Java Pro 3.6

TEL-RAN
by Starta Institute

Неизменяемые классы. Дата и время

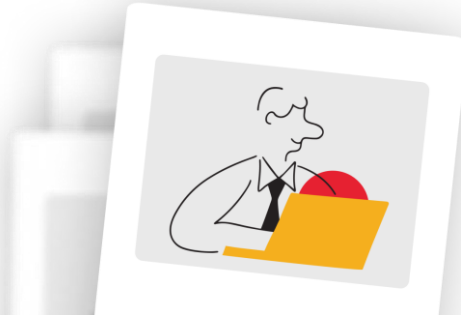
ПРЕПОДАВАТЕЛЬ

TEL-RAN
by Starta Institute

Юрий Костяной

Java/Kotlin backend-разработчик

- 3+ года опыта в коммерческой разработке
- 2+ года опыта в преподавании
- Проекты по интеграции сторонних платформ, CRM
- Проблемно-ориентированный подход в преподавании



ВАЖНО:

- Камера должна быть включена на протяжении всего занятия.
- Если у Вас возник вопрос в процессе занятия, пожалуйста, поднимите руку и дождитесь, пока преподаватель закончит мысль и спросит Вас, также можно задать вопрос в чате или когда преподаватель скажет, что начался блок вопросов.
- Организационные вопросы по обучению решаются с кураторами, а не на тематических занятиях.
- Вести себя уважительно и этично по отношению к остальным участникам занятия.
- Во время занятия будут интерактивные задания, будьте готовы включить камеру или демонстрацию экрана по просьбе преподавателя.

TEL-RAN
by Starta Institute

ПЛАН ЗАНЯТИЯ

1. Повторение
2. Основной блок
3. Вопросы по основному блоку
4. Домашняя работа

TEL-RAN
by Starta Institute

1 ПОВТОРЕНИЕ ИЗУЧЕННОГО

Повторение

Какой принцип программирования нарушен?

```
import java.util.Iterator;
import java.util.List;

public class DataHolder {
    private List<String> data;

    public DataHolder(List<String> data) {
        this.data = data;
    }

    public int removeEmpty() { // на всякий случай
        int count = 0;
        for (Iterator<String> i = data.iterator(); i.hasNext();) {
            String s = i.next();
            if (s==null || s.isBlank()) {
                i.remove();
                count++;
            }
        }
        return count;
    }
}
```

Повторение

Какой принцип программирования нарушен?

Метод removeEmpty не используется.
Нарушен принцип YAGNI

```
import java.util.Iterator;
import java.util.List;

public class DataHolder {
    private List<String> data;

    public DataHolder(List<String> data) {
        this.data = data;
    }

    public int removeEmpty() { // на всякий случай
        int count = 0;
        for (Iterator<String> i = data.iterator(); i.hasNext();) {
            String s = i.next();
            if (s==null || s.isBlank()) {
                i.remove();
                count++;
            }
        }
        return count;
    }
}
```

Повторение

TEL-RAN
by Starta Institute

Какой принцип программирования нарушен?

```
public class Calculator {
    public int add(int a, int b) {
        return a + b;
    }

    public int subtract(int a, int b) {
        return a - b;
    }

    public int multiply(int a, int b) {
        return a * b;
    }

    public int addThreeNumbers(int a, int b, int c) {
        return a + b + c;
    }
}
```

Повторение

TEL-RAN
by Starta Institute

Какой принцип программирования нарушен?

В данном примере метод addThreeNumbers повторно выполняет операцию сложения, хотя она уже реализована в методе add. Это нарушение принципа DRY.

```
public class Calculator {
    public int add(int a, int b) {
        return a + b;
    }

    public int subtract(int a, int b) {
        return a - b;
    }

    public int multiply(int a, int b) {
        return a * b;
    }

    public int addThreeNumbers(int a, int b, int c) {
        return a + b + c;
    }
}
```

Повторение

TEL-RAN
by Starta Institute

Какой принцип программирования нарушен?

```
public class MyCalculator {
    public static int add(String num1, String num2) {
        int intA = Integer.parseInt(num1);
        int intB = Integer.parseInt(num2);
        return intA + intB;
    }
}
```

Повторение

TEL-RAN
by Starta Institute

Какой принцип программирования нарушен?

```
public class MyCalculator {
    public static int add(String num1, String num2) {
        int intA = Integer.parseInt(num1);
        int intB = Integer.parseInt(num2);
        return intA + intB;
    }
}
```

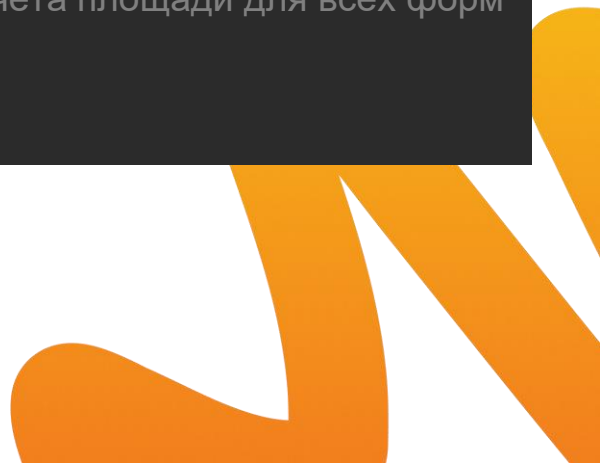
Метод сложения принимает две строки вместо двух чисел. При этом аргументы называются num1 и num2, что не соответствует их типу. Нарушен принцип POLA.

Повторение

TEL-RAN
by Starta Institute

Какой принцип программирования нарушен?

```
public class Shape {
    private double calculateArea() {
        double result = 0;
        // Логика расчета площади для всех форм
        return result;
    }
}
```



Повторение

TEL-RAN
by Starta Institute

Какой принцип программирования нарушен?

```
public class Shape {
    private double calculateArea() {
        double result = 0;
        // Логика расчета площади для всех форм
        return result;
    }
}
```

Класс Shape закрыт для расширения, т.к. метод calculateArea нельзя переопределить в наследниках. Если потребуется добавить новый тип фигуры, это потребует изменения внутри класса. Нарушение OCP.

Повторение

TEL-RAN
by Starta Institute

Какой принцип программирования нарушен?

```
public class StateChecker {
    public boolean isCorrect(boolean isOk) {
        if (isOk == true) return true;
        else if (isOk == false) return false;
        throw new RuntimeException("Некорректный статус");
    }
}
```

Повторение

TEL-RAN
by Starta Institute

Какой принцип программирования нарушен?

Чрезмерно усложнённый код.
Нарушен принцип KISS

```
public class StateChecker {
    public boolean isCorrect(boolean isOk) {
        if (isOk == true) return true;
        else if (isOk == false) return false;
        throw new RuntimeException("Некорректный статус");
    }
}
```

Можно простить до. Но, вероятно, такой код вообще не нужен.

```
public class StateChecker {
    public boolean isCorrect(boolean isOk) {
        return isOk;
    }
}
```

Повторение

TEL-RAN
by Starta Institute

Какой принцип программирования нарушен?

```
public class Employee {
    public void calculateSalary() {
        // Логика расчета зарплаты
    }

    public void saveToDatabase() {
        // Логика сохранения сотрудника в базу данных
    }
}
```

Повторение

TEL-RAN
by Starta Institute

Какой принцип программирования нарушен?

Класс Employee выполняет и расчет зарплаты, и сохранение в базу данных. Это нарушение SRP.

```
public class Employee {
    public void calculateSalary() {
        // Логика расчета зарплаты
    }

    public void saveToDatabase() {
        // Логика сохранения сотрудника в базу данных
    }
}
```

Повторение

TEL-RAN
by Starta Institute

Какой принцип программирования нарушен?

```
import java.util.ArrayList;

public class CustomList<T> {
    private ArrayList<T> items;

    public CustomList() { this.items = new ArrayList<>(); }

    public void addItem(T item) { items.add(item); }

    public T getItem(int index) { return items.get(index); }

    public int size() { return items.size(); }

    public void removeItem(T item) { items.remove(item); }
}
```

Повторение

TEL-RAN
by Starta Institute

Какой принцип программирования нарушен?

Класс CustomList повторяет функциональность класса ArrayList из стандартной библиотеки Java, что нарушает принцип "не изобретай велосипед".

```
import java.util.ArrayList;

public class CustomList<T> {
    private ArrayList<T> items;

    public CustomList() { this.items = new ArrayList<>(); }

    public void addItem(T item) { items.add(item); }

    public T getItem(int index) { return items.get(index); }

    public int size() { return items.size(); }

    public void removeItem(T item) { items.remove(item); }
}
```

Повторение

TEL-RAN
by Starta Institute

Какой принцип программирования нарушен?

```
public interface Worker {
    void work();
    void eat();
}
```

```
public class Robot implements Worker{
    @Override
    public void work() {
        // Логика работы
    }

    @Override
    public void eat() {
        throw new NotImplementedException("Робот не ест");
    }
}
```

```
public class Workman implements Worker{
    @Override
    public void work() {
        // Логика работы
    }

    @Override
    public void eat() {
        System.out.println("Обед! Я кушать пошёл.");
    }
}
```

Повторение

TEL-RAN
by Starta Institute

Какой принцип программирования нарушен?

Интерфейс Worker содержит методы для работы и еды. Класс Robot вынужден реализовывать метод eat. Нарушение ISP.

```
public interface Worker {
    void work();
    void eat();
}
```

```
public class Robot implements Worker{
    @Override
    public void work() {
        // Логика работы
    }

    @Override
    public void eat() {
        throw new NotImplementedException("Робот не ест");
    }
}
```

```
public class Workman implements Worker{
    @Override
    public void work() {
        // Логика работы
    }

    @Override
    public void eat() {
        System.out.println("Обед! Я кушать пошёл.");
    }
}
```

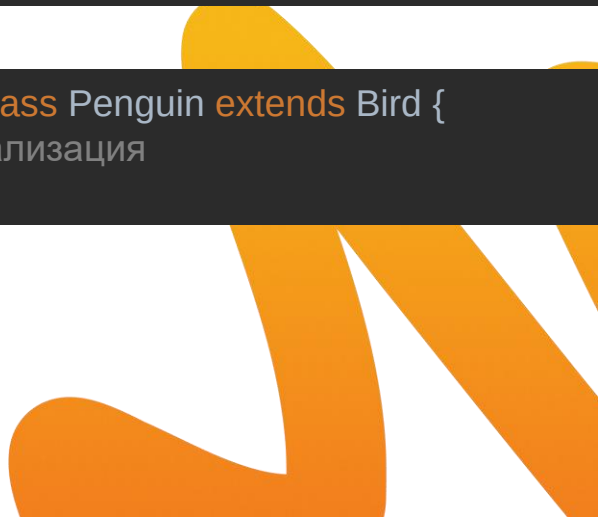
Повторение

TEL-RAN
by Starta Institute

Какой принцип программирования нарушен?

```
public class Bird {
    public void fly() {
        // Реализация
    }
}
```

```
public class Penguin extends Bird {
    // Реализация
}
```



Повторение

TEL-RAN
by Starta Institute

Какой принцип программирования нарушен?

```
public class Bird {
    public void fly() {
        // Реализация
    }
}
```

```
public class Penguin extends Bird {
    // Реализация
}
```

Класс Penguin нарушает принцип LSP, т.к. метод fly не может быть корректно реализован для пингвинов – они не летают.

TEL-RAN
by Starta Institute

2 ОСНОВНОЙ БЛОК

Введение

TEL-RAN
by Starta Institute

- Без перемен
- Когдаты

Проблема

TEL-RAN
by Starta Institute

С курса Basic мы знаем, что класс *String*, а также классы-обёртки (*Byte*, *Short*, *Character*, *Integer*, *Long* и даже *Boolean*) являются неизменяемыми. Это позволяет экономить оперативную память с помощью хранения значений этих классов в специальных пулах (реализация паттерна Легковес). Но существуют и другие неизменяемые классы, которые не используют механизм пулов.

Для чего их создавать, если удобнее работать с изменяемыми объектами?

Без перемен

Причины применения

TEL-RAN
by Starta Institute

1. Возможность использования пулов для экономии памяти.
2. Потокобезопасность. Любой поток, изменяющий объект, будет получать новый объект, т.е. нет борьбы за ресурс.
3. Защита от изменений по ошибке. Это позволяет, например, использовать объекты таких классов в качестве надёжных ключей мапы или хранить чувствительную информацию, которую могут случайно изменить в программе.
4. Хорошо подходят для механизма кеширования.

Без перемен

Неизменяемые классы

TEL-RAN
by Starta Institute

Иммутабельный (неизменяемый, immutable) класс – это класс, объекты которого после создания не могут изменить свое состояние.

То есть если в коде есть ссылка на экземпляр иммутабельного класса, то любые изменения в нем приводят к созданию нового экземпляра.

Без перемен

Как создать

TEL-RAN
by Starta Institute

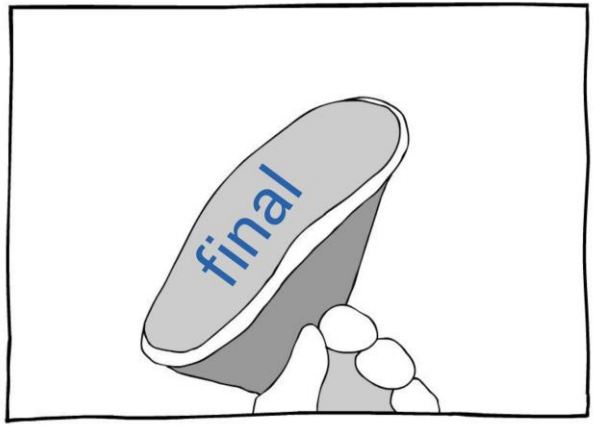
- Указать *final* перед *class* при объявлении класса. Иначе дочерние классы могут нарушить иммутабельность.
- Пометить все поля как final. Все поля класса также должны быть приватными в соответствии с принципами инкапсуляции.
- Поля инициализируются только в конструкторе.
- Не менять состояние объекта после его создания (у класса нет сеттеров и mutator-методов).
- Использовать клонирование значений полей. Для ссылочных типов использовать глубокое клонирование (создание полных клонов, а не копирование ссылок). Например, для коллекций использовать глубокое клонирование вложенных объектов.

Без перемен

Минусы

TEL-RAN
by Starta Institute

- Нарушают OCP, т.к. закрыты для расширения.
- Каждое изменение объекта требует сохранить новое значение.
- Легко забить память промежуточными значениями.



Задание

TEL-RAN
by Starta Institute

1 Создайте неизменяемый класс Point, представляющий точку на плоскости с координатами x и y.

2 Напишите класс, который создаёт точки для графика функции параболы по переданным начальному значению xStart, конечному значению xEnd и шагу.

Без перемен

Современный подход

В Java 14 появились записи (record).

Records – это неизменяемые классы, которые можно создавать упрощённым способом (более коротким кодом).

Для определения записей применяется ключевое слово record, после которого идет название и далее в круглых скобках список полей record. Это называется **каноническим** конструктором.

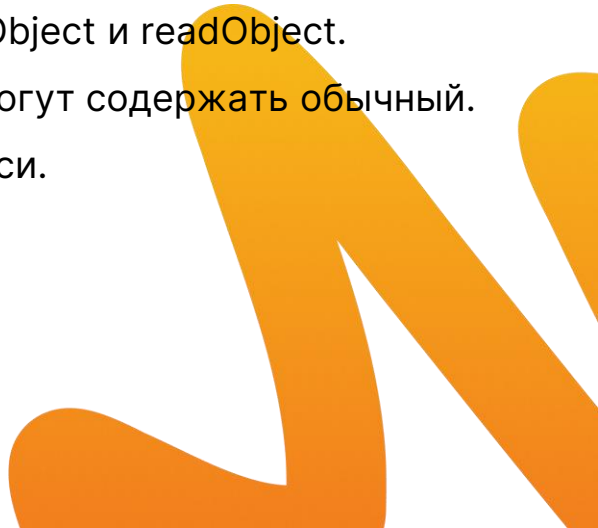
```
public record ItemRecord(String name, double price, String description, List<String> hashTags) {
    public Item changeName(String newName) {
        return new Item(newName, this.price, this.description, this.hashTags);
    }
}
```

Без перемен

Особенности записей

TEL-RAN
by Starta Institute

1. Большинство методов генерируются автоматически (геттеры, equals и hachCode, toString).
2. Позволяют создать/переопределять методы, в том числе геттеры и сеттеры, в теле.
3. Позволяют создавать статические члены.
4. Имплементируют Serializable (перевод объекта в двоичный вид, например, для записи в файл), но процесс сериализации нельзя настроить с writeObject и readObject.
5. Могут содержать статический блок инициализации, но не могут содержать обычный.
6. Может содержать внутренние классы, перечисления и записи.
7. Не могут быть предками других классов.
8. Могут быть обобщённым типом.
9. Могут имплементировать интерфейсы.
10. Являются наследниками java.lang.Record.



Задание

TEL-RAN
by Starta Institute

1 Создайте запись Student, представляющую информацию о студенте (имя, возраст, средний балл).

2 Создайте множество студентов в отдельном методе.

3 Изначально средний балл у студентов не заполнен, т.к. они только начали учиться.

4 В отдельном методе заполните средний балл каждого студента случайным образом.

Проблема

TEL-RAN
by Starta Institute

Важнейшим типом данных является дата и время:

- работа с персональными данными
- записи в электронных очередях
- проверка на совершеннолетие
- логирование и многое другое.

В Java есть много классов, которые позволяют хранить дату и время. Например, *java.util.Date* и *java.sql.Date*, *Calendar, GregorianCalendar*. Можно ещё получить текущее время системы с помощью *System.currentTimeMillis()*. Но в документации класса *Date* многие методы и конструкторы обозначены как *Deprecated* (“нерекомендуемый‘). System.currentTimeMillis() возвращает число.

Что лучше использовать в программе?

Когдаты

Пакет java.time

TEL-RAN
by Starta Institute

java.time - пакет, содержащий современные классы для работы с датами и временем.

Предыдущие классы показали себя плохо на практике по разным причинам, поэтому классы из *java.time* обладают свойствами, исправляющими эти недостатки. В частности:

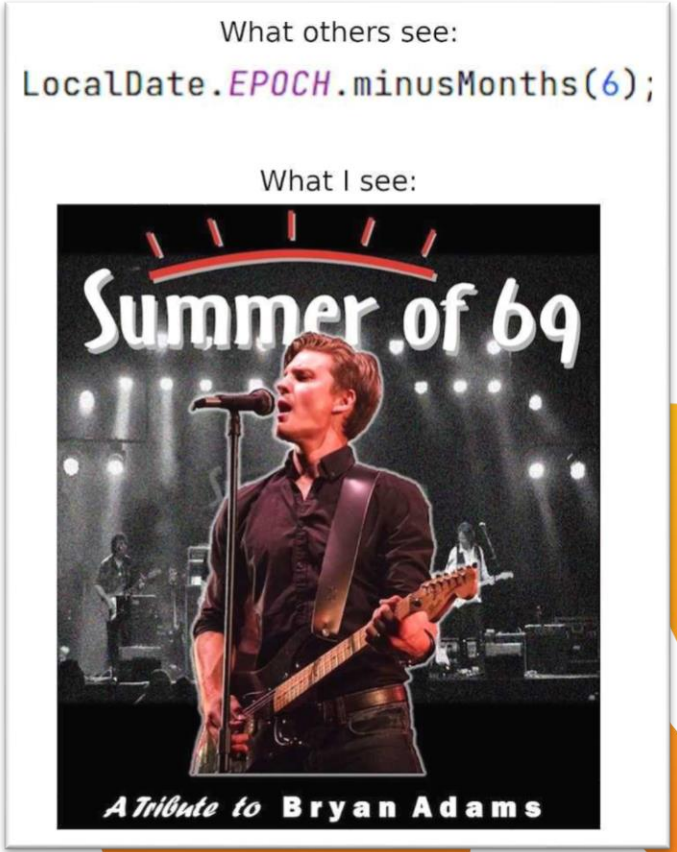
- Классы неизменяемы, что упрощает работу с ними в многопоточной среде и не защищает от случайного изменения.
- Каждый класс следует принципу единой ответственности. Существуют отдельные классы для хранения даты и времени без часового пояса и аналогичные классы с хранением часового пояса. Отдельный класс отвечает за длительность.
- Легко становится строкой в нужном формате. Также можно получить дату/время из строки заданного формата.
- Реализованы все служебные методы (equals, hashCode, compare, toString).
- Обладают удобными методами для работы.

Когдаты
LocalDate

TEL-RAN
by Starta Institute

Класс **LocalDate** из пакета *java.time* предназначен для работы с датами. Функционал этого класса позволяет создавать даты и изменять их, добавляя и отнимая необходимое количество дней/месяцев/лет.

```
LocalDate today = LocalDate.now();
```



Когдаты

Важные методы LocalDate

LocalDate now() - возвращает объект, который представляет текущую дату

LocalDate of(int year, int month, int day) - возвращает объект, который представляет дату с определенными годом, месяцем и днем

getYear() - возвращает год даты

getMonthValue() - возвращает месяц

getDayOfMonth() - возвращает день месяца (значение от 1 до 31)

getDayOfYear() - возвращает номер дня года (значение от 1 до 365)

DayOfWeek getDayOfWeek() - возвращает день недели в виде значения перечисления

DayOfWeekLocalDate plusDays(int n) - добавляет к дате некоторое количество дней

Когдаты

Важные методы LocalDate

TEL-RAN
by Starta Institute

LocalDate plusWeeks(int n) - добавляет к дате некоторое количество недель

LocalDate plusMonths(int n) - добавляет к дате некоторое количество месяцев

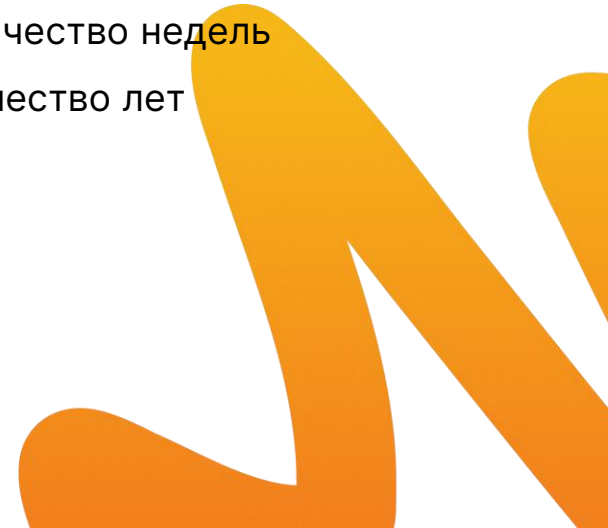
LocalDate plusYears(int n) - добавляет к дате некоторое количество лет

LocalDate minusDays(int n) - отнимает от даты некоторое количество дней

LocalDate minusMonths(int n) - отнимает от даты некоторое количество месяцев

LocalDate minusWeeks(int n) - отнимает от даты некоторое количество недель

LocalDate minusYears(int n) - отнимает от даты некоторое количество лет



Когдаты

LocalTime

TEL-RAN
by Starta Institute

Класс **LocalTime** из пакета *java.time* предназначен для работы со временем суток. Класс представляет время с часами, минутами, секундами и наносекундами. Например, *13:21.05.123456789*.

Когдаты

LocalDateTime

TEL-RAN
by Starta Institute

LocalDateTime из пакета *java.time* предназначен для работы с датами и временем. Объединяет *LocalDate* и *LocalTime*, но не хранит часовой пояс.

LocalDateTime реализует интерфейс ChronoLocalDateTime и наследует класс объекта.

Везде, где нам нужно представить дату и время без ссылки на часовой пояс, мы можем использовать экземпляры *LocalDateTime*.

Например, *LocalDateTime* можно использовать для запуска пакетных заданий в любом приложении.

Задания будут выполняться в фиксированное время в часовом поясе, в котором находится сервер.

Когдаты

Важные методы LocalDateTime

TEL-RAN
by Starta Institute

format() - используется для форматирования этой даты-времени с помощью указанного модуля форматирования.

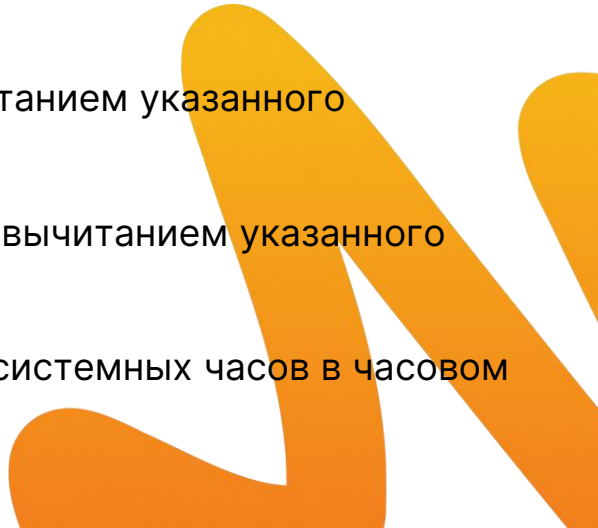
get() - используется для получения значения указанного поля из этой даты-времени в виде int.

minusMinutes() - возвращает копию этого объекта LocalDateTime с заданным количеством вычитаемых минут

minusYears() - возвращает копию объекта LocalDateTime с вычитанием указанного количества лет.

minusDays() - возвращает копию этого объекта LocalDateTime с вычитанием указанного количества дней.

now() - используется для получения текущей даты-времени от системных часов в часовом поясе по умолчанию.



Когдаты

Важные методы LocalDateTime

TEL-RAN
by Starta Institute

plusHours() - возвращает копию объекта LocalDateTime с добавленным указанным количеством часов.

plusYears() - возвращает копию этого объекта LocalDateTime с добавленным указанным количеством лет.

plusDays() - возвращает копию этого объекта LocalDateTime с добавленным указанным количеством дней.

Когдаты

ZonedDateTime

TEL-RAN
by Starta Institute

ZonedDateTime – это класс, представляющий дату-время вместе с часовым поясом.

ZonedDateTime хранит все поля даты и времени с точностью до наносекунд и часового пояса, а зона *Offset* используется для обработки локальных дат и времени.

Этот класс также используется для преобразования локальной временной шкалы *LocalDateTime* в мгновенную временную шкалу *Instant*.

Когдаты

Важные методы ZonedDateTime

TEL-RAN
by Starta Institute

String format(DateTimeFormatter formatter) - используется для форматирования этой даты-времени с помощью указанного модуля форматирования.

int get(TemporalField field) используется для получения значения указанного поля из этой даты-времени в виде int.

ZoneId getZone() - используется для получения часового пояса

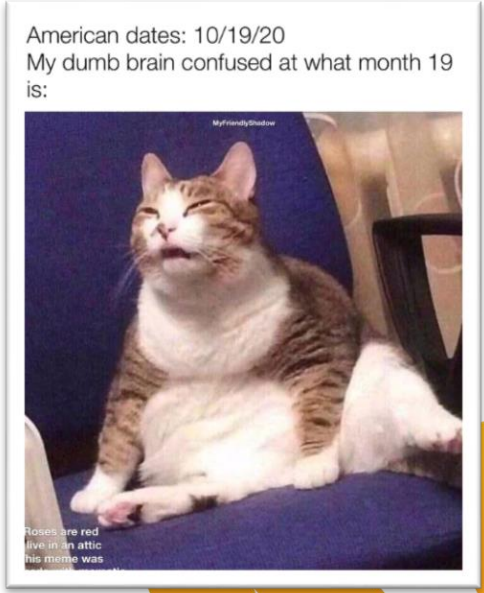
ZonedDateTime now() - используется для получения текущей даты-времени от системных часов в часовом поясе по умолчанию

Когдаты

Форматирование дат и времени

TEL-RAN
by Starta Institute

Класс *DateTimeFormatter* позволяет задавать формат даты при переводе в строку или распознавании из строки.




```
DateTimeFormatter formatterUs = DateTimeFormatter.ofPattern("MMMM, dd, yyyy HH:mm:ss", Locale.US);
System.out.println(zonedDateTime.format(formatterUs));

DateTimeFormatter formatter = DateTimeFormatter.ofPattern("dd-MM-yyyy HH:mm:ss");
System.out.println(zonedDateTime.format(formatter));
```

Когдаты

Длительность

TEL-RAN
by Starta Institute

Для хранения периода времени применяются классы

- **Instant** - представление мгновения в секундах, которые прошли с 1 января 1970 года. Например, 923,456,789 секунд и 186,054,812 наносекунд.
- **Duration** – для хранения небольших интервалов времени.
- **Period** – для хранения длительных интервалов времени.

Задание

TEL-RAN
by Starta Institute

Пользователь вводит дату своего рождения в формате dd.MM.yyyy. Выведите на экран список дат его юбилеев до 100 лет. Укажите также возраст в очередной юбилей.

TEL-RAN
by Starta Institute

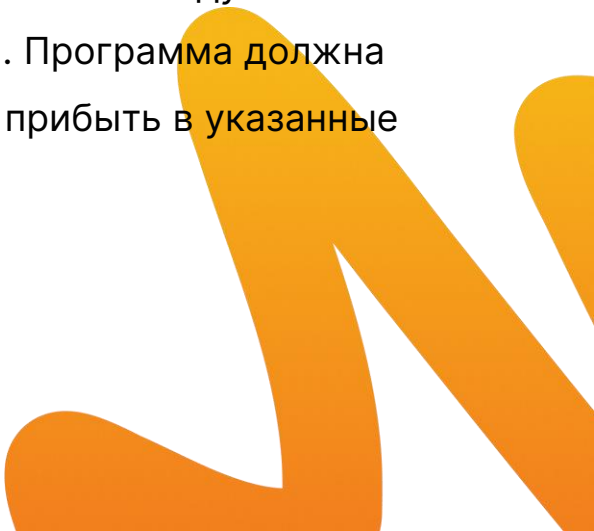
3 Домашнее задание

Домашнее задание

TEL-RAN
by Starta Institute

1 Если Вы знаете что такое комплексные числа, создайте неизменяемый класс комплексных чисел. В противном случае создайте неизменяемый класс Book, описывающий книгу с атрибутами, такими как название, автор и год издания.

2 Пользователь планирует поездку из Берлина в Стамбул на автомобиле. Выберите несколько точек на маршруте и прикиньте длительность путешествия между этими точками. Пользователь вводит дату и время своего отправления. Программа должна вывести расписание (дату и время), когда пользователь должен прибыть в указанные пункты.



Полезные ссылки

TEL-RAN
by Starta Institute

- https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/time/package-summary.html

ЗАКЛЮЧЕНИЕ

TEL-RAN
by Starta Institute