**МИНОБРНАУКИ РОССИИ**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ  
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»  
(БГТУ им. В.Г.Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Лабораторная работа №8  
дисциплина: Объектно-ориентированное программирование

тема: «**Создание шаблонов классов в С++»**

Выполнил: ст. группы ВТ-22  
Егоров Владислав Юрьевич  
Проверил: Буханов Д. Г.

Белгород 2020

**Цель работы:** Получение теоретических знаний о шаблонах классов в С++. Получение практических навыков по созданию классов-шаблонов С++.

Задания к лабораторной работе

1. Изучить теоретические сведения о шаблонах классов в С++.
2. Разработать программу в соответствии с заданным вариантом задания.
3. Оформить отчет.

Задание 1

Реализовать шаблон класса в соответствии с указанным вариантом. Предусмотреть необходимые методы для работы со структурой данных, указанной в варианте. Предусмотреть исключительные ситуации, которые могут возникнуть в процессе работы.

Задание 4

Множество

1. #include <iostream>
2. #include <locale.h>
3. using namespace std;
4. class SetException : public exception
5. {
6. protected:
7. string message;
8. public:
9. SetException(string m):message(m) {};
10. const char\* what()
11. {
12. return message.c\_str();
13. }
14. };
15. class SetBadAlloc : public SetException
16. {
17. public:
18. SetBadAlloc()
    * 1. :SetException("Incorrectly allocated memory") {};
19. };
20. class SetEmpty : public SetException
21. {
22. SetEmpty()
    * 1. :SetException("Set is empty") {};
23. };
24. template <class T>
25. class Set
26. {
27. private:
28. int size;
29. T\* array;
30. public:
31. Set()
32. {
    * 1. size = 0;
      2. array = nullptr;
33. };
34. Set(int s)
35. {
    * 1. size = s;
      2. array = new T[size];
      3. if (array == nullptr)
         1. throw SetBadAlloc();
      4. for (int i = 0;i < size;i++)
         1. array[i] = 0;
36. };
37. void Show()
38. {
    * 1. cout << "Множество: ";
      2. if (size == 0)
      3. {
         1. cout << "пустое" << endl;
         2. return;
      4. }
      5. for (int i = 0; i < size; i++)
         1. cout << array[i] << " ";
      6. cout << endl;
39. };
40. Set<T> operator+(Set<T> a)
41. {
    * 1. int i, j, count = 0;
      2. int tsize = size + a.size;
      3. Set<T> b(tsize);
      4. for (int i = 0;i < size;i++)
         1. b.array[i] = array[i];
      5. int n = size;
      6. for (i = 0; i < a.size; i++)
      7. {
         1. for (j = 0; j < size; j++)
            1. if (b.array[j] != a.array[i])

count++;

* + - 1. if (count == size)
         1. b.array[n++] = a.array[i];
      2. count = 0;
    1. }
    2. b.size = n;
    3. return b;

1. };
2. Set<T> operator+(T a)
3. {
   * 1. size++;
     2. Set<T> b(size);
     3. for (int i = 0;i < size - 1;i++)
        1. b.array[i] = array[i];
     4. b.array[size - 1] = a;
     5. return b;
4. };
5. Set<T> operator\*(Set<T> a)
6. {
   * 1. int i, j, n = 0;
     2. for (i = 0; i < size; i++)
     3. {
        1. for (j = 0; j < a.size; j++)
           1. if (array[i] == a.array[j])
           2. array[n++] = array[i];
     4. }
     5. size = n;
     6. return \*this;
7. };
8. Set<T> operator-(Set<T> a)
9. {
   * 1. if (isEmpty())
        1. throw SetEmpty();
     2. int i, j, n=size;
     3. for (i = 0; i < size; i++)
     4. {
        1. for (j = 0; j < a.size; j++)
           1. if (array[i] == a.array[j])
           2. {

for (int k = i;k < size;k++)

array[k] = array[k + 1];

n--;

* + - * 1. }
    1. }
    2. size = n;
    3. return \*this;

1. };
2. Set<T> operator-(T a)
3. {
   * 1. for(int i=0;i<size;i++)
        1. if (array[i] == a)
        2. {
           1. for (int j = i;j < size;j++)

array[j] = array[j + 1];

* + - 1. }

1. };
2. bool operator<=(Set<T> a)
3. {
   * 1. int count = 0;
     2. for (int i = 0;i < size;i++)
        1. for (int j = 0;j < a.size;j++)
           1. if (array[i] == a.array[j])

count++;

* + 1. return (count == size);

1. };
   1. bool operator>=(T a)
2. {
   * 1. for (int i = 0;i < size;i++)
        1. if (array[i] == a)
           1. return true;
     2. return false;
3. };
   1. bool operator==(Set<T> a)
4. {
   * 1. return(\*this <= a && a <= \*this);
5. }
6. bool operator!=(Set<T> a)
7. {
   * 1. return(\*this <= a && a <= \*this);
8. }
9. };
10. }

Задание 2.

На основе разработанного шаблона решить прикладную задачу в соответствии с выбранным вариантом?

2. Дан файл содержащий текст, который представляет собой различные предложения. Считать предложения из текста в разработанную структуру данных таким образом, чтобы первыми словами были существительные, далее глаголы, затем все остальное. Для получения информации к кокой части речи принадлежит слово использовать справочники, хранящиеся в других файлах(использовать те же разработанные шаблоны).

1. #include "SET.h"
2. #include <fstream>
3. class FileOfProgram
4. {
5. protected:
6. string fname;
7. Set<string> wordsOfFile;
8. public:
9. FileOfProgram(string \_fname) : fname(\_fname) {};
10. virtual void readContentOfFile() = 0;
11. void writeList() { wordsOfFile.Show(); };
12. };
13. class Directory : public FileOfProgram
14. {
15. public:
16. Directory(string \_fname) : FileOfProgram(\_fname) {};
17. void readContentOfFile()
18. {
    * 1. ifstream file(fname);
      2. string word;
      3. while (!file.eof())
      4. {
         1. file >> word;
         2. wordsOfFile=wordsOfFile+word;
      5. }
      6. file.close();
19. };
20. bool inCatalog(string word) { return wordsOfFile>=word; };
21. };
22. class FileOfWord : public FileOfProgram
23. {
24. Directory nouns;
25. Directory verbs;
26. Directory others;
27. public:
28. FileOfWord(string dirNouns, string dirVerbs, string dirOthers, string \_fname)
    * 1. : nouns(dirNouns), verbs(dirVerbs), others(dirOthers), FileOfProgram(\_fname)
29. {
    * 1. nouns.readContentOfFile();
      2. verbs.readContentOfFile();
      3. others.readContentOfFile();
30. };
31. void readContentOfFile()
32. {
    * 1. ifstream file(fname);
      2. string word;
      3. while (!file.eof())
      4. {
         1. file >> word;
         2. if (nouns.inCatalog(word))
            1. wordsOfFile = wordsOfFile + word;
      5. }
      6. file.seekg(0, file.beg);
      7. while (!file.eof())
      8. {
         1. file >> word;
         2. if (verbs.inCatalog(word))
            1. wordsOfFile = wordsOfFile + word;
      9. }
      10. file.seekg(0, file.beg);
      11. while (!file.eof())
      12. {
          1. file >> word;
          2. if (others.inCatalog(word))
             1. wordsOfFile = wordsOfFile + word;
      13. }
      14. file.close();
33. };
34. };

Main.cpp

1. #include "SET.h"
2. #include "FILE.h"
3. using namespace std;
4. int main()
5. {
6. setlocale(LC\_ALL, "rus");
7. try
8. {
   * 1. FileOfWord file("nouns.txt", "verbs.txt", "others.txt", "data.txt");
     2. file.readContentOfFile();
     3. file.writeList();
9. }
10. catch (SetException & e) {
    * 1. cout << e.what() << endl;
11. }
12. }

   

