

This document contains information about laboratory exercises.

Object programming - classes 5

Subjects - Friend functions, function, class templates.

1. Friend functions

1. Open file *safe.cpp*
 - a) Add printing on a screen list of element of class *Safe* in definition of overloaded operator `<<`.
 - b) Add to class *safe* friend function

```
int passkey(Safe safe,int position)
```

that will return contents of a given box. Do not use *getElement()* method! Check if it works.

2. Templates

1. Define new specialized function that will sum up two argument of type *const int&*.

```
int sum(const int& a,const int& b);
```
2. Generalize function above by define a template function

```
type sum(type a,type b);
```

, which sum up two elements of the same type.
3. Invoke *sum(2.3,2)* what will be the result ? What will change if you add line

```
template<>
```

before the definition of the template function *sum*??
4. Change declaration of class *Safe* to class template, that will allow to hide in boxes elements of any type. Comment in code definition and declaration of method *passkey*.
5. Define synonym for class name *Safe < int >* and call it *SmallBox* (hint: use typedef)
6. Overload operator `+` for adding *const int&* and elements of class *Safe < class T >* in such a way that it will return sum of an integer and all elements from boxes in given *safe*. Change function template *sum* such that it will work for pair of elements of different types.

3. Do it yourself tasks

1. For class `Safe` add definition of *operator+*, that sum up elements from two different `Safes`.
2. Try to define template of class that will inherits another class.
3. Change class template such that beside the type of elements in safe you could define also numbers for boxes in given safe (so *template < class T, int size > Safe*)
4. What to do if you want to have in class template a friend function ? (for example function *passkey* from taks above) ??