15.01.18.

SEMINAR 7

Implement Futures in C++ in the C# style, having the continuation possibility

- get & wait are to be called by the consumer (future)
- set is to be called by the producer (promise)
- if someone calls continueWith after set() is called, then the function passed to the continueWith() function will be executed immediately.
- what thread will execute the continueWith() function?
    - if the listener counts after the event, then it gets the event
    - if the future already has a value, then continueWith continues on the thread being called so
    - if the producer calls the set function, then all the futures will be enqueued.
- continueWithOnSameThread(func)
    - function executes on the thread calling this function (if the future is already completed) or on the thread calling set() (if the future is not completed yet)
- continueWith(func, threadPool)

```
template <typename R>
std::shared_ptr<Future<R>> continueWith(std::function<R(T)> func,
                    IThreadPool* threadPool);
```

Are futures movable? No, don't copy the future, but copy a reference.