A programozás alapjai 2.

Házi Feladat Dokumentáció

**Adatcsomagok statisztikája**

Készítette: Lovászi Márton, IKQOC2

lovaszi.marton@edu.bme.hu

2025. 05. 05.

Tartalomjegyzék

[Felhasználói dokumentáció 3](#_Toc198501083)

[Osztályok statikus leírása 4](#_Toc198501084)

[IP\_PACKET 4](#_Toc198501085)

[Felelőssége 4](#_Toc198501086)

[Attribútumok 4](#_Toc198501087)

[Metódusok 4](#_Toc198501088)

[PRIORITY\_IP\_PACKET 5](#_Toc198501089)

[Felelőssége 5](#_Toc198501090)

[Ősosztályok 5](#_Toc198501091)

[Attribútumok 5](#_Toc198501092)

[Metódusok 5](#_Toc198501093)

[IP\_STATISTIC 5](#_Toc198501094)

[Felelőssége 5](#_Toc198501095)

[Attribútumok 5](#_Toc198501096)

[Metódusok 6](#_Toc198501097)

[UML osztálydiagramm 7](#_Toc198501098)

[Összegzés 8](#_Toc198501099)

[Mit sikerült és mit nem sikerült megvalósítani a specifikációból? 8](#_Toc198501100)

[Mit tanultál a megvalósítás során? 8](#_Toc198501101)

[Továbbfejlesztési lehetőségek 8](#_Toc198501102)

[Képernyőképek a futó alkalmazásról 9](#_Toc198501103)

# Felhasználói dokumentáció

Az alkalmazás használatához kell egy log fájl, amely tartalmazza a router adatait. Ebből fogja a program kiolvasni a statisztikákat. Ha nincsen log fájl, akkor a program nem tud működni. A log fájlnak abban a mappában kell lennie, amelyben a program van. A futtatható fájlt elindítva egy konzol ablak jelenik meg amely arra kér, hogy adjuk meg a router log fájljának a nevét, ezzel lehetővé teszi, hogyha van több log fájl a mappában akkor ki tudjuk választani, hogy melyiket szeretnénk használni. A fájlnév megadása után 6 darab opciót kínál fel a program. Az 1-4-ig opciók arra vannak, hogy egyenként külön-külön tudjuk lefuttatni a specifikációban szereplő 4db statisztikai funkciót. Az 5. opció az összes futtatása, ilyenkor a program végigmegy az összes funkción és nem kell egyenként lefuttatni. A 6. opció a programból való kilépés. A választott lehetőség megadását az opció számának a konzolra írásával tudjuk megtenni. Több opciónál kell megadni utólagosan további információkat pl: egy tetszőleges feladó-címzett IP cím pár megadása szükséges az első opciónál mivel e között a két IP cím között fog átlagos sávszélességet számolni. Ezeket szintén a konzolon lehet megadni. Ha befejeztük a statisztikák futtatását és kilépünk a programból a 6. opció választásával akkor az összes statisztika elmentődik egy AnalyzerLog.txt fájlba. Fontos, hogy elmentsük egy másik fájlba a tartalmát mivel ez a fájl minden futtatás elején törlődik és csak az adott futtatás adatait tárolja el. A statisztikákat nem csak az AnalyzerLog.txt fájlba menti el a program, hanem az adott funkció futtatása után kiírja a program a konzolra is az eredményt.

A router log fájljának követnie kell formai mintákat:

102.45.187.23 87.162.34.10 00:00:00 00:00:03 421 2

Az adatok szóközzel vannak elválasztva. A példa sorban az első két elem a feladó és a címzett IP címe. A harmadik és a negyedik elem a beérkezés és a tovább küldés időpontja. Az ötödik elem a csomag mérete Mb-ben. A hatodik elem pedig a csomag prioritása. Ha a prioritás 0 akkor nem prioritásos csomag.

A programban ha például az 1-es opciót választjuk akkor az választott szám beírása után be kell írnunk a két IP címet. Majd a program pedig szöveges válasz formájában kiírja a statisztikát

példa az alábbi bemenet-kimenet párra:

**Bemenet:**

1)Atlagos savszelesseg szamitasa egy adott forras-cel parra

2)Adott prioritasu csomagok atlagos erkezesi idokozenek szamitasa

3)Az osszes csomagbol adodo savszelesseg szamitasa

4)Az atvitt prioritasos csomagok adatmennyisegenek aranya az osszes adathoz kepest

5) Az osszes funkcio futtatasa

6) kilepes

A valasztott opcio szama: 1

Adjon meg egy kereseshez hasznalando forras-cel ip cim part

A forras ip cim: 102.45.187.23

A cel ip cim: 87.162.34.10

**Kimenet:**

Az atlagos savszelesseg az adott forras-cel parra: 145.167Mbps

Ez a szöveges válasz pedig megtalálható lesz az AnalyzerLog.txt fájlban is a következő alkalomig ameddig újra nem indítjuk a programot.

# Osztályok statikus leírása

## IP\_PACKET

### Felelőssége

Ennek az osztálynak az a dolga, hogy reprezentáljon egy IP csomagot a beolvasott router log fájl egy sora mentén. Ez az osztály tárolni fogja a csomagok összes információját.

### Attribútumok

#### Privát

* SENDER\_IP string: Ez a változó a csomag küldőjének IP címét rögzíti
* RECEIVER\_IP string: Ez a változó a csomag címzettjének IP címét rögzíti
* TIME\_OF\_RECEIVING int: Ez változó a csomag fogadásának egy időpontját tárolja el int formában mivel ezzel könnyebb lesz később számolni.
* TIME\_OF\_SENDING int: Ez a változó a csomag tovább küldésének időpontját rögzíti szintén int formátumban.
* P\_SIZE int: az adott csomag méretét rögzíti Mb-ben.

### Metódusok

#### Publikus

* IP\_PACKET() void : Konstruktor az IP csomaghoz
* ~IP\_PACKET() virtual void : Destruktor az IP csomaghoz

Getter függvények amelyekkel hozzá lehet férni a privát tagváltozók értékéhez:

* + getSenderIp(): string
  + getReceiverIp(): string
  + getTimeOfSending(): int
  + getPsize(): int
* PrintInfo() virtual void : Kiírja a csomag adatait.
* GetSecondsInRouter() virtual int : Megmutatja hogy hány másodpercet töltött a routerben, azaz mennyi idő telt el a beérkezés és küldés között. Ez azért szükséges, hogy a statisztikai számolásoknál ne kelljen kódot ismételni.

## PRIORITY\_IP\_PACKET

### Felelőssége

Ez az osztály prioritási értékkel rendelkező IP csomagokat valósít meg.

### Ősosztályok

Az IP\_PACKET osztály leszármazottja ez az osztály mivel minden tulajdonsága megegyezik egy általános IP csomaggal, csak van egy plusz prioritási értéke.

### Attribútumok

#### Privát

* PRIORITY\_RATING int: A csomag prioritási besorolását tárolja

### Metódusok

#### Publikus

* PRIORITY\_IP\_PACKET() void: Az osztály konstruktora
* ~PRIORITY\_IP\_PACKET() void: Az osztály destruktora
* getPriorityRating() int : A prioritási érték getter függvénye
* Printinfo() void: Kiírja a csomag adatait
* GetSecondsInRouter() int: Megmutatja hogy hány másodpercet töltött a routerben, azaz mennyi idő telt el a beérkezés és küldés között. Ez azért szükséges, hogy a statisztikai számolásoknál ne kelljen kódot ismételni.

## IP\_STATISTIC

### Felelőssége

Ez az osztály a statisztikai függvényeket valósítja meg. Fő célja annak hogy osztály formában vannak, hogy jobban átláthatóbb legyen a kód, és el legyenek szeparálva a futtatáshoz szükséges UI függvények és számolásokat csináló statisztikai függvények.

### Attribútumok

#### privát

* Data std::vector<IP\_PACKET\*> : Ebben a vektorban a az eltárolt csomagok szerepelnek, heterogén kollekciót valósít meg mert PRIORITY\_IP\_PACKET elemek is vannak benne. Ezeken az elemeken lesznek lefuttatva a statisztikai függvények.
* AvarageBandwith\_SENDER\_RECEIVER float: A kiszámolt átlag sávszélesség az adott IP cím pár között van benne eltárolva
* AvarageArival\_PRIORITY float: A kiszámolt átlag érkezési idő az adott prioritású csomagok között van benne eltárolva
* Bandwith\_ALL float : Az összes adatból származó sávszélességet tárolja
* Ratio\_PRIO\_NON\_PRIO float: A prioritásos csomagok adatmennyiségének aránya az összes adathoz képest van benne tárolva

### Metódusok

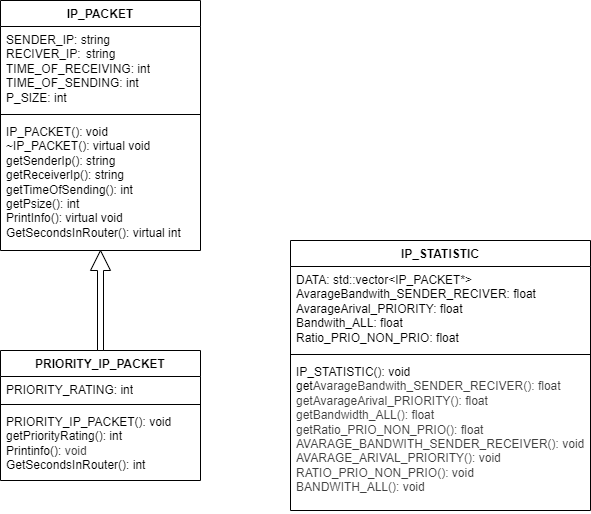
#### Publikus

* IP\_STATISTIC() void : Az osztály konstruktora

Getter függvények, amelyekkel hozzá lehet férni a privát tagváltozók értékéhez:

* + getAvarageBandwith\_SENDER\_RECEIVER() float:
  + getAvarageArival\_PRIORITY(): float
  + getBandwidth\_ALL(): float
  + getRatio\_PRIO\_NON\_PRIO(): float
* AVARAGE\_BANDWITH\_SENDER\_RECIVER() void: Megállapítja az átlag sávszélességet egy adott küldő-cél IP cím párra
* AVARAGE\_ARRIVAL\_PRIORITY() void: Megállapítja az átlag érkezési időközét egy adott prioritású csomag csoportnak
* RATIO\_PRIO\_NON\_PRIO() void: Megállapítja a prioritásos csomagok adatmennyiségének arányát az összes adathoz képest
* BANDWITH\_ALL() void: Megállapítja az összes adatból származó sávszélességet

# UML osztálydiagramm



# Összegzés

## Mit sikerült és mit nem sikerült megvalósítani a specifikációból?

A specifikációból mindent sikerült megvalósítanom amit elterveztem.

## Mit tanultál a megvalósítás során?

Amit a legnagyobb tanulságnak éltem meg, hogy mennyire fontos az, hogy legyen egy átlátható terv mielőtt nekikezdek a feladatnak. Habár a házifeladat leírás nem kérte specifikusan, de készítettem magamnak folyamatábrákat a függvények működéséről és az algoritmusok kivitelezéséről melyek után már csak szinte le kellett fordítanom c++ nyelvre a jegyzeteimet és sokkal kevesebbet kellett debuggolnom mert jobban átláttam a program működését. Persze voltak utána dolgok, amiket menet közben kellet megoldanom viszont azok már nem voltak akkora kihívások. A gyengeségem, amit még gyakorolnom kell az a jobban értelmezhető és konzisztens változónév adás

## Továbbfejlesztési lehetőségek

Szerintem ezt a programot könnyen tovább lehetne fejleszteni mivel statisztikai adatokat számol és abból bőven van még, amit hozzá lehetne adni.

# Képernyőképek a futó alkalmazásról



