HAW Hamburg
Prof. Dr. Franz Korf/

Topic: Graph structures in Rust

Deadline: At the beginning of the third exercise session

Hint for 4-persons teams working on a small self-defined project in Rust during exercise 3 to 6:

- Put together your team of four persons until your second exercise.
- Create a git-repo with a descriptive name on our git server. Please grant Silke Behn and me read-only access to this repo.
- Think about what you want to work on in the small project. You are welcome to access other crates or services, as long as the project still contains a relevant amount of Rust code written by the team.

This task is not labor-intensive, but it will take some time until you have put together your team and found an interesting project of the right size. **Hence, start this subtask as soon as possible.**

Exercise: Implementation of a sorted doubly linked list

You are certainly familiar with this data structure. You can find a description of an unsorted doubly linked list at https://en.wikipedia.org/wiki/Doubly_linked_list. **Please note that you should implement a sorted doubly linked list in this task.** Of course, other data structures implement the task that a sorted list should fulfill very efficiently. But in the current exercise you implement a sorted doubly linked list to exercise the handling of data structures that have cyclic references in Rust.

Please implement the standard methods of a sorted list. Especially implement a method that inserts a new element at the right position in a sorted list.

You have to work on the problem that reference cycles can leak memory (s. lecture 2). The page https://rtoch.com/posts/rust-doubly-linked-list/ (the quality of which is not checked up to now) avoids the memory leak using an adapted drop function. As shown in lecture 2, an alternative approach bases on weak references. **Please implement both solutions.**

To fulfill this exercise, you should be familiar with following Rust topics:

- The use of **`RefCell<T>`**
- The use of **`struct`**
- The Rust **`impl`** statement, which binds methods to a **`struct`**.
- The use of **`Drop`** trait. Please note that you only need to know the basics of the Rust trait concept for this exercise. (see page 304 of the book Blandy, J., & Orendorff, J. (2021). Programming rust. O'Reilly Media, Incorporated).

Minimum requirements for the sorted doubly linked list:

- The type of list elements is defined by a type-parameter `<T>`. `T` must implement the `Ord` trait.
- Implementation of typical methods of a sorted list. Especially a method that inserts an element into a sorted list.
- Documentation of user defined types. Represent an empty list and a list containing two elements graphically.
- Suitable test functions. Especially tests validating that memory leaks do not occur.