Élettartam szabályok

Pataki Norbert

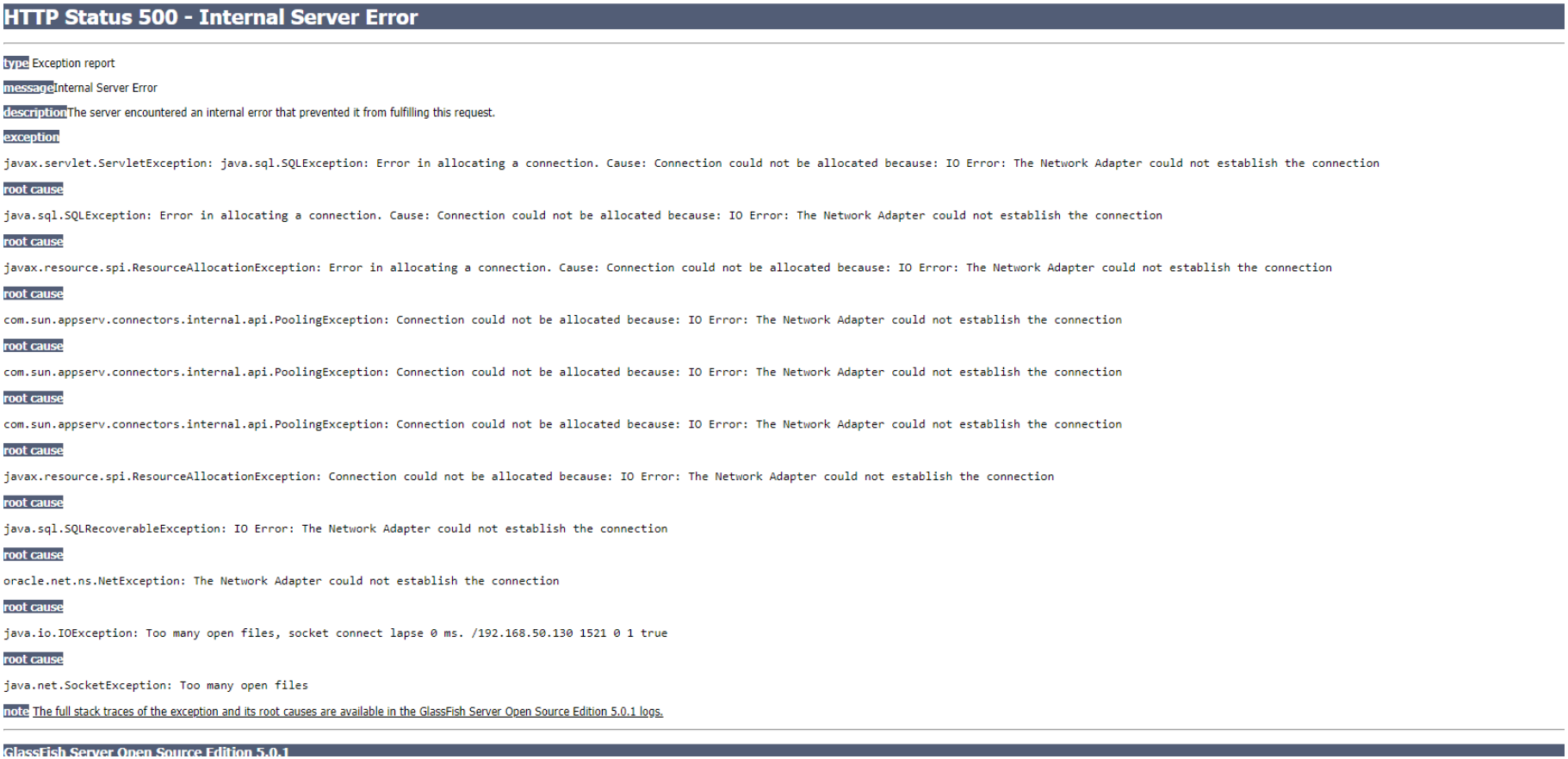
Programozási Nyelvek és
Fordítóprogramok Tanszék

Programozási Nyelvek (C++)

Fogalmak

- Láthatóság ✓
- Memória modell ✓
- Élettartam:
  - Mikor jön létre? Mikortól használható?
  - Mikor szűnik meg? Meddig használható?
  - C vs C++
  - Felhasználói típusok
  - Konstruktor, destruktor
  - `std::string`, `std::vector`

Java és az erőforrások



HTTP Status 500 - Internal Server Error

type Exception report

message Internal Server Error

description The server encountered an internal error that prevented it from fulfilling this request.

exception

```
javax.servlet.ServletException: java.sql.SQLException: Error in allocating a connection. Cause: Connection could not be allocated because: IO Error: The Network Adapter could not establish the connection
```

root cause

```
java.sql.SQLException: Error in allocating a connection. Cause: Connection could not be allocated because: IO Error: The Network Adapter could not establish the connection
```

root cause

```
javax.resource.spi.ResourceAllocationException: Error in allocating a connection. Cause: Connection could not be allocated because: IO Error: The Network Adapter could not establish the connection
```

root cause

```
com.sun.appserv.connectors.internal.api.PoolingException: Connection could not be allocated because: IO Error: The Network Adapter could not establish the connection
```

root cause

```
com.sun.appserv.connectors.internal.api.PoolingException: Connection could not be allocated because: IO Error: The Network Adapter could not establish the connection
```

root cause

```
com.sun.appserv.connectors.internal.api.PoolingException: Connection could not be allocated because: IO Error: The Network Adapter could not establish the connection
```

root cause

```
javax.resource.spi.ResourceAllocationException: Connection could not be allocated because: IO Error: The Network Adapter could not establish the connection
```

root cause

```
java.sql.SQLRecoverableException: IO Error: The Network Adapter could not establish the connection
```

root cause

```
oracle.net.ns.NetException: The Network Adapter could not establish the connection
```

root cause

```
java.io.IOException: Too many open files, socket connect lapse 0 ms. /192.168.50.130 1521 0 1 true
```

root cause

```
java.net.SocketException: Too many open files
```

note The full stack traces of the exception and its root causes are available in the GlassFish Server Open Source Edition 5.0.1 logs.

GlassFish Server Open Source Edition 5.0.1

Láthatóság, élettartam

```
void f( int s )
{
  int x = s;
  ++x;
  if ( s > 0 )
  {
    ++s;
    int x = s;
    ++x;
  }
  ++x;
}
```

Élettartam

```
void f( int s )
{
  std::vector<int> v( s );
  v.push_back( s );
  if ( v[ 0 ] < v[ 1 ]  )
  {
    ++s;
    std::vector<int> v( s );
    v.push_back( s );
  }
  v.push_back( s );
}
```

Élettartam szabályok

- Automatikus változók
- Globális változók, névtérbeli, osztály statikus tagok
- Lokális statikusok
- Dinamikus változók
- Tömbelemek
- Osztály adattagok
- Temporálisok

Automatikus változók

```
void f( int s )
{
  std::vector<int> v( s );
  if ( v.empty() )
  {
    std::vector<int> x( s + 1 );
    ++s;
    x.push_back( s );
  }
  v.push_back( s + 1 );
}
```

- Létrejönnek: a vezérlés eléri a deklarációt
- Megszűnnek: a vezérlés elhagyja a deklaráló blokkot

Példa: lokális statikus

```
#include <iostream>

void f()
{
  static int x = 1;
  std::cout << x << std::endl;
  ++x;
}

int main()
{
  f();
  f();
}
```

Lokális statikus változók

- Létrejönnek: a vezérlés először eléri a deklarációt
- Megszűnnek: a program futásának a végén

Osztály statikus tag – példa

```
class Foo
{

  static const char* filename;

};
```

Globális, osztály statikus tagok, névtérbeli változók

- Létrejönnek: a program indulásakor
- Megszűnnek: a program futásának a végén

Dinamikus változók

- Létrejönnek: a vezérlés eléri a `new`-t
- Megszűnnek: a vezérlés eléri a `delete`-et.

Problémák

Osztály adattagok

- Létrejönnek: a tartalmazó objektum létrejöttekor
- Megszűnnek: a tartalmazó objektum megszűnésekor

Példa

```
class Statistics
{
public:
  std::vector<double> data;
};

// ...

void f()
{
  Statistics stat;
  std::cout << stat.data.size();
  stat.data.push_back( 1.23 );
}
```

Tömbelemek

```
void f()
{
  Statistics arr[ 5 ];
  arr[ 2 ].data.push_back( 4.44 );
}
```

- Létrejönnek: a tartalmazó tömb létrejöttekor
- Megszűnnek: a tartalmazó tömb megszűnésekor

Kifejezések kiértékelése

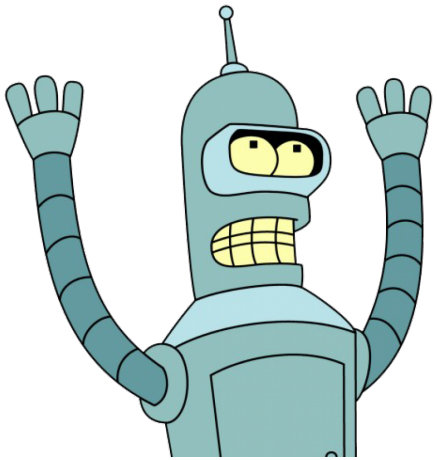
```
// int, Matrix, string, stb.
e = a + b + c + d;
```

Temporálisok

- Létrejönnek: a (rész)kifejezés kiértékelésekor
- Megszűnnek: a teljes kifejezés kiértékelésének végén

Példa

```
std::string h = "Hello";
std::string w = "World";
const char* p = (h + w).c_str();
std::cout << p;
```



Feladat

- Feladat: függvény implementációja
- `answer`
- Input: szövegesen megadott kérdés, `question`
- Működés: felteszi a kérdést a felhasználónak, a választ beolvassa
- Visszaadja a hívó számára: a beolvasott választ
- `std::cout « answer( "Hogy vagy?" );`

Implementáció

```
char* answer( const char* question )
{
  std::cout << question << std::endl;
  char ans[ 80 ];
  std::cin >> ans;
  return ans;
}
```

Válasz

Válasz

Implementáció

```
char* answer( const char* question )
{
  std::cout << question << std::endl;
  char ans[ 80 ];
  std::cin.getline( ans, sizeof( ans ) );
  return ans;
}
```

Implementáció

Every time you do this:

```
char* answer( const char* question )
{
  std::cout << question << std::endl;
  char ans[ 80 ];
  std::cin.getline( ans, sizeof( ans ) );
  return ans;
}
```

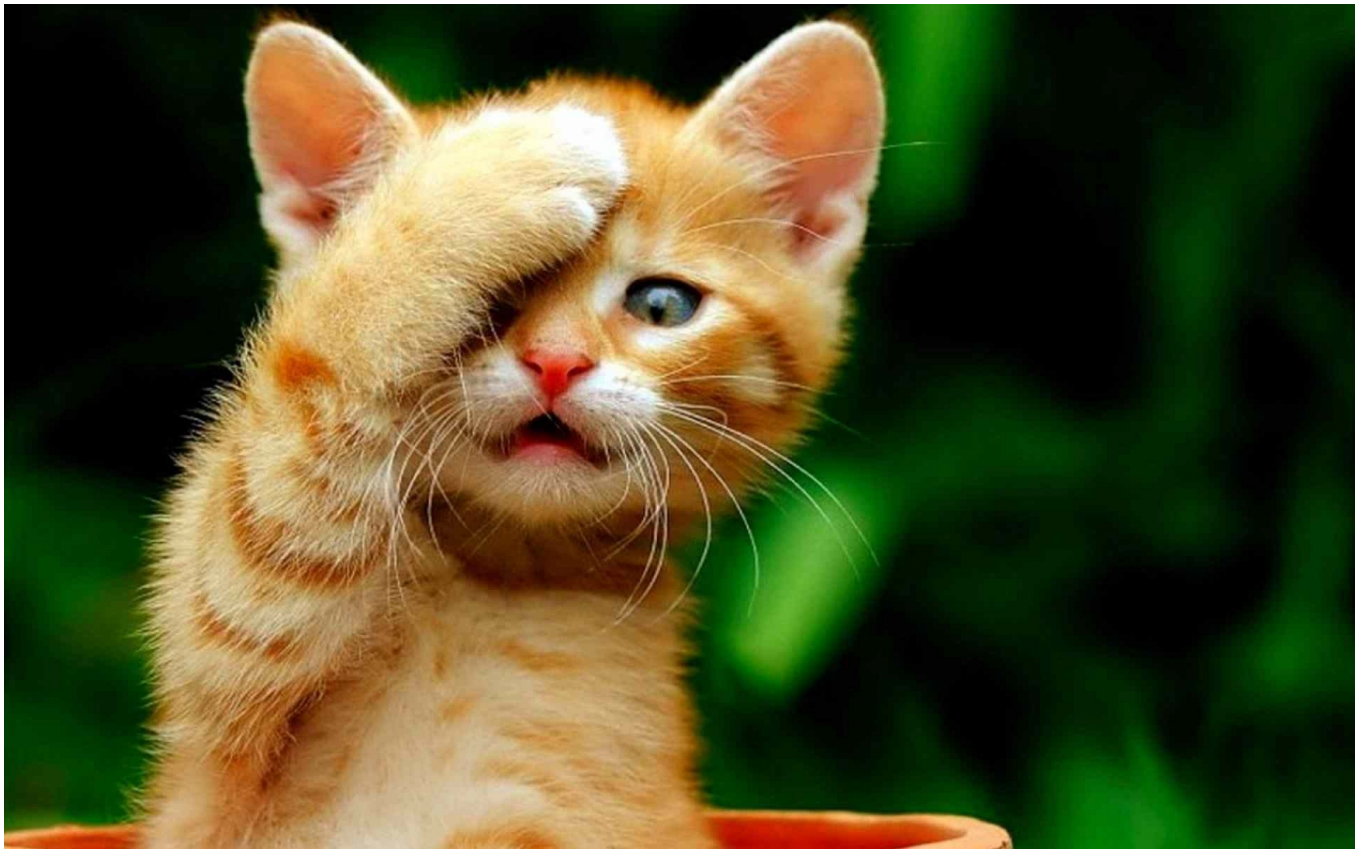
a kitten dies

Implementáció

```
char ans[ 80 ];

char* answer( const char* question )
{
  std::cout << question << std::endl;
  std::cin.getline( ans, sizeof( ans ) );
  return ans;
}
```

Válasz



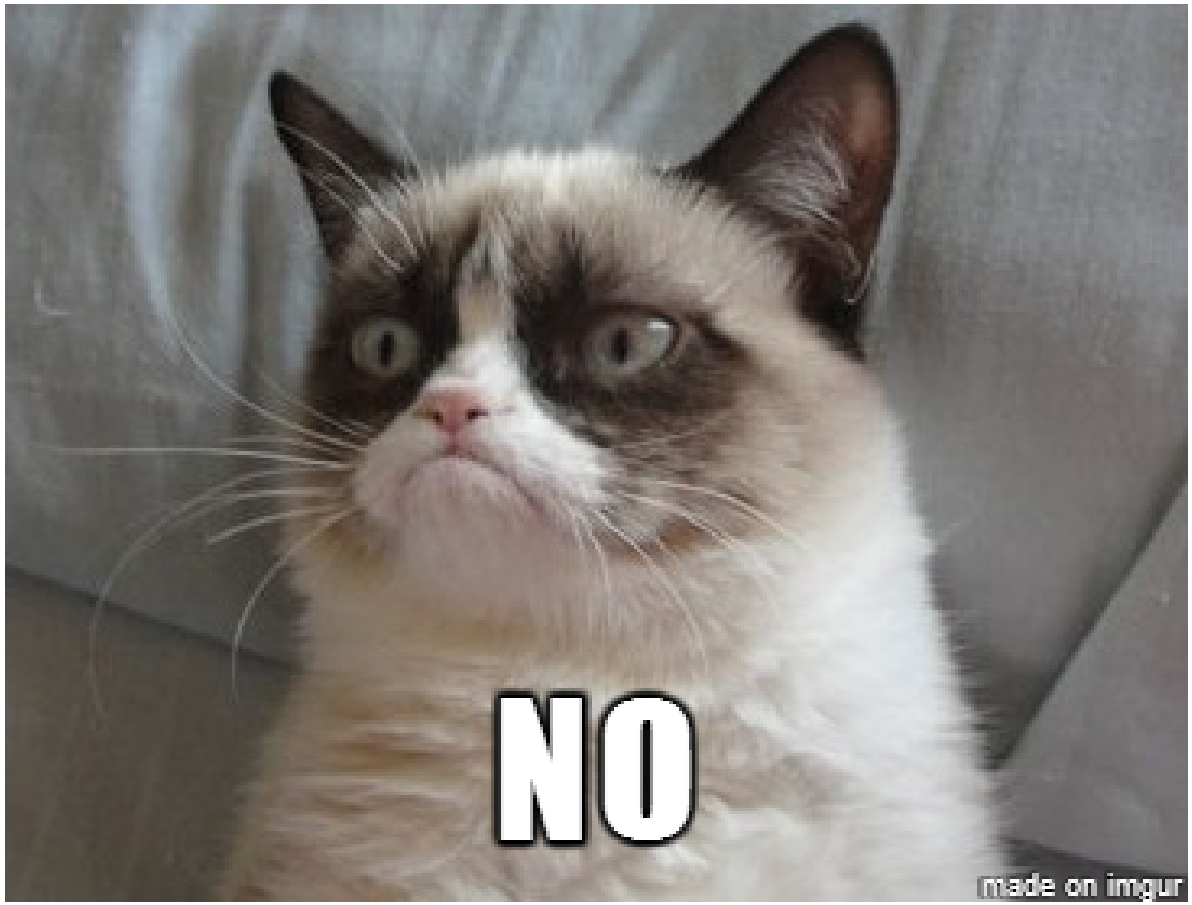
Probléma

```
std::cout << answer( "Hogy vagy?" )
          << std::endl
          << answer( "Biztos?" )
          << std::endl;
```

Implementáció

```
char* answer( const char* question )
{
  std::cout << question << std::endl;
  static char ans[ 80 ];
  std::cin.getline( ans, sizeof( ans ) );
  return ans;
}
```

Válasz



Probléma

```
std::cout << answer( "Hogy vagy?" )
          << std::endl
          << answer( "Biztos?" )
          << std::endl;
```

Implementáció

```
char* answer( const char* question )
{
  std::cout << question
            << std::endl;
  char* ans = new char[ 80 ];
  std::cin.getline( ans, 80 );
  return ans;
}
```

Válasz

Ellenőrző kérdések

Implementáció

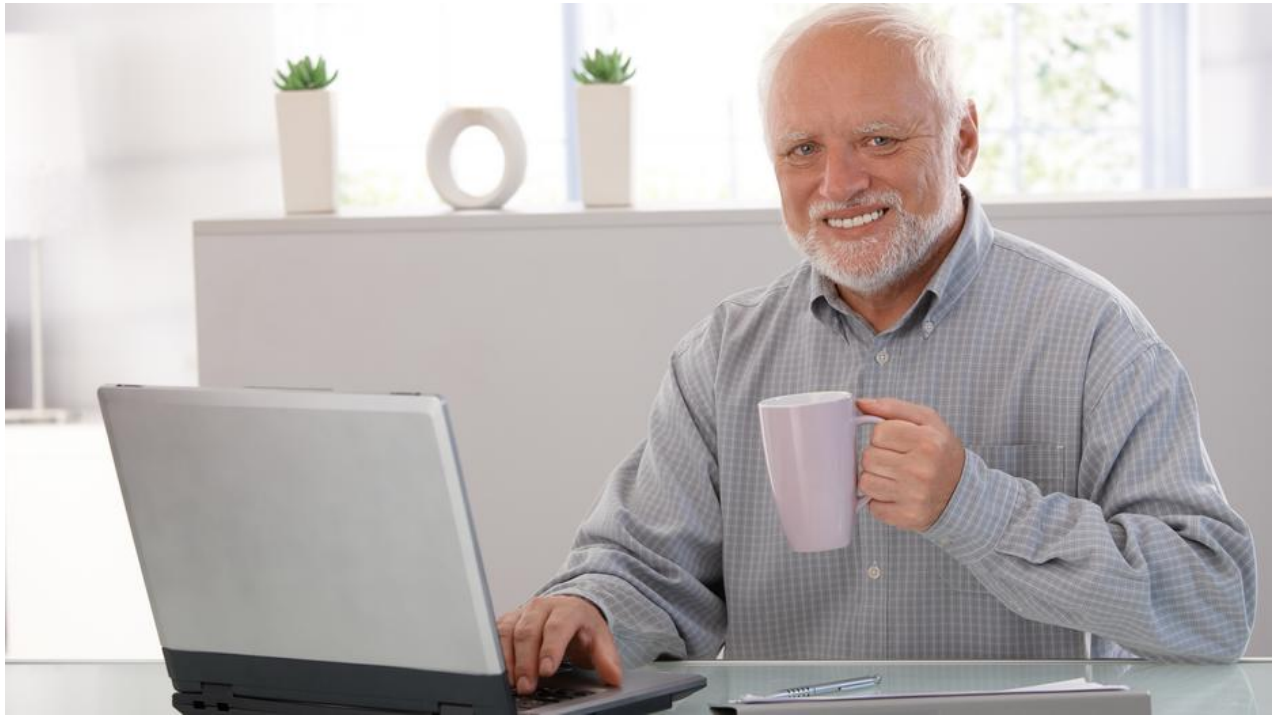
```
char* answer( const char* question,
              char* ans,
              int s )
{
  std::cout << question
            << std::endl;
  std::cin.getline( ans, s );
  return ans;
}
```

Válasz

Implementáció

```
std::string answer( const char* question )
{
  std::cout << question
            << std::endl;
  std::string ans;
  ans.getline( std::cin );
  return ans;
}
```

Válasz



Válasz

- Jelentés
- Hatékonyság
- Optimalizációk, RVO
- Move