

## Практическая работа №2 Знакомство с Dart и Flutter. ООП.

Красноперов Дмитрий еозИСИП-207(307)

The screenshot shows two instances of the Visual Studio Code (VS Code) interface, both displaying Dart code. The top instance shows code for function overloading, and the bottom instance shows code for inheritance.

**Top Instance (Function Overloading):**

```
Функции высшего порядка + ООП.dart
import 'dart:math';

class MathProcessor {
  List<double> numbers;

  MathProcessor(this.numbers);

  List<double> process(double Function(double) processor) {
    return numbers.map(processor).toList();
  }

  static List<double> getRandomNumbers(int count, double min, double max) {
    Random random = Random();
    List<double> randomNumbers = [];
    for (int i = 0; i < count; i++) {
      randomNumbers.add(min + (max - min) * random.nextDouble());
    }
    return randomNumbers;
  }

  void printNumbers() {
    print(numbers.toString());
  }
}

@Override
```

**Bottom Instance (Inheritance):**

```
Abstract Classes + Inheritance.dart
import 'dart:math';

abstract class Shape {
  double area();
}

factory Shape.fromJson(Map<String, dynamic> json) {
  final String type = json['type'];
  switch (type) {
    case 'circle':
      final double radius = json['radius'] as double;
      return Circle(radius);
    case 'rectangle':
      final double width = json['width'] as double;
      final double height = json['height'] as double;
      return Rectangle(width, height);
    default:
      throw ArgumentError('Unknown shape type: $type');
  }
}

class Circle implements Shape {
  final double radius;
  Circle(this.radius);

  @override
  double area() {
    return pi * radius * radius;
  }
}
```

Файл Правка Выделение Вид Переход ... ← → Q.dart

ПРОВОДНИК ... Функции высшего порядка + ООП.dart Abstract Classes + Inheritance.dart

ОТКРЫТЫЕ РЕДАКТОРЫ Функции высшего... Abstract Classes + ... DART vscode Функции высшего п... launch.json settings.json Калькулятор матриц... наследование.dart Несимметричные преобразования.dart Переопределение методов.dart Переопределение операторов.dart Переисключения.dart Подсчёт гласных.dart Расширения классов.dart Сочетание миксинов.dart Умный калькулятор ... Abstract Classes + Inh... abstract.dart generic.dart get.dart interface.dart late.dart mixin.dart null class.dart proba.dart probaz.dart proverka.dart super.dart superz.dart user.dart

25 class Circle implements Shape {  
26 final double radius;  
27  
28 Circle(this.radius);  
29  
30 @override  
31 double area() {  
32 return pi \* radius \* radius;  
33 }  
34  
35 @override  
36 String toString() {  
37 return 'Circle(radius: \$radius, area: \${area()})';  
38 }  
39 }  
40  
41 class Rectangle implements Shape {  
42 final double width;  
43 final double height;  
44  
45 Rectangle(this.width, this.height);  
46  
47 @override  
48 double area() {  
49 return width \* height;  
50 }  
51  
52 @override  
53 String toString() {  
54 return 'Rectangle(width: \$width, height: \$height, area: \${area()})';  
55 }

ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ РЕЗУЛЬТАТЫ ЗАПРОСА

Connecting to VM Service at ws://127.0.0.1:13934/oaA-greyYaQ=/ws  
Connected to the VM Service.  
Circle(radius: 5.0, area: 78.53981633974483)  
Rectangle(width: 4.0, height: 6.0, area: 24.0)  
Error creating shape: Invalid argument(s): Unknown shape type: triangle

Exited.

Строка 6, столбец 2 Пробелов: 2 UTF-8 CRLF Dart

The screenshot shows the Dart IDE interface with the following details:

- File Explorer (Left):** Shows various Dart files and assets like launch.json and settings.json.
- Editor (Top):** Displays the code for `Abstract Classes + Inheritance.dart`. The cursor is at the start of the `abstract` keyword. A tooltip "Функции высшего порядка + ООП.dart" is visible above the editor area.
- Bottom Status Bar:** Includes tabs for PROBLEMS, ВХОДИЩИЕ ДАННЫЕ (INPUT DATA), КОНСОЛЬ ОТЛАДКИ (DEBUG CONSOLE), ТЕРМИНАЛ (TERMINAL), ПОРТЫ (PORTS), РЕЗУЛЬТАТЫ ЗАПРОСА (REQUEST RESULTS), and a search bar for "Файл (например: text, json...)".
- Terminal (Bottom):** Shows log output:

```
Connecting to VM Service at ws://127.0.0.1:13934/oaA-greyYqA==ws
Connected to the VM Service.
Circle(radius: 5.0, area: 78.53981633974483)
Rectangle(width: 4.0, height: 6.0, area: 24.0)
Error creating shape: Invalid argument(s): Unknown shape type: triangle
Exited.
```

Файл Правка Выделение Вид Переход ... ⏪ ⏩ ⏴ ⏵ Q.dart

ПРОВОДНИК

- ОТКРЫТЫЕ РЕДАКТОРЫ
  - Функции высшего...
  - Abstract Classes + ...
  - Static methods + C...
- DART
  - .vscode
    - launch.json
    - settings.json
  - Калькулятор матриц...
  - наследование.dart
  - Нейвные преобразов...
  - Переопределение ми...
  - Переопределение он...
  - Перечисления.dart
  - Подсёт гласных.dart
  - Расширения класса.d...
  - Сочетание миксинов...
  - Умный калькулятор.c...
- Abstract Classes + In...
- abstract.dart
- generics.dart
- get.dart
- interface.dart
- late.dart
- mixin.dart
- null class.dart
- proba.dart
- probad.dart
- proverka.dart
- Static methods + Cons...
- super.dart
- superz.dart
- user.dart

> СТРУКТУРА

> ВРЕМЕННАЯ ШКАЛА

> DEPENDENCIES

Функции высшего порядка + ООП.dart Abstract Classes + Inheritance.dart Static methods + Constants.dart

```
Static methods + Constants.dart @ main
1 class PasswordValidator {
2     static const int minLength = 8;
3     static const String digitRegex = r'[0-9]';
4     static bool validate(String password) {
5         if (password.length < minLength) {
6             print('Пароль должен содержать не менее $minLength символов.');
7             return false;
8         }
9         if (!RegExp(digitRegex).hasMatch(password)) {
10             print('Пароль должен содержать хотя бы одну цифру.');
11             return false;
12         }
13         return true;
14     }
15     PasswordValidator._();
16 }
Run | Debug
void main() {
18     String password = "DimaPassword45";
19     if (PasswordValidator.validate(password)) {
20         print("Пароль валидный.");
21     } else {
22         print("Пароль не прошел валидацию.");
23     }
24
25     password = "Dima";
26     if (PasswordValidator.validate(password)) {
27         print("Пароль валидный.");
28     } else {
29         print("Пароль не прошел валидацию.");
30     }
31 }
```

ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ РЕЗУЛЬТАТЫ ЗАПРОСА

Фильтр (например: text, lex...)

Connecting to VM Service at ws://127.0.0.1:2783/22cklroKv1g-/ws  
Connected to the VM Service.  
Пароль валидный.  
Пароль должен содержать не менее 8 символов.  
Пароль не прошел валидацию.

Exited.

Строка 18, столбец 36 Пробелов: 2 UTF-8 CRLF ⏪ Dart

Файл Правка Выделение Вид Переход ... ⏪ ⏩ ⏴ ⏵ Q.dart

ПРОВОДНИК

- ОТКРЫТЫЕ РЕДАКТОРЫ
  - Функции высшего...
  - Abstract Classes + ...
  - Static methods + C...
  - Banking system.dart
- DART
  - .vscode
    - launch.json
    - settings.json
  - Калькулятор матриц...
  - наследование.dart
  - Нейвные преобразов...
  - Переопределение ми...
  - Переопределение он...
  - Перечисления.dart
  - Подсёт гласных.dart
  - Расширения класса.d...
  - Сочетание миксинов...
  - Умный калькулятор.c...
- Abstract Classes + In...
- abstract.dart
- Banking system.dart
- generics.dart
- get.dart
- interface.dart
- late.dart
- mixin.dart
- null class.dart
- proba.dart
- probad.dart
- proverka.dart
- Static methods + Cons...
- super.dart
- superz.dart
- user.dart

> СТРУКТУРА

> ВРЕМЕННАЯ ШКАЛА

> DEPENDENCIES

Функции высшего порядка + ООП.dart Abstract Classes + Inheritance.dart Static methods + Constants.dart Banking system.dart

```
Banking system.dart @ main
1 import 'dart:math';
2 class BankAccount {
3     late String accountNumber;
4     double balance;
5     String ownerName;
6     BankAccount(this.ownerName) : balance = 0.0 {
7         accountNumber = _generateAccountNumber();
8     }
9     String _generateAccountNumber() {
10         Random random = Random();
11         String number = '';
12         for (int i = 0; i < 10; i++) {
13             number += random.nextInt(10).toString();
14         }
15         return number;
16     }
17     void deposit(double amount) {
18         if (amount <= 0) {
19             print("Сумма для пополнения должна быть положительной.");
20             return;
21         }
22         balance += amount;
23         print("Счет ${accountNumber}: Пополнено на $amount. Новый баланс: $balance");
24     }
25     void withdraw(double amount) {
26         if (amount <= 0) {
27             print("Сумма для снятия должна быть положительной.");
28             return;
29         }
30         if (amount > balance) {
31             print("Недостаточно средств на счете ${accountNumber}.");
32             return;
33         }
34     }
35 }
```

ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ РЕЗУЛЬТАТЫ ЗАПРОСА

Фильтр (например: text, lex...)

Счет 9747232247: Текущий баланс: 5000.0  
Счет 6492469143: Снято 300.0. Новый баланс: 9700.0  
Счет 9747232247: Пополнено на 300.0. Новый баланс: 5300.0  
Перевод 300.0 со счета 6492469143 на счет 9747232247 выполнен.  
Счет 6492469143: Текущий баланс: 9700.0  
Счет 9747232247: Текущий баланс: 5300.0  
Счет 6492469143: Снято 2000.0. Новый баланс: 7700.0

Строка 105, столбец 4 Пробелов: 2 UTF-8 CRLF ⏪ Dart

Файл Правка Выделение Вид Переход ... Q.dart

ПРОВОДНИК ...

ФУНКЦИИ ВЫСШЕГО ПОРЯДКА + OOP.dart Abstract Classes + Inheritance.dart Static methods + Constants.dart Banking system.dart

ОТКРЫТИЕ РЕДАКТОРЫ

Функции высшего... Abstract Classes + ... Static methods + ... Banking system.dart

DART .vscode

launch.json settings.json Калькулятор матриц... наследование... Несимметричные преобразования... Переопределение методов... Переопределение операторов... Переопределение классов... Подсчет гласных.dart Расширение класса.dart Сочетание миксинов... Умный калькулятор ... Abstract Classes + Inh... abstract.dart Banking system.dart generis.dart get.dart interface.dart late.dart mixin.dart null class.dart proba.dart proba2.dart proverka.dart Static methods + Cons... super.dart super2.dart

СТРУКТУРА > ВРЕМЕННАЯ ШКАЛА > DEPENDENCIES >

```
1 Banking system.dart > main
2   class BankAccount {
3     void withdraw(double amount) {
4       if (amount > balance) {
5         print("Недостаточно средств на счете ${accountNumber}.");
6         return;
7       }
8       balance -= amount;
9       print("Счет ${accountNumber}: снято $amount. Новый баланс: $balance");
10    }
11    void displayBalance() {
12      print("Счет ${accountNumber}: Текущий баланс: $balance");
13    }
14    @override
15    String toString() {
16      return "BankAccount(accountNumber: $accountNumber, balance: $balance, ownerName: $ownerName)";
17    }
18  }
19  class Bank {
20    List<BankAccount> accounts = [];
21    BankAccount addAccount(String ownerName) {
22      BankAccount newAccount = BankAccount(ownerName);
23      accounts.add(newAccount);
24      print("Создан счет ${newAccount.accountNumber} для ${ownerName}.");
25      return newAccount;
26    }
27    BankAccount? findAccount(String accountNumber) {
28      for (var account in accounts) {
29        if (account.accountNumber == accountNumber) {
30          return account;
31        }
32      }
33    }
34  }
35  void transfer(String fromAccountNumber, String toAccountNumber, double amount) {
36    if (amount <= 0) {
37      print("Сумма перевода должна быть положительной.");
38      return;
39    }
40    BankAccount? fromAccount = findAccount(fromAccountNumber);
41    BankAccount? toAccount = findAccount(toAccountNumber);
42
43    if (fromAccount == null) {
44      print("Счет отправителя ${fromAccountNumber} не найден.");
45      return;
46    }
47    if (toAccount == null) {
48      print("Счет получателя ${toAccountNumber} не найден.");
49      return;
50    }
51    if (fromAccount.balance < amount) {
52      print("Недостаточно средств на счете ${fromAccountNumber} для перевода.");
53      return;
54    }
55    fromAccount.withdraw(amount);
56    toAccount.deposit(amount);
57    print("Перевод $amount со счета ${fromAccountNumber} на счет ${toAccountNumber} выполнен.");
58  }
59
60 Run | Debug
61 void main() {
62
63 }
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86 }
```

ПРОБЛЕМЫ ВХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ РЕЗУЛЬТАТЫ ЗАПРОСА Фильтр (например: text, lexed...)

Счет 9747232247: Текущий баланс: 5000.0  
Счет 6492469143: Снято 300.0. Новый баланс: 4700.0  
Счет 9747232247: Пополнено на 300.0. Новый баланс: 5300.0  
Перевод 300.0 со счета 6492469143 на счет 9747232247 выполнен.  
Счет 6492469143: Текущий баланс: 4700.0  
Счет 9747232247: Текущий баланс: 5300.0  
Счет 6492469143: Снято 2000.0. Новый баланс: 7700.0

Файл Правка Выделение Вид Переход ... Q.dart

ПРОВОДНИК ...

ФУНКЦИИ ВЫСШЕГО ПОРЯДКА + OOP.dart Abstract Classes + Inheritance.dart Static methods + Constants.dart Banking system.dart

ОТКРЫТИЕ РЕДАКТОРЫ

Функции высшего... Abstract Classes + ... Static methods + ... Banking system.dart

DART .vscode

launch.json settings.json Калькулятор матриц... наследование... Несимметричные преобразования... Переопределение методов... Переопределение операторов... Переопределение классов... Подсчет гласных.dart Расширение класса.dart Сочетание миксинов... Умный калькулятор ... Abstract Classes + Inh... abstract.dart Banking system.dart generis.dart get.dart interface.dart late.dart mixin.dart null class.dart proba.dart proba2.dart proverka.dart Static methods + Cons... super.dart super2.dart

СТРУКТУРА > ВРЕМЕННАЯ ШКАЛА > DEPENDENCIES >

```
1 Banking system.dart > main
2   class Bank {
3     BankAccount? findAccount(String accountNumber) {
4       return null;
5     }
6     void transfer(String fromAccountNumber, String toAccountNumber, double amount) {
7       if (amount <= 0) {
8         print("Сумма перевода должна быть положительной.");
9         return;
10      }
11      BankAccount? fromAccount = findAccount(fromAccountNumber);
12      BankAccount? toAccount = findAccount(toAccountNumber);
13
14      if (fromAccount == null) {
15        print("Счет отправителя ${fromAccountNumber} не найден.");
16        return;
17      }
18      if (toAccount == null) {
19        print("Счет получателя ${toAccountNumber} не найден.");
20        return;
21      }
22      if (fromAccount.balance < amount) {
23        print("Недостаточно средств на счете ${fromAccountNumber} для перевода.");
24        return;
25      }
26      fromAccount.withdraw(amount);
27      toAccount.deposit(amount);
28      print("Перевод $amount со счета ${fromAccountNumber} на счет ${toAccountNumber} выполнен.");
29    }
30
31    Run | Debug
32    void main() {
33
34  }
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
69
70
71
72
73
74
75
76
77
78
79
79
80
81
82
83
84
85
86 }
```

ПРОБЛЕМЫ ВХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ РЕЗУЛЬТАТЫ ЗАПРОСА Фильтр (например: text, lexed...)

Счет 9747232247: Текущий баланс: 5300.0  
Счет 6492469143: Снято 2000.0. Новый баланс: 7700.0  
Счет получателя 1234567898 не найден.  
Найден аккаунт: Иван Иванов  
[BankAccount{accountNumber: 6492469143, balance: 7700.0, ownerName: Иван Иванов}, BankAccount{accountNumber: 9747232247, balance: 5300.0, ownerName: Мария Петрова}]

Exited.

× ⌂ ⌂ ⌂ Dart (dart) Connect Стока 105, столбец 4 Проблем: 2 UTF-8 CRLF () Dart

**Banking system.dart**

```

Функции высшего порядка + ООП.dart Abstract Classes + Inheritance.dart Static methods + Constants.dart Banking system.dart

class Bank {
    void transfer(String fromAccountNumber, String toAccountNumber, double amount) {
        return;
    }
    fromAccount.withdraw(amount);
    toAccount.deposit(amount);
    print("Перевод $amount со счета ${fromAccountNumber} на счет ${toAccountNumber} выполнен.");
}
void main() {
    Bank myBank = Bank();
    BankAccount account1 = myBank.addAccount("Иван Иванов");
    BankAccount account2 = myBank.addAccount("Мария Петрова");
    account1.deposit(10000);
    account2.deposit(5000);
    account1.displayBalance();
    account2.displayBalance();
    myBank.transfer(account1.accountNumber, account2.accountNumber, 300);
    account1.displayBalance();
    account2.displayBalance();
    account1.withdraw(2000);
    myBank.transfer(account1.accountNumber, "1234567890", 100);
    BankAccount? foundAccount = myBank.findAccount(account1.accountNumber);
    if (foundAccount != null) {
        print("Найден аккаунт: ${foundAccount.ownerName}");
    } else {
        print("Аккаунт не найден");
    }
    print(myBank.accounts);
}

Connecting to VM Service at ws://127.0.0.1:7258/P_2-cId0deg=/ws
Connected to the VM Service.
Создан счет 6492469143 для Иван Иванов.
Создан счет 9747232247 для Мария Петрова.
Счет 6492469143: Пополнено на 10000.0. Новый баланс: 10000.0
Счет 9747232247: Пополнено на 5000.0. Новый баланс: 5000.0
Счет 6492469143: Текущий баланс: 10000.0

```

**The RPG Characters game.dart**

```

Функции высшего порядка + ООП.dart Abstract Classes + Inheritance.dart Static methods + Constants.dart Banking system.dart The RPG Characters game.dart

import 'dart:math';

abstract class Skill {
    void use();
}

abstract class Character {
    String name;
    double health = 100;
    int level = 1;

    Character(this.name);

    double attack();
    void defend(double damage);

    bool isAlive() {
        return health > 0;
    }

    void displayStatus() {
        print('$name (Level: $level, Health: $health)');
    }
}

class Warrior extends Character {
    Warrior(String name) : super(name);
    @override
    double attack() {
        return 15 + level * 2;
    }
    @override
    void defend(double damage) {
        health -= damage * 0.8;
        if (health < 0) health = 0;
    }
}

Артур атакует Мерлин и наносит 17.0 урона.
Мерлин (Level: 1, Health: 88.0)
Мерлин атакует Артур и наносит 13.0 урона.
Артур (Level: 1, Health: 89.6)
Артур атакует Мерлин и наносит 17.0 урона.

```

The RPG Characters game.dart ...

```
23 class Warrior extends Character {  
24     void defend(double damage) {  
25         if (health < 0) health = 0;  
26     }  
27 }  
28 class Mage extends Character {  
29     Mage(String name) : super(name);  
30     @override  
31     double attack() {  
32         return 10 + level * 3;  
33     }  
34     @override  
35     void defend(double damage) {  
36         health -= damage;  
37         if (health < 0) health = 0;  
38         health += 5;  
39         if (health > 100) health = 100;  
40     }  
41 }  
42 class Rogue extends Character {  
43     Rogue(String name) : super(name);  
44     @override  
45     double attack() {  
46         Random random = Random();  
47         if (random.nextDouble() < 0.2) {  
48             print("Name наносит критический удар!");  
49             return (10 + level * 2) * 3;  
50         } else {  
51             return 10 + level * 2;  
52         }  
53     }  
54 }
```

ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ РЕЗУЛЬТАТЫ ЗАПРОСА

Фильтр (например: text, lex...)

Артур атакует Мерлин и наносит 17.0 урона.  
Мерлин (Level: 1, Health: 88.0)  
Мерлин атакует Артур и наносит 13.0 урона.  
Артур (Level: 1, Health: 89.6)

② Арут атакует Мерлин и наносит 17.0 урона.

The RPG Characters game.dart ...

```
49 class Rogue extends Character {  
50     double attack() {  
51     }  
52     @override  
53     void defend(double damage) {  
54         Random random = Random();  
55         if (random.nextDouble() < 0.5) {  
56             print("$name увернулся от атаки!");  
57             return;  
58         } else {  
59             health -= damage;  
60             if (health < 0) health = 0;  
61         }  
62     }  
63 }  
64 void battle(Character character1, Character character2) {  
65     print("Начинается битва между ${character1.name} и ${character2.name}!\n");  
66     while (character1.isAlive() && character2.isAlive()) {  
67         double damage1 = character1.attack();  
68         print("${character1.name} атакует ${character2.name} и наносит $damage1 урона.");  
69         character2.defend(damage1);  
70         character2.displayStatus();  
71         if (!character2.isAlive()) {  
72             print("${character2.name} повержен!\n${character1.name} побеждает!");  
73             break;  
74         }  
75         double damage2 = character2.attack();  
76         print("${character2.name} атакует ${character1.name} и наносит $damage2 урона.");  
77         character1.defend(damage2);  
78         character1.displayStatus();  
79     }  
80 }
```

ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ РЕЗУЛЬТАТЫ ЗАПРОСА

Фильтр (например: text, lex...)

Мерлин (Level: 1, Health: 64.0)  
Мерлин атакует Артур и наносит 13.0 урона.  
Артур (Level: 1, Health: 68.79999999999998)

② Артур атакует Мерлин и наносит 17.0 урона.  
Мерлин (Level: 1, Health: 52.0)  
Мерлин атакует Артур и наносит 13.0 урона.

The screenshot shows a Dart code editor interface with the following code:

```
void battle(character character1, character character2) {
    ...
}

void main() {
    Warrior warrior = Warrior("Артур");
    Mage mage = Mage("Мерлин");
    Rogue rogue = Rogue("Робин");

    battle(warrior, mage);

    print("-----\n");
    rogue.level = 5;
    print("${rogue.name} достиг ${rogue.level} уровня!");
    battle(rogue, warrior);
}

@override
void attack() {
    double damage = attackPower * level;
    ...
}
```

The code defines a `battle` function that takes two `character` parameters. It prints a message when one character attacks another. The `main` function creates instances of `Warrior`, `Mage`, and `Rogue` characters and calls the `battle` function between them. It also prints the current level of the `Rogue` character.