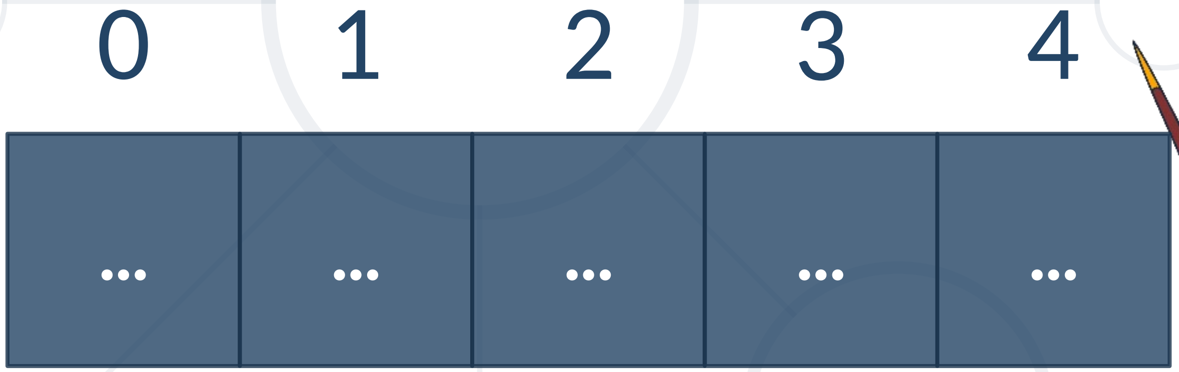
Stacks and Queues

Processing Sequences of Elements



SoftUni Team

Technical Trainers

Software University

http://softuni.bg

Table of Contents

Have a Question?

SoftUni Foundation

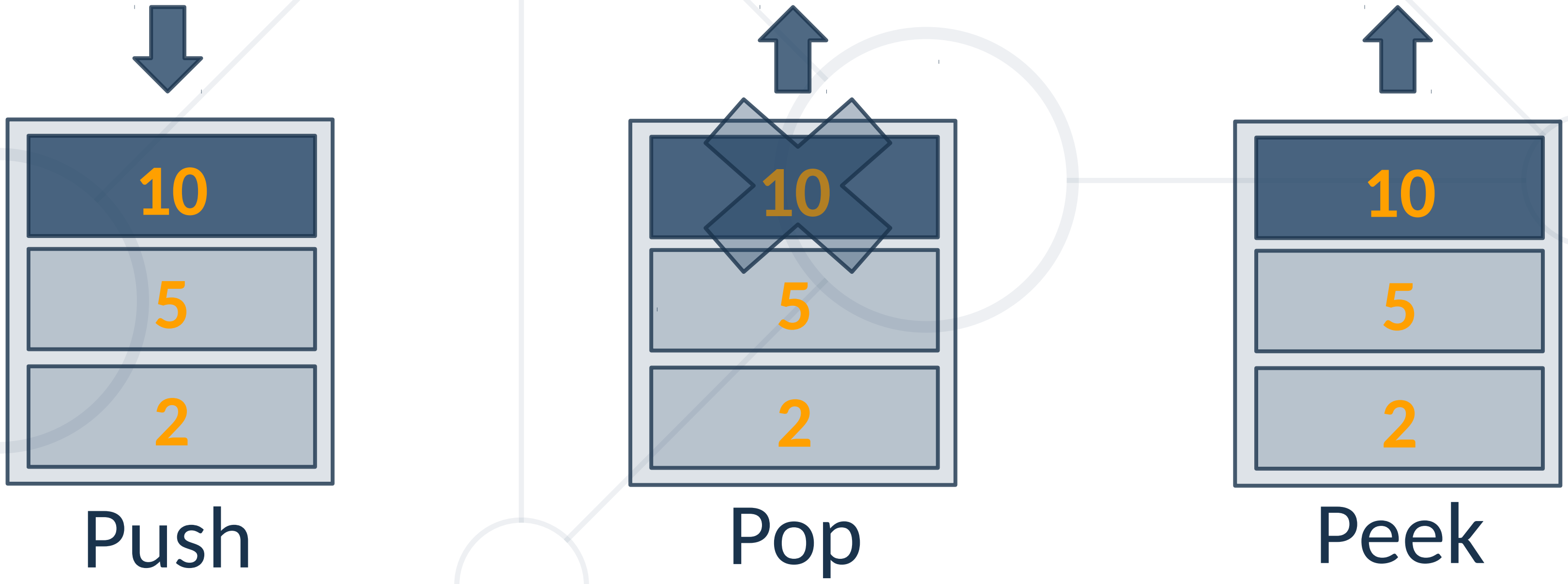
sli.do

#csharp-advanced

Stack<T>
Overview and Working with Stack

Stack – Abstract Data Type

- Stacks provide the following functionality:
  - Pushing an element at the top of the stack
  - Popping element from the top of the stack
  - Getting the topmost element without removing it



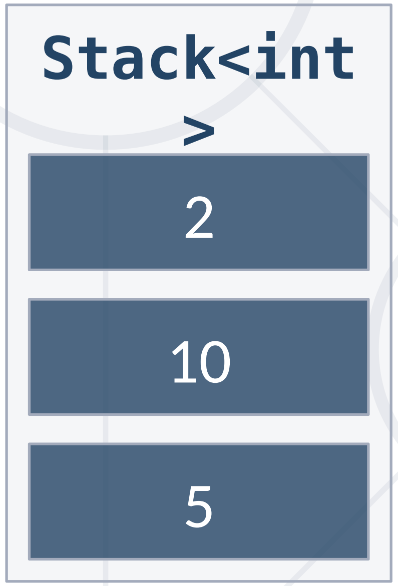
Push() – Adds an element on top of the Stack

5

`Stack<int>`

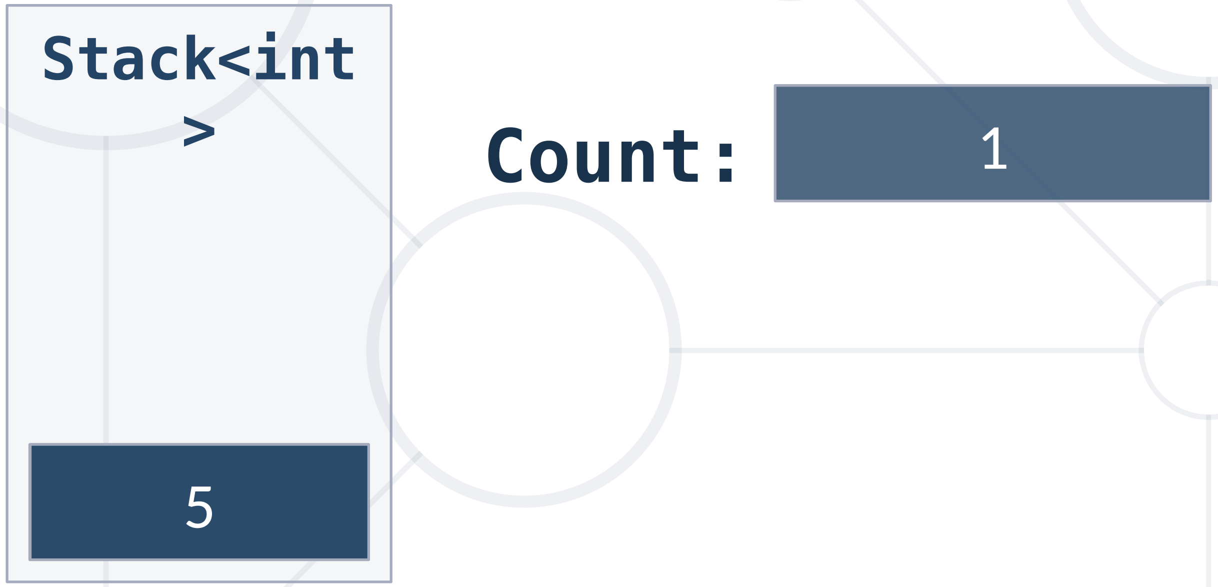
Count: 3

Pop() – Returns and removes the last element



`Count:` 2

Peek() – Returns the last element



Problem: Reverse Strings

- Create a program that:
  - Reads an input string
  - Reverses it using a Stack

```
I Love C#             →  #C evoL I
Stacks and Queues     →  seueuQ dna skcatS
```

Check your solution here: https://judge.softuni.bg/Contests/1445/Stacks-and-Queues-Lab

Solution: Reverse Strings

```
var input = Console.ReadLine();
var stack = new Stack<char>();
foreach (var ch in input)
{
  stack.Push(ch);
}
while (stack.Count != 0)
{
  Console.Write(stack.Pop());
}
Console.WriteLine();
```

Check your solution here: https://judge.softuni.bg/Contests/1445/Stacks-and-Queues-Lab

Stack – Utility Methods

```csharp
Stack<int> stack = new
Stack<int>();

int count = stack.Count;
bool exists = stack.Contains(2);
int[] array = stack.ToArray();
stack.Clear();
stack.TrimExcess();
```

Retains order of elements

Remove all elements

Resize the internal array

Problem: Simple Calculator

- Implement a simple calculator that can evaluate simple expressions (only addition and subtraction)

`2 + 5 + 10 – 2 - 1`	`14`
`2 – 2 + 5`	`5`
`2 – 1 + 5`	`6`
`2 – 0 + 5`	`7`

Check your solution here: https://judge.softuni.bg/Contests/1445/Stacks-and-Queues-Lab

Solution: Simple Calculator

```
var input = Console.ReadLine();
var values = input.Split(' ');
var stack = new Stack<string>(values.Reverse());
while (stack.Count > 1)
{
  int first = int.Parse(stack.Pop());
  string operator = stack.Pop();
  int second = int.Parse(stack.Pop());
  //TODO: Add switch for operation (look next
slide)
}
Console.WriteLine(stack.Pop());
```

Check your solution here: https://judge.softuni.bg/Contests/1445/Stacks-and-Queues-Lab

Solution: Simple Calculator

```
switch (operator)
{
  case "+":
    stack.Push((first +
second).ToString());
    break;
  case "-":
    stack.Push((first -
second).ToString());
    break;
}
```

Check your solution here: https://judge.softuni.bg/Contests/1445/Stacks-and-Queues-Lab

Problem: Stack Sum

- Calculate the sum in the stack
- Before that you will receive commands
  - Add – adds the two numbers
  - Remove – removes count numbers

```
1 2 3 4
adD 5 6
REmove 3
eNd
```

```
Sum: 6
```

```
3 5 8 4 1
9
add 19 32
remove 10
add 89 22
end
```

```
Sum: 192
```

Check your solution here: https://judge.softuni.bg/Contests/1445/Stacks-and-Queues-Lab

Solution: Stack Sum

```
var input =
Console.ReadLine().Split().Select(int.Parse).ToArr
ay();
Stack<int> stack = new Stack<int>(input);
var commandInfo = Console.ReadLine().ToLower();

while (commandInfo != "end"){
  var tokens = commandInfo.Split();
  var command = tokens[0].ToLower();
  if (command == "add")
    // Parse the numbers and add them
}
```

Check your solution here: https://judge.softuni.bg/Contests/1445/Stacks-and-Queues-Lab

Solution: Stack Sum

```
else if(command == "remove") {
    var countOfRemovedNums = int.Parse(tokens[1]);
    if (stack.Count < countOfRemovedNums)
{ continue; }
    for (int i = 0; i < countOfRemovedNums; i++) {
      stack.Pop();
    }
  }
  commandInfo = Console.ReadLine().ToLower();
}
var sum = stack.Sum();
Console.WriteLine($"Sum: {sum}");
```

Check your solution here: https://judge.softuni.bg/Contests/1445/Stacks-and-Queues-Lab

Problem: Matching Brackets

- We are **given an arithmetic expression** with brackets (**with nesting**)
- **Extract all sub-expressions** in brackets

```
1 + (2 - (2 + 3) * 4 / (3 + 1))
* 5
```

```
(2 + 3)
(3 + 1)
(2 - (2 + 3) * 4 / (3 +
1))
```

Check your solution here: https://judge.softuni.bg/Contests/1445/Stacks-and-Queues-Lab

Solution: Matching Brackets

SoftUni Foundation

```
var input = Console.ReadLine();
var stack = new Stack<int>();
for (int i = 0; i < input.Length; i++) {
  char ch = input[i];
  if (ch == '(') {
    stack.Push(i);
  } else if (ch == ')') {
    int startIndex = stack.Pop();
    string contents = input.Substring(
                         startIndex, i - startIndex
+ 1);
    Console.WriteLine(contents);
}
```

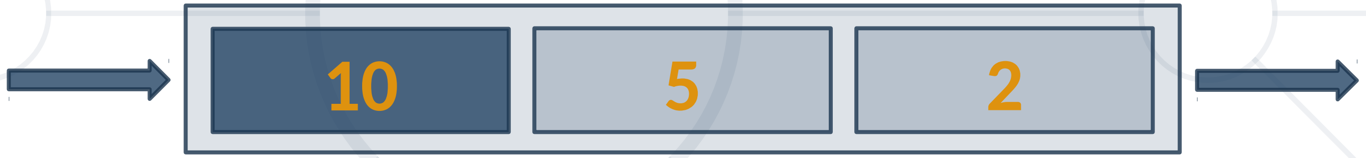
Check your solution here: https://judge.softuni.bg/Contests/1445/Stacks-and-Queues-Lab

Queue<T>

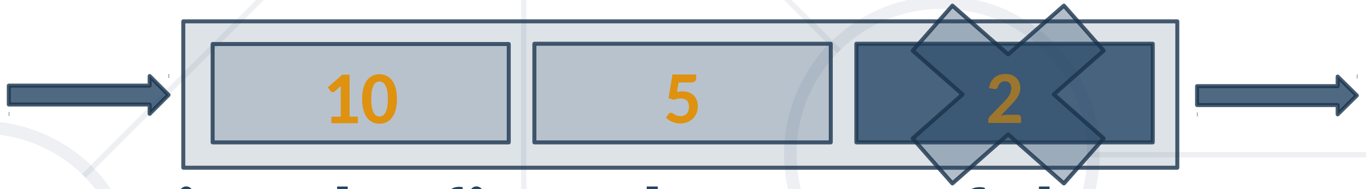
Overview and Working with Queue

Queue – Abstract Data Type

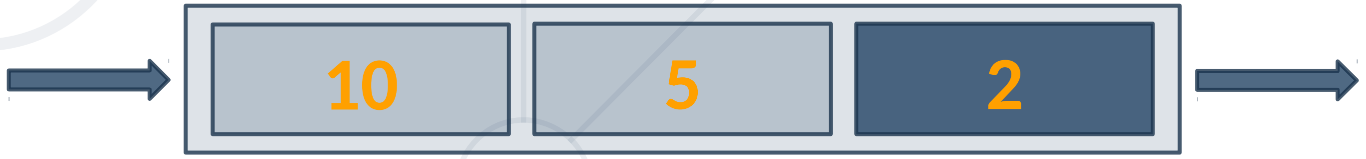
- Queues provide the following functionality:
  - Adding an element at the end of the queue



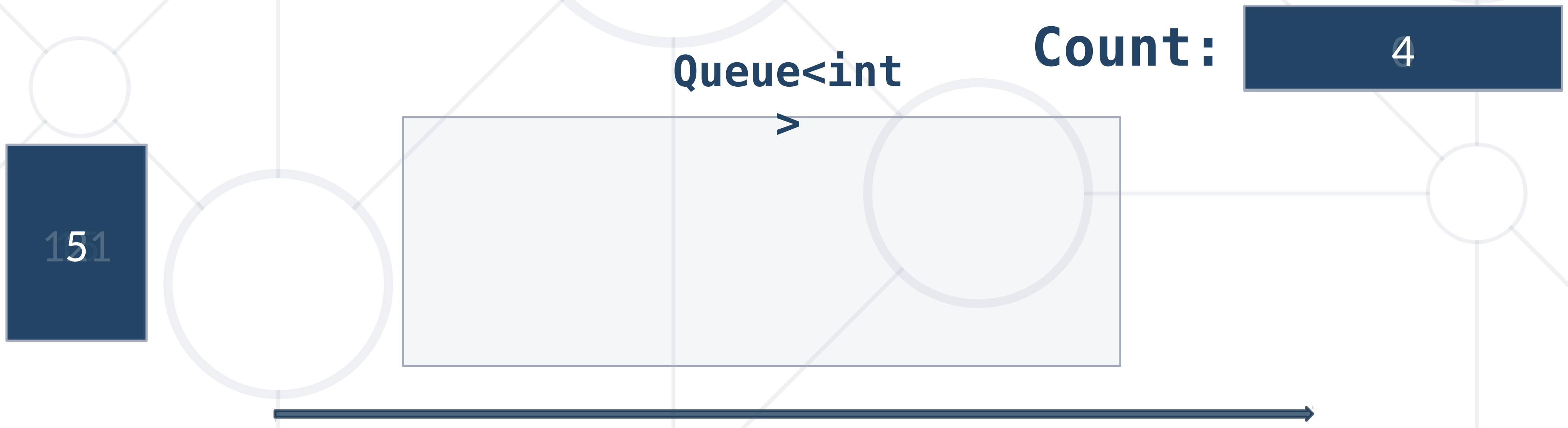
  - Removing the first element from the queue



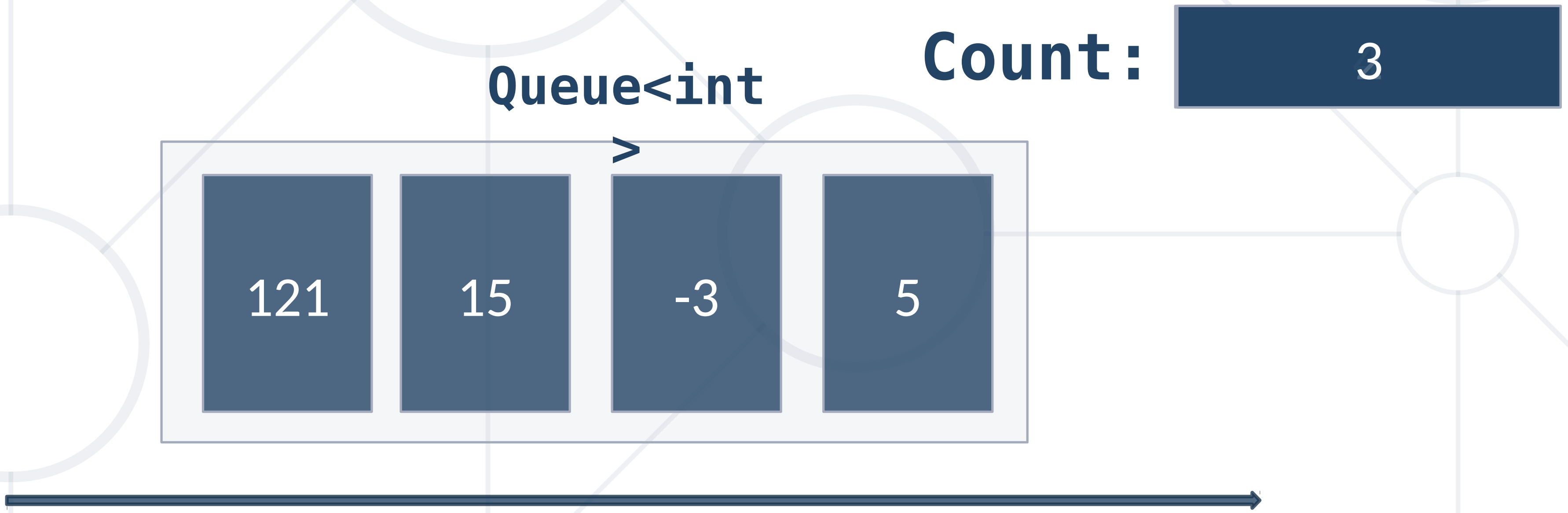
  - Getting the first element of the queue without removing it



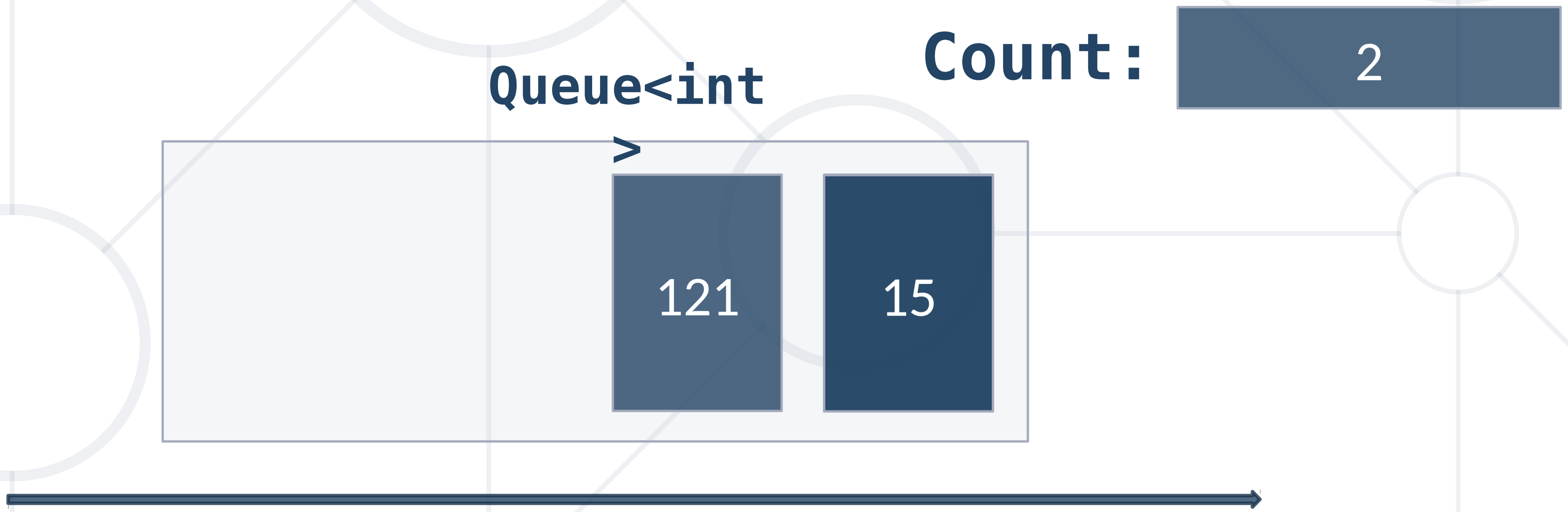
Enqueue() – Adds an element to the front



Dequeue() – Returns and removes the first element



Peek() – Returns the first element



Problem: Hot Potato

- Children **form a circle** and pass a hot potato **clockwise**
- Every nth toss **a child is removed** until **only one remains**
- **Upon removal** the potato is passed **along**
- Print the child that remains last

```
Mimi Pepi
Toshko
2
```

→

```
Removed Pepi
Removed Mimi
Last is
Toshko
```

Check your solution here: https://judge.softuni.bg/Contests/1445/Stacks-and-Queues-Lab

Solution: Hot Potato

```
var children = Console.ReadLine().Split(' ');
var number = int.Parse(Console.ReadLine());
Queue<string> queue = new
Queue<string>(children);
while (queue.Count != 1) {
  for (int i = 1; i < number; i++) {
    queue.Enqueue(queue.Dequeue());
  }
  Console.WriteLine($"Removed
{queue.Dequeue()}");
}
Console.WriteLine($"Last in
{queue.Dequeue()}");
```

Copies elements from the specified collection and keeps their order

Check your solution here: https://judge.softuni.bg/Contests/1445/Stacks-and-Queues-Lab

Queue – Utility Methods

```
Queue<int> queue = new
Queue<int>();

int count = queue.Count;
bool exists = queue.Contains(2);
int[] array = queue.ToArray();
queue.Clear();
queue.TrimExcess();
```

Retains order of elements

Remove all elements

Resize the internal array

Problem: Traffic Jam

- Cars are **queuing up** at a **traffic light**
- Every **green light** n cars pass the crossroads
- After the **end command**, print **how many cars** have **passed**

```
3
Pesho's car
Gosho's car
Mercedes CLS
Nekva troshka
green
BMW X5
green
end
```

```
Pesho's car passed!
Gosho's car passed!
Mercedes CLS passed!
Nekva troshka passed!
BMW X5 passed!
5 cars passed the crossroads.
```

Check your solution here: https://judge.softuni.bg/Contests/1445/Stacks-and-Queues-Lab

Solution: Traffic Jam

```
int n = int.Parse(Console.ReadLine());
var queue = new Queue<string>();
int count = 0;
string command;
while ((command = Console.ReadLine()) != "end")
{
  if (command == "green")
    //TODO: Add green light logic
  else
    queue.Enqueue(command);
}
Console.WriteLine($"{count} cars passed the
crossroads.");
```

Check your solution here: https://judge.softuni.bg/Contests/1445/Stacks-and-Queues-Lab

Summary

- Stack<T>
  - **LIFO** data structure
- Queue<T>
  - **FIFO** data structure
- Working with **built-in methods**

Questions?

SoftUni

Software University

SoftUni Svetlina

SoftUni Creative

SoftUni Digital

SoftUni Foundation

SoftUni Kids

SoftUni Diamond Partners

SoftUni Foundation

XSsoftware

SBTech we know sports

telenor

SoftwareGroup doing it right

NETPEAK

SmartIT

Postbank Решения за твоето утре

SUPERHOSTING.BG

INDEAVR Serving the high achievers

INFRAGISTICS

LIEBHERR

æternity

SoftUni Organizational Partners

SoftUni Foundation

IS ИНФОРМАЦИОННО ОБСЛУЖВАНЕ

OneBit SOFTWARE

Lukanet.com

WORLD OF MYTHS

Trainings @ Software University (SoftUni)

- Software University – High-Quality Education and Employment Opportunities
  - softuni.bg
- Software University Foundation
  - http://softuni.foundation/
- Software University @ Facebook
  - facebook.com/SoftwareUniversity
- Software University Forums
  - forum.softuni.bg

License