    }
} // Deliver Package
```

COURIER APP



엘톤