

Wetseparate logische expressies

```
TaggedAmountPtrs to_bas_saldos(TaggedAmountPtrs const& ta_ptrs) {  
    TaggedAmountPtrs result{};  
    auto is_bas_account_transaction = [](TaggedAmountPtr const& ta_ptr){  
        return (  
    };  
    return result;  
}
```

```
TaggedAmountPtrs to_bas_saldos(TaggedAmountPtrs const& ta_ptrs) {  
    TaggedAmountPtrs result{};  
    auto is_bas_account_transaction = [](TaggedAmountPtr const& ta_ptr){  
        return (  
    };  
    return result;  
}
```

West separate logical expressions

```
TaggedAmountPtrs to_bas_saldos(TaggedAmountPtrs const& ta_ptrs) {  
    TaggedAmountPtrs result{};  
    auto is_bas_account_transaction = [](TaggedAmountPtr const& ta_ptr){  
        return (  
    };  
    return result;  
}
```

West separate logical expressions

```
bool operator==(TaggedAmountClass const& other) const {
    auto result =      this->date() == other.date()
        and this->cents_amount() == other.cents_amount()
        and std::all_of(
            m_tags.begin()
            ,m_tags.end()
            , [&other](Tags::value_type const& entry) {
                return (
                    (entry.first.starts_with("_"))
                    or (
                        other.tags().contains(entry.first)
                        and other.tags().at(entry.first) == entry.second));
            });
    return result;
}
```