

**Московский государственный технический  
университет им. Н.Э. Баумана**

Факультет Радиотехнический  
Кафедра РТ5

Курс «Парадигмы и конструкции языков программирования»

Отчет по домашнему заданию

Выполнил:

студент группы РТ5-31Б:

Бабасанова Н. С.

Руководитель:

преподаватель каф.  
ИУ5

Гапанюк Ю. Е.

Москва, 2023г.

## Текст программы

```
use std::error::Error;
use sqlx::postgres::PgPoolOptions;
use sqlx::{FromRow, Row};
use chrono::NaiveDate;
// use: Внутри области видимости использование ключевого слова use
// создаёт псевдонимы для элементов, чтобы уменьшить повторение
// длинных путей.

// #[типаж()]
// С помощью типажа fmt::Debug все типы могут выводить
// (автоматически создавать, derive) реализацию fmt::Debug.
#[derive(Debug, FromRow)] // derive - "встроенные реализации"
struct Flight {           // структура Flight
    id: i32,
    name: String,
    country: String,
    flight_date: NaiveDate, // дата
    spaceship: String,
}

// async/.await - это специальный синтаксис Rust, который позволяет
// не блокировать поток, а передавать управление другому коду,
// пока ожидается завершение операции.
#[tokio::main]
async fn main() -> Result<(), Box<dyn Error>> { // либо результат, либо error
    // Подключение к базе данных PostgreSQL
    // переменная пул, адрес/имя:пароль
    let pool = PgPoolOptions::new()
        .connect("postgres://postgres:222@localhost:5432/testdb")
        .await?;

    // Создание таблицы в базе данных
    // Приводим в соответствие типы переменных Rust и Postgres:
    // i32 Rust соответствует INT, SERIAL Postgres;
    // String - VARCHAR, CHAR(n), TEXT, CITEXT, NAME, UNKNOWN;
    // chrono::NaiveDate - DATE;
    // i64 - BIGINT (для вывода COUNT Postgres.В последнем запросе, ниже.)
    sqlx::query(
        "CREATE TABLE IF NOT EXISTS flights (
            id SERIAL NOT NULL PRIMARY KEY,
            name VARCHAR(255) NOT NULL,
            country VARCHAR(50) NOT NULL,
            flight_date DATE NOT NULL,
            spaceship VARCHAR(50) NOT NULL
        )",
    )
    .execute(&pool)
    .await?;
```

```

// Парсинг CSV файла
let mut row_count = 0; // mut - это *изменяемое* связанное имя
//   переменной. Когда связанное имя изменяемо, это означает, что
//   мы можем поменять связанное с ним значение.
let file_path = "./space_travellers.csv"; // путь файла для парсинга
let file = std::fs::File::open(file_path)?;
let mut reader = csv::Reader::from_reader(file);
// Чтение содержимого файла.

// Чтение данных из CSV и запись в базу данных
// core::option::Option Some(value) - кортежная структура, обёртка для значения типа T,
//   для перехвата ошибок вместо вызова паники с помощью макроса panic!
while let Some(result) = reader.records().next() {
    let record = result?;
    let flight = Flight {
        // unwrap() - если возвращенный объект Result представляет константу Ok, то метод
unwrap()
        // возвращает объект, который содержится в константе Ok. Но если объект Result
// представляет константу Err, то метод unwrap() вызывает макрос panic!.
        id: record[0].parse().unwrap(),
        name: record[1].to_string(),
        country: record[2].to_string(),
        // парсинг строки из заданного пользователем формата в значение NaiveDate.
        flight_date: NaiveDate::parse_from_str(&record[3], "%Y-%m-%d").unwrap(),
        spaceship: record[4].to_string(),
    };

    // Вставка данных в базу данных
    sqlx::query(
        "INSERT INTO flights (
            id, name, country, flight_date, spaceship
        ) VALUES ($1, $2, $3, $4, $5)",
    )
    .bind(flight.id)
    .bind(&flight.name)
    .bind(&flight.country)
    .bind(flight.flight_date)
    .bind(&flight.spaceship)
    .execute(&pool)
    .await?;
    row_count += 1;
}

println!("В таблице: {} записей", row_count);
println!("*****");

// Пример запроса 1 к базе данных для вывода в консоль
println!("Список космонавтов Советского Союза, взлетающих в 1960-х г.г.");
let mut index_number = 0;
let time_interval = sqlx::query(
    "SELECT name, country, flight_date, spaceship FROM flights

```

```

WHERE flight_date >= '1961-01-01' AND flight_date < '1969-12-31' AND country LIKE 'Co%'
")
.fetch_all(&pool)
.await?;
// Итератор. Из time_interval: Vec<PgRow> присваиваем переменным значения строки. Вывод.
for interval in time_interval {
    let name: String = interval.get("name");
    let country: String = interval.get("country");
    let flight_date: NaiveDate = interval.get("flight_date");
    let spaceship: String = interval.get("spaceship");
    index_number += 1;
    println!( // чтобы не скакали надписи 1,2,3, ..., были вровень с 10, т.д.
        "{index_number:>2}. {}, {}, {}, {}",
        name, country, flight_date, spaceship
    );
}
println!("*****");

// Пример запроса 2 к базе данных для вывода в консоль
println!("Список стран, космонавты которых побывали в космосе 10 и более раз (по
убыванию)");
index_number = 0;
let select_query = sqlx::query
    ("SELECT country, COUNT(country) FROM flights
    GROUP BY country HAVING COUNT (country) > 10 ORDER BY COUNT(country) desc;
    ")
    .fetch_all(&pool)
    .await?;
for el in select_query {
    let country: String = el.get("country");
    let cnt: i64 = el.get("count");
    index_number += 1;
    println!(
        "{index_number}. {} ({})",
        country, cnt
    );
}
println!("*****");

Ok()
}

```

Результаты вывода

```
PS C:\Users\nanny\Desktop\rust\hello\travellers> cargo run
Finished dev [unoptimized + debuginfo] target(s) in 0.18s
Running `target\debug\travellers.exe`
```

В таблице: 645 записей

\*\*\*\*\*

Список космонавтов Советского Союза, взлетающих в 1960-х г.г.

1. Юрий Гагарин, Советский Союз, 1961-04-12, Восток 1
2. Герман Титов, Советский Союз, 1961-08-06, Восток 2
3. Андриян Николаев, Советский Союз, 1962-08-11, Восток 3
4. Павел Попович, Советский Союз, 1962-08-12, Восток 4
5. Валерий Быковский, Советский Союз, 1963-06-14, Восток 5
6. Валентина Терешкова, Советский Союз, 1963-06-16, Восток 6
7. Константин Феоктистов, Советский Союз, 1964-10-12, Восход 1
8. Владимир Комаров, Советский Союз, 1964-10-12, Восход 1
9. Борис Егоров, Советский Союз, 1964-10-12, Восход 1
10. Павел Беляев, Советский Союз, 1965-03-18, Восход 2
11. Алексей Леонов, Советский Союз, 1965-03-18, Восход 2
12. Георгий Береговой, Советский Союз, 1968-10-26, Союз 3
13. Владимир Шаталов, Советский Союз, 1969-01-14, Союз 4
14. Евгений Хрунов, Советский Союз, 1969-01-15, Союз 5
15. Борис Воынов, Советский Союз, 1969-01-15, Союз 5
16. Алексей Елисеев, Советский Союз, 1969-01-15, Союз 5
17. Валерий Кубасов, Советский Союз, 1969-10-11, Союз 6
18. Георгий Шонин, Советский Союз, 1969-10-11, Союз 6
19. Анатолий Филипенко, Советский Союз, 1969-10-12, Союз 7
20. Виктор Горбатко, Советский Союз, 1969-10-12, Союз 7
21. Владислав Волков, Советский Союз, 1969-10-12, Союз 7

\*\*\*\*\*

Список стран, космонавты которых побывали в космосе 10 и более раз (по убыванию)

1. United States (391)
2. Советский Союз (72)
3. Россия (62)
4. China (20)
5. Japan (14)
6. Canada (12)

\*\*\*\*\*