Assignment #7 - Design by Contract
Discrete Mathematics

Anders Kalhauge

Fall 2017

1 Introduction

Consider an Account class. It should have two methods:

- `Deposit(double amount)` which should increase the account's balance with the amount, the amount should always be positive

- `Withdraw(double amount)` which on the other hand should decrease the account's balance. Also this amount should be positive. If the amount exceeds the balance, the balance should be left untouched and an exception should be thrown.

2 Assignment

Create a class that implements the behaviour mentioned in 1 Introduction. Use Code Contracts to enforce Design by Contract$^{\text{TM}}$ principles. Also create a small program, that uses the account class.

The report should besides from a git url to the project include a short textual description on how the code contract should be interpreted.

Hand in

A link to the github repository including a short textual description on how the code contract should be interpreted (README.md). In groups on Moodle by Friday December 5$^{\text{th}}$