

Rust. Установка, 1-ая программа

Выполнили: Золин Иван, Рубанова Валерия

Группа: 5030102/00201

Установка

- Установка Rust (для debug)
 - <https://www.rust-lang.org/tools/install>
- Установка инструментов C++ (если не установлен Visual Studio)
 - <https://visualstudio.microsoft.com/downloads/?q=build+tools>
- Установка VsCode(IDE)
 - <https://code.visualstudio.com/download>
- Установка Docker
 - <https://docs.docker.com/engine/install/>
- Образ Rust для Docker
 - https://hub.docker.com/_/rust
- Установка Cargo(если не установился вместе с Rust)
 - <https://doc.rust-lang.org/cargo/getting-started/installation.html>

Начало работы

- Первая программа: main.rs

- ```
fn main() {
 println!("Hello, world!");
}
```

 //main() – точка входа в программу, ! – после функции означает вызов макроса

- Сборка и запуск

- ```
rustc main.rs  
./main
```

- Замечания

- Требуется явное указание имени файла.
 - Если потребуется указать флаги компилятора или внешние зависимости, то команда усложнится.
 - Вручную получение правильных версий зависимостей и поддержание их в актуальном состоянии.

Использование Cargo

- *Cargo* — менеджер пакетов Rust.
- Создание нового проекта
 - `$ cargo new hello_world`
- После этого структура проекта:
 - `hello_world/`
 - ├── `Cargo.toml`
 - ├── `Cargo.lock`
 - ├── `.gitignore`
 - ├── `target/`
 - └── `src/`
 - └── `main.rs`
- Запуск
 - `cargo build`
`./target/debug/hello_world`

Docker

- Dockerfile

- FROM rust:latest

```
WORKDIR /app
```

```
COPY ./Cargo.toml ./Cargo.lock ./
```

```
COPY ./src ./src
```

```
RUN cargo build --release
```

```
CMD ["/target/release/hello_world"]
```

- Docker commands

- docker build -t hello_world_image .
 - docker run hello_world_image

Материалы

- Для ознакомления с ЯП Rust можете посмотреть репозиторий в GitHub:
 - <https://github.com/IMZolin/New-programming-languages>

Спасибо за внимание!

