Лабораторная работа № 1

**Динамические структуры данных, работа с файлами**

**Цель работы:** Закрепить практические навыки, полученные при изучении дисциплины «Основы программирования». Решение задач с использованием динамических структур, работа с файлами.

**Теоретическая подготовка**

Лабораторная работа № 1 представляет собой повторение практических навыков, изучаемых в предыдущем семестре, поэтому новых теоретических материалов не приводится. При подготовке к лабораторной работе смотрите материалы, приведенные в папке «Самостоятельная работа», а также разработки к лабораторным работам и БДЗ прошлого семестра.

### Контрольные вопросы

В качестве контрольного вопроса может быть задан любой вопрос по темам

(смотрите вопросы к лабораторным работам прошлого семестра):

* типы и модификаторы типов данных
* операции и операторы
* массивы и строки
* указатели и ссылки
* пользовательские функции
* структуры
* динамическая работа с памятью
* ввод-вывод на консоль
* ввод-вывод в файл

**Общие требования к выполнению заданий**

1. Во всех вариантах нужно создать базу данных, согласно варианту задания
2. Данные размещаются в динамической памяти.
3. Обязательные функции для всех вариантов:

* добавить новый элемент в базу
* распечатка данных в табличном виде
* выход из программы

1. Остальные функции для работы с базой указаны в задании индивидуально.
2. Для выполнения функций, указанных в задании, написать диалоговый интерфейс, позволяющий выполнять функции в произвольном порядке многократно
3. Первичное создание базы – ввод данных с клавиатуры
4. Если программа уже запускалась, то данные загружаются из файла перед выходом на диалоговый интерфейс. Иными словами вносятся изменения и дополнения в уже существующую базу данных.
5. При выполнении функции «выход из программы» нужно сохранить базу на диске

**Пример диалогового интерфейса**

**(База данных «Склад товаров»)**

Добавить новый элемент …… 1

Распечатать базу товаров …… 2

Поиск товара по названию …… 3

Фильтр по цене ………….. 4

Выход из программы ………….. 5

……………………………………..

Введите номер функции

**Пример распечатки данных в табличном виде**

(**База данных «Склад товаров»)**

**Название Цена Количество Общая сумма**

**Товара (руб) (кг) (руб)**

**- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -**

Сыр «Российский» 560 26.5 14840.00

Масло сливочное 380.5 100.25 38145.12

Рис длинный 68 25.0 1700.00

Рис круглый 62 56.75 3518.50

**- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -**

Всего товаров на сумму 58203.62

Количество записей в базе 4

## Варианты заданий

|  |  |
| --- | --- |
| **Номер компьютера** | **Задание** |
| 1, 11, 21 | База данных: Пищевое производство (расчеты).  Создать базу данных со следующей структурой :  char\* name ; //Название изделия  int quant; // количество  double cost; //стоимость  Смотрите «Общие требования к выполнению заданий»  Создать обязательные функции, указанные в общих требованиях.  Создать функции для данного варианта:   * Поиск изделий по названию * Найти все изделия больше заданной стоимости |
| 2, 12, 22 | База данных: Поезд (заказ билетов).  Создать базу данных со следующей структурой :  char\* fam; //Фамилия пассажира  char\* name; // Имя пассажира  char\* arrival; // пункт прибытия  double price; // цена билета  Смотрите «Общие требования к выполнению заданий»  Создать обязательные функции, указанные в общих требованиях. Создать функции для данного варианта:   * Поиск билета по имени и фамилии пассажира * Найти все билеты с ценой более заданного значения |

|  |  |  |
| --- | --- | --- |
| 3, 13, 23 | | База данных: Почта (ценное письмо).  Создать базу данных со следующей структурой :  char\* adr1; //адрес получателя  char\* fam1; //фамилия получателя  char\* name1; // имя получателя  char\* adr2; // адрес отправителя  char\* fam2; //фамилия отправителя  char\* name2; // имя отправителя  double cost; // стоимость письма  Смотрите «Общие требования к выполнению заданий»  Создать обязательные функции, указанные в общих требованиях. Создать функции для данного варианта:   * Поиск письма по полям fam2, name2 (фамилия и имя отправителя) * Найти все письма со стоимостью более заданного значения |
| 4, 14, 24 | База данных: Студент.  Создать базу данных со следующей структурой :  char\* fam; //фамилия студента  char\* name; // имя студента  int grup // номер группы  Смотрите «Общие требования к выполнению заданий»  Создать обязательные функции, указанные в общих требованиях. Создать функции для данного варианта:   * Поиск студента по фамилии и имени * Найти всех студентов заданной группы | |
| 5, 15, 25 | База данных: Производство (поставщики сырья)  Создать базу данных со следующей структурой :  char\* firma; // название фирмы поставщика  double raw\_ material // сумма поставки сырья  double payment; // оплата сырья  Смотрите «Общие требования к выполнению заданий»  Создать обязательные функции, указанные в общих требованиях.  Создать функции для данного варианта:   * Поиск поставщика по названию фирмы * Найти поставщиков, которые получили не полную оплату за сырье | |

|  |  |
| --- | --- |
| 6, 16, 26 | База данных: Поликлиника.  Создать базу данных со следующей структурой :  char\* fam; //фамилия врача  char\* name; // имя врача  int specialty; // специальность  (1-терапевт, 2-кардиолог, 3- стоматолог)  Смотрите «Общие требования к выполнению заданий»  Создать обязательные функции, указанные в общих требованиях. Создать функции для данного варианта:   * Поиск врача по фамилии и имени * Найти всех врачей заданной специальности |
| 7, 17, 27 | База данных: Склад (товары).  Создать базу данных со следующей структурой :  char\* name; //Название товара  double price; // цена  int quant; // количество  Смотрите «Общие требования к выполнению заданий»  Создать обязательные функции, указанные в общих требованиях.  Создать функции для данного варианта:   * Поиск товара по названию * найти все товары с количеством меньше заданного |
| 8, 18, 28 | База данных: Банк (депозит).  Создать базу данных со следующей структурой :  char\* fam; // фамилия клиента  char\* name; // имя клиента  double sum; // Сумма на счете (сумма депозита )  Смотрите «Общие требования к выполнению заданий»  Создать обязательные функции, указанные в общих требованиях.  Создать функции для данного варианта:   * Поиск клиента по фамилии и имени * Найти клиентов с суммой на счете больше заданной |

|  |  |
| --- | --- |
| 9, 19, 29 | База данных: Работник (зарплата).  Создать базу данных со следующей структурой :  char\* fam; // фамилия работника  char\* name; // имя работника  int department; // отдел  double salary; // оклад  Смотрите «Общие требования к выполнению заданий»  Создать обязательные функции, указанные в общих требованиях.  Создать функции для данного варианта:   * Поиск работника по фамилии и имени * Найти всех работников заданного отдела |
| 10, 20, 30 | База данных: Преподаватель.  Создать базу данных со следующей структурой :  char\* fam; //фамилия преподавателя  char\* name; // имя преподавателя  int status; //ученое звание  ( 1- преподаватель, 2- доцент, 3-доктор наук)  Смотрите «Общие требования к выполнению заданий»  Создать обязательные функции, указанные в общих требованиях.  Создать функции для данного варианта:   * Поиск преподавателя по имени и фамилии * Найти всех преподавателей заданного ученого звания (status) |