

Templates

Dienstag, 16. April 2024 13:30



templates (2)

```
//templates.h
#ifndef _templates_h
#define _templates_h
#include<iostream>
#include<string>

using namespace std;

template<class T>
class beispiel
{
private:
    T variable;
    T* array; // Zeiger auf ein Array
    T vektor[4];
    int i; //Anzahl der
           // Speicherplätze in array

public:
    void eingabeVariable(T);
    void eingabeArray(T,int);
    void eingabeVektor(T,int);

    T rueckgabeVariable();
    T rueckgabeArray(int);
    T rueckgabeVektor(int);

    beispiel(int n){array = new T[n]; i = n;};

};
#endif
```

```
//templates.cpp

#include "templates.h"

template<class T>
void beispiel<T>::eingabeVariable(T var)
{
    variable = var;
}

template<class T>
void beispiel<T>::eingabeArray(T zahl,int
n)
{
    array[n]= zahl;
}

template<class T>
void beispiel<T>::eingabeVektor(T zahl,int
n)
{
    vektor[n]= zahl;
}

template<class T>
T beispiel<T>::rueckgabeVariable()
{
    return variable;
}

template<class T>
T beispiel<T>::rueckgabeArray(int n)
{
    return array[n];
}

template<class T>
T beispiel<T>::rueckgabeVektor(int n)
{
    return vektor[n];
}
```

```
//main.cpp
//main.cpp
#include "templates.h"
#include "templates.cpp"
#include <iostream>
using namespace std;

int main()
{
    beispiel<double> sinnlos(4);
    double temp =0;
    int i=0;
    //Eingabe von Werten
    cout<<"Geben Sie eine Zahl ein: ";
    cin>>temp;
    sinnlos.eingabeVariable(temp);

    for(i=0;i<4;i++)
    {
        cout<<"Geben Sie eine Zahl fuer den Array ei
        cin>>temp;
        sinnlos.eingabeArray(temp,i);
    }
    for(i=0;i<4;i++)
    {
        cout<<"Geben Sie eine Zahl fuer den Vektor ei
        cin>>temp;
        sinnlos.eingabeVektor(temp,i);
    }
    //werte wieder ausgeben
    cout<<sinnlos.rueckgabeVariable()<<"\n"<<endl;

    for(i=0;i<4;i++)
    {
        cout<<sinnlos.rueckgabeArray(i)<<endl;

        cout<<"\n"<<endl;

    for(i=0;i<4;i++)
    {
        cout<<sinnlos.rueckgabeVektor(i)<<endl;

        getchar();
        return 0;
    }
}
```

