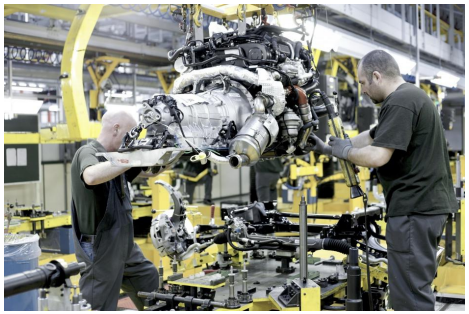
Tipo Abstrato de Dados

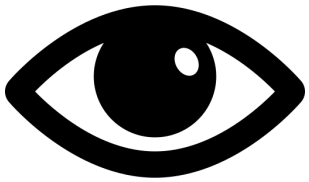
Prof. Martín Vigil

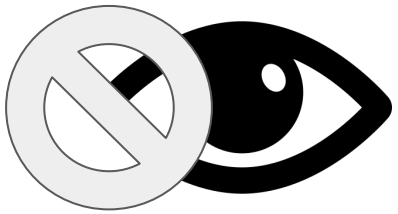
Interface vs. Implementação no dia-a-dia



Interface vs. Implementação no dia-a-dia

Caixa Preta

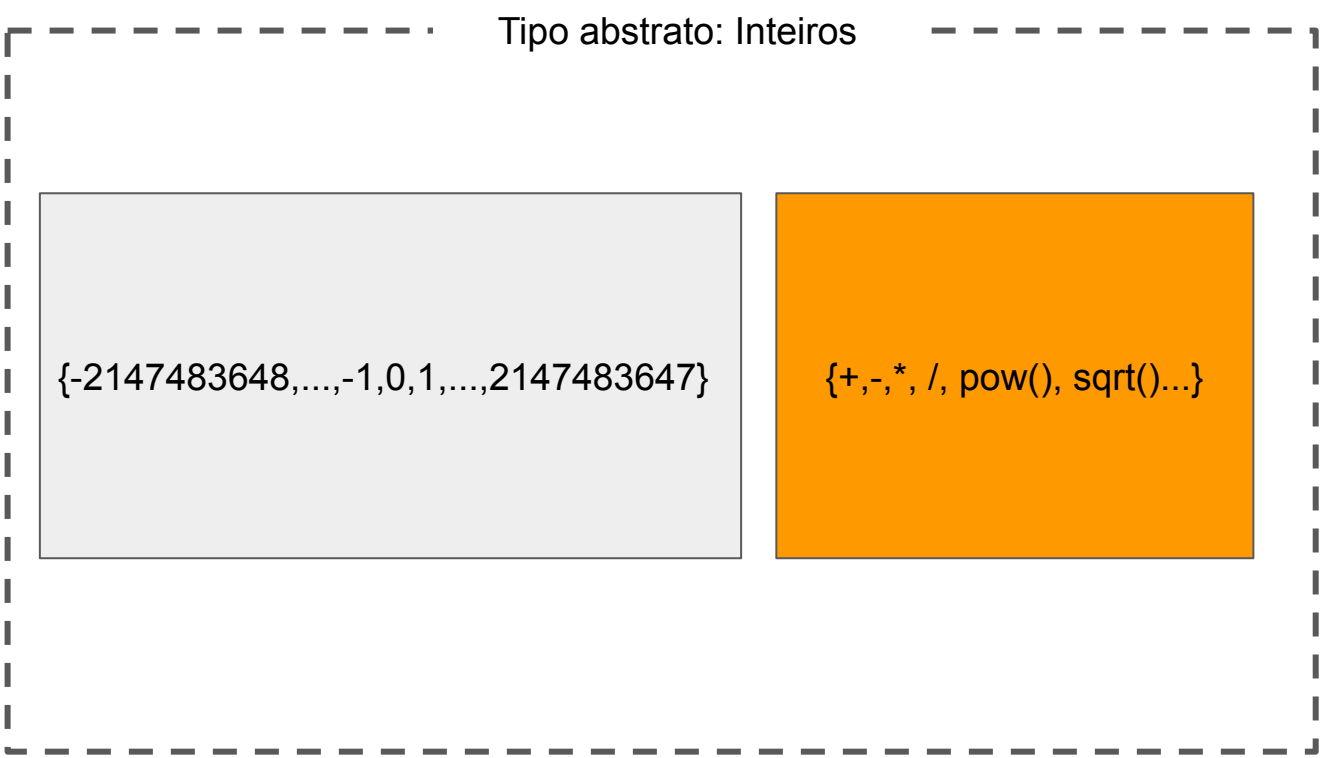


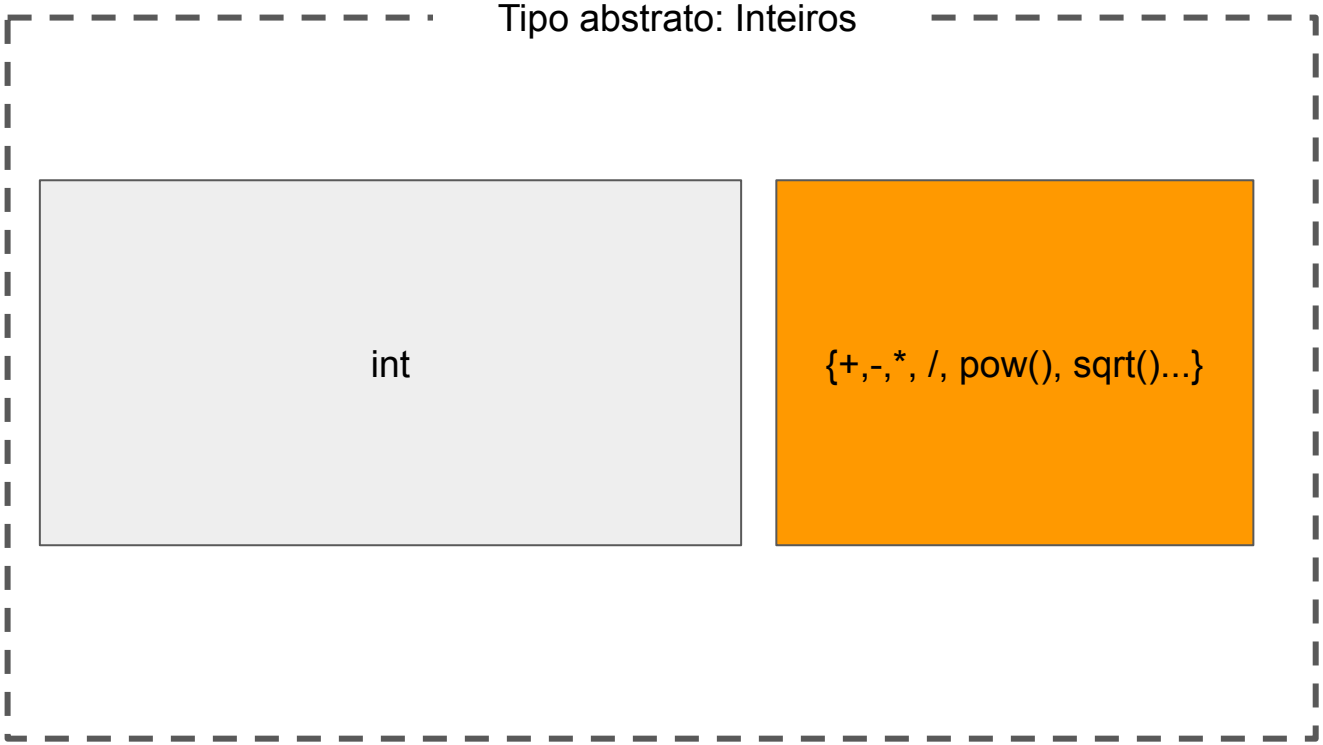


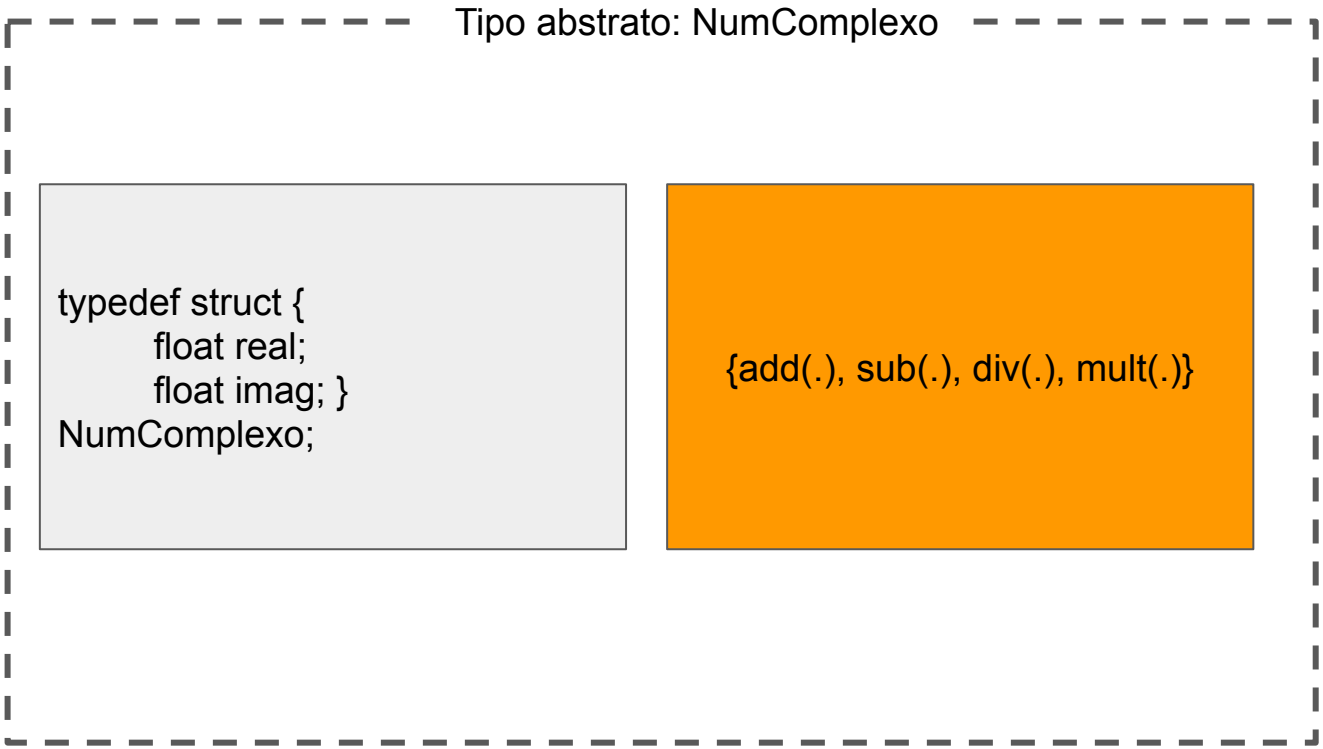
Tipo abstrato de dados

Valores possíveis para dados

Operações sobre dados


Tipo abstrato: Inteiros
{-2147483648,...,-1,0,1,...,2147483647}
{+,-,*, /, pow(), sqrt()...}


Tipo abstrato: Inteiros
int
{+,-,*, /, pow(), sqrt()...}


Tipo abstrato: NumComplexo
typedef struct {
float real;
float imag; }
NumComplexo;
{add(.), sub(.), div(.), mult(.)}

Tipo abstrato: NumComplexo.h

```
typedef struct
{
        float real;
        float imag;
} NumComplexo;
```

```
NumComplexo* add(NumComplexo a, NumComplexo b);
NumComplexo* sub(NumComplexo a, NumComplexo b);
NumComplexo* mult(NumComplexo a, NumComplexo b);
NumComplexo* div(NumComplexo a, NumComplexo b);
```

Tipo abstrato: NumComplexo.h

```
typedef struct
{
        float real;
        float imag;
} NumComplexo;

NumComplexo* add(NumComplexo a, NumComplexo b);
NumComplexo* sub(NumComplexo a, NumComplexo b);
NumComplexo* mult(NumComplexo a, NumComplexo b);
NumComplexo* div(NumComplexo a, NumComplexo b);
```

Tipo concreto: MeuNumComplexo.c

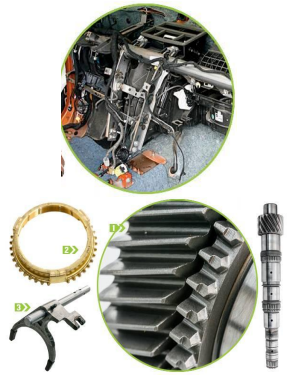
```
#include "NumComplexo.h"

NumComplexo* add(NumComplexo a, NumComplexo b){
        NumComplexo* num = (NumComplexo*) calloc(1,sizeof(NumComplexo));
        num->real = a.real + b.real;
        num->imag = a.imag + b.imag;
        return num;
}
//continua
```

Tipo abstrato: NumComplexo.h

```
typedef struct
{
        float real;
        float imag;
} NumComplexo;

NumComplexo* add(NumComplexo a, NumComplexo b);
NumComplexo* sub(NumComplexo a, NumComplexo b);
NumComplexo* mult(NumComplexo a, NumComplexo b);
NumComplexo* div(NumComplexo a, NumComplexo b);
```

Tipo concreto: MeuNumComplexo.c

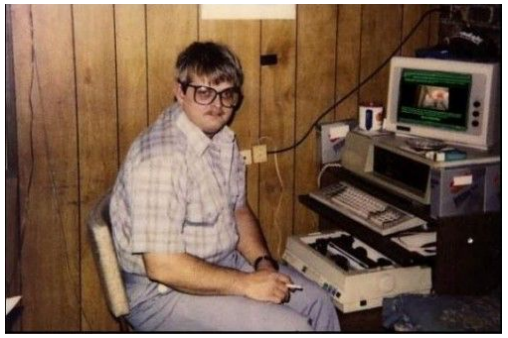
```
#include "NumComplexo.h"

NumComplexo* add(NumComplexo a, NumComplexo b){
        NumComplexo* num = (NumComplexo*) calloc(1,sizeof(NumComplexo));
        num->real = a.real + b.real;
        num->imag = a.imag + b.imag;
        return num;
}
//continua
```



Tipo abstrato: NumComplexo.h

```
typedef struct
{
        float real;
        float imag;
} NumComplexo;
```

```
NumComplexo* add(NumComplexo a, NumComplexo b);
NumComplexo* sub(NumComplexo a, NumComplexo b);
NumComplexo* mult(NumComplexo a, NumComplexo b);
NumComplexo* div(NumComplexo a, NumComplexo b);
```



Tipo concreto: MeuNumComplexo.c

```
#include "NumComplexo.h"

NumComplexo* add(NumComplexo a, NumComplexo b){
        NumComplexo* num = (NumComplexo*) calloc(1,sizeof(NumComplexo));
        num->real = a.real + b.real;
        num->imag = a.imag + b.imag;
        return num;
}
//continua
```

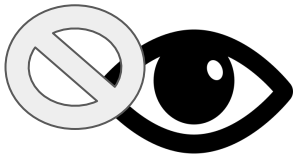
Tipo abstrato: NumComplexo.h

```
typedef struct
{
        float real;
        float imag;
} NumComplexo;
```

```
NumComplexo* add(NumComplexo a, NumComplexo b);
NumComplexo* sub(NumComplexo a, NumComplexo b);
NumComplexo* mult(NumComplexo a, NumComplexo b);
NumComplexo* div(NumComplexo a, NumComplexo b);
```

Tipo concreto: MeuNumComplexo.c

Caixa Preta



Tornando a estrutura *opaca* para *esconder* valores

Tipo abstrato: NumComplexo.h

```
typedef struct
{
        void * valoresPrivados
} NumComplexo;
```

```
NumComplexo* create(float real, float imag);
NumComplexo* add(NumComplexo a, NumComplexo b);
NumComplexo* sub(NumComplexo a, NumComplexo b);
NumComplexo* mult(NumComplexo a, NumComplexo b);
NumComplexo* div(NumComplexo a, NumComplexo b);
```

Classes e encapsulamento

Tipo abstrato: CNumComplexo.hpp

```
class CNumImaginario {
        private :
                float real;
                float imag;
        public :
                CNumImaginario(float real, float imag);
                CNumImaginario add(CNumImaginario a, CNumImaginario b);
                CNumImaginario sub(CNumImaginario a, CNumImaginario b);
                CNumImaginario mult(CNumImaginario a, CNumImaginario b);
                CNumImaginario div(CNumImaginario a, CNumImaginario b)
};
```

Síntese da aula

Bibliografia

Tenenbaum et al., "Estruturas de dados usando C", 1995.