# Московский государственный технический университет им. Н.Э.Баумана

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
| УТВЕРЖДАЮ: |  |
| Правдина А.Д. | "\_\_"\_\_\_\_\_\_\_\_\_\_\_\_\_2019 Г. |

**Комплексная лабораторная работа по дисциплине ПКШ**

**“**Система классов словарей и гнёзд**”**

Руководство пользователя

(вид документа)

писчая бумага

(вид носителя)

9

(количество листов)

|  |  |
| --- | --- |
| ИСПОЛНИТЕЛЬ: |  |
| студент группы ИУ5Ц-41 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Рябкин А.В. | "\_\_"\_\_\_\_\_\_\_\_\_\_\_\_\_2019 г. |

Москва - 2019

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

СОДЕРЖАНИЕ

[1 НАЗНАЧЕНИЕ И УСЛОВИЯ ПРИМЕНЕНИЯ ПРОГРАММЫ 3](#__RefHeading___Toc6186_1642094031)

[2 ПОДКЛЮЧЕНИЕ СИСТЕМЫ КЛАССОВ И ДИАГРАММА КЛАССОВ 3](#__RefHeading___Toc6188_1642094031)

[3 РАБОТА С СИСТЕМОЙ КЛАССОВ 3](#__RefHeading___Toc6190_1642094031)

[4 Общие сведения о системе классов улиц и домов 3](#__RefHeading___Toc6192_1642094031)

[5 Диаграмма классов улиц и домов 4](#__RefHeading___Toc6194_1642094031)

[6 Подключение системы классов в программу 4](#__RefHeading___Toc6196_1642094031)

[7 Общие данные и переменные 4](#__RefHeading___Toc6198_1642094031)

[8 КЛАСС гнезд - Nidus 5](#__RefHeading___Toc6200_1642094031)

[9 Данные и переменные класса Nidus 5](#__RefHeading___Toc6202_1642094031)

[10 Конструкторы класса Nidus 5](#__RefHeading___Toc6204_1642094031)

[11 Деструктор класса Nidus 5](#__RefHeading___Toc6206_1642094031)

[12 Методы класса Nidus 5](#__RefHeading___Toc6208_1642094031)

[13 Операции класса Nidus 6](#__RefHeading___Toc6210_1642094031)

[14 Дружественные функции класса Nidus 6](#__RefHeading___Toc6212_1642094031)

[15 КЛАСС Словарей - Dictionary 6](#__RefHeading___Toc6214_1642094031)

[16 Данные и переменные класса Dictionary 7](#__RefHeading___Toc6216_1642094031)

[17 Конструкторы класса Dictionary 7](#__RefHeading___Toc6218_1642094031)

[18 Деструктор класса Dictionary 7](#__RefHeading___Toc6220_1642094031)

[19 Методы класса Dictionary 7](#__RefHeading___Toc6222_1642094031)

[20 Операции класса Dictionary 8](#__RefHeading___Toc6224_1642094031)

[21 Дружественные функции класса Dictionary 8](#__RefHeading___Toc6226_1642094031)

[22 ОТКЛЮЧЕНИЕ СИСТЕМЫ КЛАССОВ 9](#__RefHeading___Toc6228_1642094031)

[23 СООБЩЕНИЙ ОБ ОШИБКАХ И ИСКЛЮЧИТЕЛЬНЫЕ СИТУАЦИИ 9](#__RefHeading___Toc6230_1642094031)

# 1 НАЗНАЧЕНИЕ И УСЛОВИЯ ПРИМЕНЕНИЯ ПРОГРАММЫ

Система классов словарей и гнезд предназначена для автоматизации работы с объектами словарей и гнезд в программных проектах и предметной области, где необходимо это учитывать. В частности, система классов предназначена для программирования задач включающих действия: создание, удаление каталога словарей и гнезд, подсчета числа словарей и гнезд. Система классов обеспечивает удобную работу с этими объектами, высокий уровень надежности программ, функциональных возможностей, а также сокращение сроков разработки и реализации программных продуктов, где необходимо использовать подобные объекты.

# 2 ПОДКЛЮЧЕНИЕ СИСТЕМЫ КЛАССОВ И ДИАГРАММА КЛАССОВ

Для того, чтобы установить данную систему классов на компьютер следует произвести следующие действия:

1. Освободить на жестком диске как минимум 200 килобайт свободного пространства.
2. Вставить флеш-накопитель или CD/DVD носитель с дистрибутивом программного продукта в USB-порт или CD/DVD устройство.
3. Скопировать файлы. С помощью системной функции копирования файлы (или файл менеджера), которые перечислены в разделе 4 этого документа, с носителя на жесткий диск в нужные каталоги. Для работы программы подойдет любой доступный каталог (DictionaryRealm.exe). Для подключения в новый проект необходимо скопировать (Header.h) в общий каталог заголовочных файлов или в специально созданный каталог для новой сборки программы (Source.cpp, Header.h).
4. Прописать в установках каталогов для используемой на компьютере системы программирования С++ полный пути к директории, куда был установлены модели системы классов (Header .h).

# 3 РАБОТА С СИСТЕМОЙ КЛАССОВ

## 4 Общие сведения о системе классов улиц и домов

Система классов описывает словари и гнезда для разработки программ, в которых учет сведений об этих объектах необходим.

**Словарь** – объекты данного типа содержат информацию о названии, авторе, годе выпуска и количестве значений. Предусматривается возможность изменения параметров словаря.

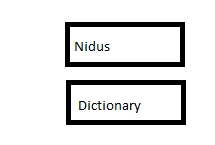
**Гнездо** – объекты данного типа в упорядоченном виде содержат значения ячеек словаря. Предусматривается возможность добавления значений и их удаления.

Система классов позволяет программировать следующие операции и функции:

* Создание и задание свойств объектов типа словарь и гнездо;
* Сложение двух словарей;
* Получение параметров словаря (названия, фамилии автора, года выпуска, количества гнезд);
* Установки и изменения параметров гнезд (значений языковых единиц);
* Удаление объектов типа словарь и гнездо;
* Распечатки свойств объектов типа словарь и гнездо;
* Добавление гнезд в конкретный словарь;
* Удаление гнезда из конкретного словаря;
* Сложение двух словарей (при объединении словарей);
* Переименование словарей и сортировку значений гнезд;
* Распечатки списка гнезд словаря с их характеристиками;
* Получение параметров гнезда (их значений);
* Установки и изменения параметров гнезд (их значений);

## 5 Диаграмма классов улиц и домов

Ниже приставлена диаграмма классов системы классов улиц и домов программного обеспечения (ПО).



Классы системы имеют следующее назначение:

Класс **Nidus** - класс гнезд, позволяющий создавать объекты, описывающие гнезда, в которых учтены их свойства(значения).

Класс **Dictionary** - класс словарей, позволяющий создавать объекты типа словаря с учетом всех их свойств: Фамилии автора, наименования, года издания, числа значений.

## 6 Подключение системы классов в программу

В данном фрагменте подключаются заголовочные файлы библиотек: ввода вывода (**iostream**), стандартных классов MFC (**stdafx.h**) и собственной системы классов улиц и домов (**Header.h**). Кроме того, объявлено использование именованного пространство стандартных функций (**std**).

#include "stdafx.h"

#include "**Header .**h"

#include <iostream>

using namespace std;

## 7 Общие данные и переменные

В программах, использующих систему классов словарей и гнезд глобальные переменные и перечисления не применяются.

# 8 КЛАСС гнезд - Nidus

**Класс Nidus. Гнездо** – объекты данного типа содержат информацию о значениях языковых единиц в словаре. Предусматривается возможность изменения параметров гнезда.

## 9 Данные и переменные класса Nidus

В классе **Nidus** объявлены следующие свойства доступные пользователю:

|  |  |  |  |
| --- | --- | --- | --- |
| **Переменная** | **Тип свойства** | **Защита** | **Назначение** |
| **en** | string | public | Значение на русском языке |
| **ru** | string | public | Значение на английском языке |

## 10 Конструкторы класса Nidus

В классе **Home** описаны следующие конструкторы:

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п/п** | **Прототип** | **Тип возврата** | **Назначение /Параметры** |
|  | Nidus() | **Nidus &** | Нет |
|  | Nidus(string a, string b) | **Nidus &** | Новое гнездо в словаре |

**Примеры** использования конструкторов:

Конструктор без параметров:

Nidus N1;

Конструктор с значениями языковых единиц:

Nidus N2("one","один");

## 11 Деструктор класса Nidus

Прототип деструктора:

~Nidus();

Пример явного и неявного использования деструктора:

Nidus \*p Nidus = new Nidus (N6);

pNidus ->printNiduses();

delete pNidus ;

Неявный вызов деструктора производится при завершении блока операторов . Явный вызов деструктора производится при выполнении операции **delete**.

## 12 Методы класса **Nidus**

В таблице представлены методы класса **Nidus**.

| **№ п/п** | **Прототип** | **Тип возврата** | **Назначение** | **Параметры** |
| --- | --- | --- | --- | --- |
|  | void printNiduses() {  cout << en << " " << ru << endl;} | void | Печать значений гнезда | Нет |
|  | string getRu() const {return ru;} | string | Получить значение на русском языке | Нет |
|  | string getEn() const {return en;} | string | Получить значение на английском языке | Нет |

Примеры использования методов класса **Nidus**:

Вывод значений:

Temp.printNiduses;

Получение значения на английском языке:

cout << "Английское значение = "<< Temp.getEn() << endl;

Получение значения на русском языке:

cout << "Русское значение = "<< Temp.getRu() << endl;

## 13 Операции класса Nidus

Операции класса в данной программе не применяются

## 14 Дружественные функции класса **Nidus**

Прототип перегруженного оператора вывода:

friend ofstream &operator << (ofstream& out, Nidus& obj)

Прототип перегруженного оператора ввода:

friend ifstream &operator >> (ifstream& in, Nidus& obj)

Пример использования перегруженного оператора вывода:

Nidus Temp;

cout <<Temp<<endl;

Пример использования перегруженного оператора ввода:

cin >>Temp;

# 15 КЛАСС Словарей - Dictionary

**Класс Dictionary. Словарь** – объекты данного типа содержат информацию о словарях Предусматривается возможность изменения параметров словаря.

## 16 Данные и переменные класса **Dictionary**

В классе **Dictionary** объявлены следующие свойства доступные пользователю:

|  |  |  |  |
| --- | --- | --- | --- |
| **Переменная** | **Тип свойства** | **Защита** | **Назначение** |
| **autor** | string | public | Фамилия автора словаря |
| **name** | string | public | Наименование словаря |
| **date** | int | public | Год выпуска словаря |
| **amount** | int | public | Количество значений |

## 17 Конструкторы класса **Dictionary**

В классе **Dictionary** описаны следующие конструкторы:

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п/п** | **Прототип** | **Тип возврата** | **Назначение /Параметры** |
| 1 | Dictionary() | **Dictionary &** | Нет |
| 2 | Dictionary(string autorn, string namen, int daten, int amountn) | **Dictionary &** | Создание нового объекта словаря без гнезд |
| 3 | Dictionary(string autorn, string namen, int daten, int amountn, vector <Nidus> defdicts) | **Dictionary &** | Создание нового объекта словаря |

**Примеры** использования конструкторов:

Конструктор без параметров:

Dictionary D1;

Конструктор с созданием объекта словаря без гнезд:

Dictionary D2("Рябкин","Этимологический", 1997, 1025);

Конструктор с созданием объекта

Dictionary D3("Рябкин","Этимологический", 1997, 1025, {("one","один")});

## 18 Деструктор класса **Dictionary**

Прототип деструктора:

~Dictionary();

Пример явного и неявного использования деструктора:

Dictionary \*pDictionary = new Dictionary (D6);

pDictionary ->printDictionaries();

delete pDictionary ;

Неявный вызов деструктора производится при завершении блока операторов . Явный вызов деструктора производится при выполнении операции **delete**.

## 19 Методы класса **Dictionary**

В таблице представлены методы класса **Dictionary**.

| № п/п | Прототип | Тип возврата | Назначение | Параметры |
| --- | --- | --- | --- | --- |
| 1 | void print() | void | Печать словаря | Нет |
| 2 | void printS() | void | Печать словаря без его гнезд | Нет |
| 3 | void printPars() | void | Печать только гнезд словаря | Нет |
|  | void AddPars() | void | Добавление гнезд в словарь | Нет |
|  | void DeletePars() | void | Удаление гнезд из словаря | Нет |
|  | int SearchWord() | int | Поиск по значениям гнезд словаря, возвращает номер гнезда | Нет |
|  | void SortNiduses() | void | Сортировка словаря по значениям гнезд | Нет |
|  | vector <Nidus> NidusesToSend() | vector <Nidus> | Вспомогательный метод для удаления словарей | Нет |
|  | int GetDictionaryInfo() | int | Возвращает 1 если в словаре есть гнезда | Нет |

Примеры использования методов класса **Dictionary**:

Вывод словаря:

D1.print;

Вывод характеристик словаря без его гнезд:

D1.printS;

Вывод только гнезд словаря

D1.printPars;

Удаление пар в словаре

D1.DeletePars;

## 20 Операции класса **Dictionary**

Операции класса в данной программе не применяются

## 21 Дружественные функции класса **Dictionary**

Прототип перегруженного оператора вывода:

friend ofstream& operator <<(ofstream& out, Dictionary& obj)

Прототип перегруженного оператора ввода:

friend ifstream& operator >>(ifstream& in, Dictionary& obj)

Пример использования перегруженного оператора вывода:

Dictionary Temp;

cout <<Temp<<endl;

Пример использования перегруженного оператора ввода:

cin >>Temp;

# 22 ОТКЛЮЧЕНИЕ СИСТЕМЫ КЛАССОВ

Чтобы отключить данную систему классов, надо с помощью системной функции удаления (или файл менеджера) удалить файлы, перечисленные в пункте 4 данного документа, из каталогов , куда они были скопированы.

# 23 СООБЩЕНИЙ ОБ ОШИБКАХ И ИСКЛЮЧИТЕЛЬНЫЕ СИТУАЦИИ

В системе классов гнезд и словаре не предусмотрена обработка исключительных ситуаций и выдача диагностических сообщений. В программах, разработанных с включением этой системы классов, может быть предусмотрена обработка исключительных ситуаций и выдача сообщений об ошибках. Выдаваемые системные диагностические сообщения и возникающие системные исключительные ситуации описаны в документации на систему программирования (MS VS 2017).