

# Nokia PARO

## Laboratorium 3

Mikołaj Szymczyk

Marzec 2022

### Spis treści

<b>1</b>	<b>Wstęp</b>	<b>2</b>
<b>2</b>	<b>Zadanie 1</b>	<b>2</b>
<b>3</b>	<b>Zadanie 2</b>	<b>2</b>
<b>4</b>	<b>Zadanie 3</b>	<b>3</b>
<b>5</b>	<b>Zadanie 4</b>	<b>3</b>
<b>6</b>	<b>Zadanie 5</b>	<b>3</b>
<b>7</b>	<b>Zadanie 6</b>	<b>4</b>

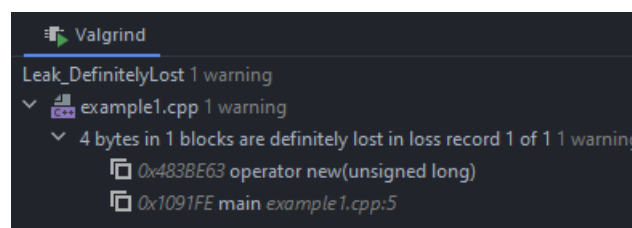
# 1 Wstęp

Celem laboratorium było zapoznanie się z:

- zarządzaniem pamięcią aplikacji napisanych w języku C++,
- narzędziem Valgrid,
- smart pointerami m.in. `shared_ptr` oraz `unique_ptr`
- zastosowaniem wyjątków

## 2 Zadanie 1

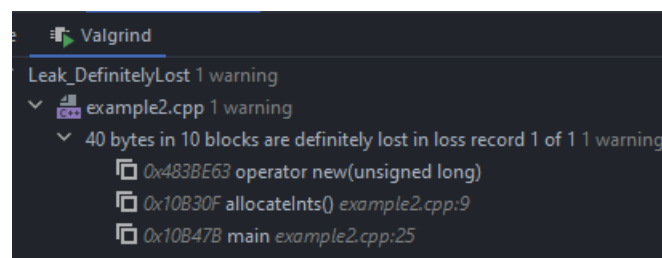
Raport narzędzia Valgrid (rys. 1) wskazał na nie zwolnienie pamięci wskaźnika *num*. Rozwiązaniem tego problemu było zastosowanie operatora *delete* na zmiennej *num*.



Rysunek 1: Raport narzędzia Valgrid dla zadania 1

## 3 Zadanie 2

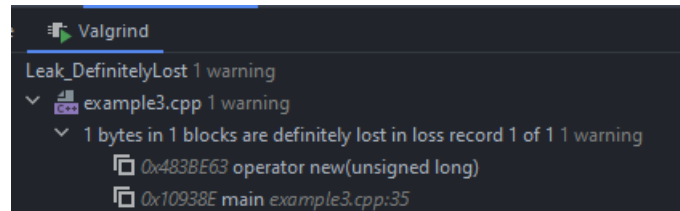
Raport narzędzia Valgrid (rys. 2) wskazał na nie zwolnienie pamięci wektora wskaźników *num*. Rozwiązaniem tego problemu było zastosowanie zaimplementowanej funkcji *deallocateInts* z parametrem *num*.



Rysunek 2: Raport narzędzia Valgrid dla zadania 2

## 4 Zadanie 3

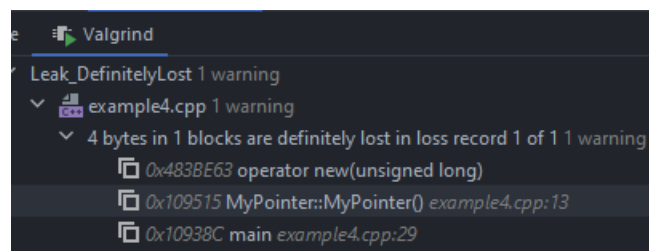
Raport narzędzia Valgrid (rys. 3) wskazał na nie zwalnianie pamięci wskaźnika *rsc*. Rozwiązaniem tego problemu było przeniesienie wywołania destruktora poza klamry try catch.



Rysunek 3: Raport narzędzia Valgrid dla zadania 3

## 5 Zadanie 4

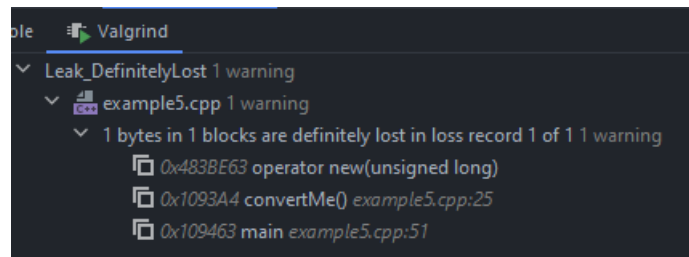
Raport narzędzia Valgrid (rys. 4) wskazał na nie zwalnianie pamięci w klasie *MyPointer*. Problemem okazało się pole klasy, które było wskaźnikiem. Rozwiązaniem tego problemu było umieszczenie funkcji *foo* w klamrach try oraz umieszczenie w klamrach catch operatora delete dla pola klasy



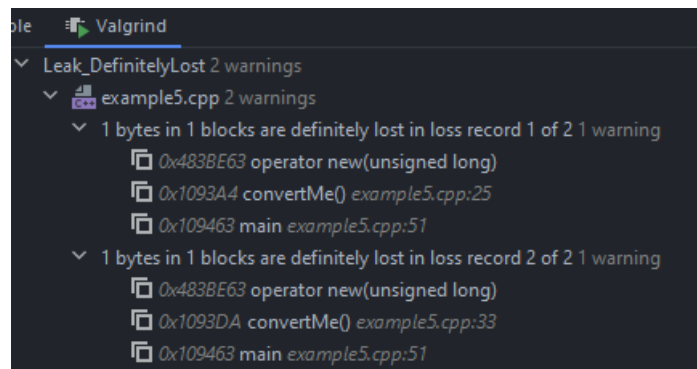
Rysunek 4: Raport narzędzia Valgrid dla zadania 4

## 6 Zadanie 5

Raporty narzędzia Valgrid (rys. 5 i 6), po małej modyfikacji kodu, wskazały na nie zwalnianie pamięci wskaźników *pFirst* oraz *pSecond*. Rozwiązaniem tego problemu było zastosowanie *unique\_ptr* z biblioteki *memory*.



Rysunek 5: Raport narzędzia Valgrid dla zadania 5



Rysunek 6: Raport narzędzia Valgrid dla zadania 5

## 7 Zadanie 6

Po zaimplementowaniu ciała funkcji makeFile oraz metody addToFile z wykorzystaniem shared\_ptr z biblioteki memory. Raporty narzędzia Valgrid nie wykazywał wycieków pamięci