# Optimizing memory

## Actual state of project

To measure the demand of memory of the application it is started in release mode and the value from the Task Manager application is taken as a baseline.



## Identifying problems

-The list of customers is passed to the **FrmEdit** directly and not via reference. Therefore the demand of memory increases twice with every added customer 🡪 pass by reference

-Function ‘**EncryptCustomer**’:

* for every field of the customer a separate variable and therefore a lot of additional string variables are needed 🡪 use the membervariables of the customer directly
* The customer is passed directly 🡪 pass by reference

🡪Customers are already reference types 🡪 passing by reference would not reduce memory demand 🡪 this measure is not used!

-Function ‘**ContainsCharacterBeforeA**t’ in Customer Class:

* Remove temporary variable temp (array of characters)

-Function ‘**NoInvalidSymbols**’ in Customer Class:

* Remove temporary variable temp (array of characters)

-Function **GetCustomerFromInput** in MainForm

* There are strings used for parts of the field 🡪 use the parts of the fields directly

-Everytime when **a new form is opend** (e.g. edit-form) a instance from this form is created and afterwards it is not disposed 🡪 dispose(), GC.collect

-Everytime the **file is read/written** a new StreamReader/Writer object is created and not released afterwards 🡪 dispose(), GC.Collect

## Result

The memory usage has decreased to steady 8kb of ram opposed to steady growing amount before optimization.

