This document contains information about labolatory exercises.

Object programming - classes 4

Subjects - Namespace, operator overloading, type conversion.

1. Namespace

1. Download and open file *wallet.cpp*
   a) Delete the line from a code
      ```
      using namespace std;
      ```
      and correct errors so program will compile and will work correctly.
   b) Put definition of class Wallet and its methods into namespace *MySpace*.
   c) Inside namespace *MySpace* add line
      ```
      using namespace std;
      ```
      Could you then delete prefix *std* :: inside namespace *Myspace*, method *main*() ?

2. Type conversion

1. Define new constructor of class Wallet $Wallet(const\ char*)$. Invoke method walletState with parameter string (ex. "string"), what will happend? How to prevent such an effect?
2. Define conversion using operator (int) in such a way that it will return total amount of money in a wallet.

3. Operator overloading

1. Define in class Wallet assignment operator =.
2. Overload operators $<<$ i $>>$ for int variables so that:
   — - operator $<<$ add money to Wallet;
   — - operator $>>$ empty wallet from all money and save the amount in *int* variable.
3. Overload operator $<<$ for object $std :: ostream\&$ (ex. cout) so that it will write to standard output total amount of money and the name of currency
   ```
   ostream& operator<<(ostream& os,Porfel& portfel)
   ```
   Check if it works when you type $cout << portfel$.
4. Add to class Wallet definition of comparing operators $<, >$, that will say in which wallet is more money.
5. Define array with 5 different Wallets. Using method *sort* from library $< algorithm >$ (use google to find out how it works) sort Wallets in array. Write out sorted array.

4. Do it yourself tasks

1. Define *operator+* for two objects of class Wallet in such a way that it will sum up the amount of money if the currency is the same.
2. Define in new namespace the same class Wallet. Overload *operator <* to compare two Wallet objects from different namespaces. Try to sort array with this two types of Wallets.