Министерство образования Республики Беларусь

Учреждение образования

БелорусскиЙ государственный университет

информатики и радиоэлектроники

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

Дисциплина: Основы алгоритмизации и программирования(ОАиП)

# **Пояснительная записка**

к учебной практике(ознакомительной)

на тему

**Каталог компьютеров**

БГУИР КР 1-40 01 01 106 ПЗ

|  |  |  |
| --- | --- | --- |
| Студент |  | К. А. Зиновенко |
| Руководитель |  | асс. Е. Е. Фадеева |

Минск, 2024

**СОДЕРЖАНИЕ**

1. Разработка технического задания…………………………………………3
2. Описание блок-схем алгоритмов………………………………………….4
3. Выделение основных структур данных…..…………..…………………..9
4. Описание структур в виде таблиц……………………………………….11
5. Определение подпрограмм и их описание……………………………...12
6. Тестирование и отладка программы…………………………………….18
7. Руководство пользователя………………………………………………...
8. Текст программы………………………………………………………….
9. Приложение А...………………………….……………………………….
10. Приложение Б…...………………………………………………………..

1. Разработка технического задания.

Для программного средства “Каталог компьютеров” было разработано техническое задание, представленное в приложении А(см. стр. 11).

2. Описание блок-схем алгоритмов.

Схемы алгоритмов представлены в приложении Б(см. стр. 12). Ниже представлены описания алгоритмов.

2.1. Основная схема программы(рис. 1)

Данная схема декомпозирует основной блок программы, который включает себя приветствие пользователя, выделение памяти под списки, и вход в цикл, который представляет собой набор функций, которые выполняются при вводе соответствующего кода функции. Выход из цикла происходит, когда вводится код функции, которая предполагает выход из программы. Как только цикл был покинут, программа очищает память, ранее выделенную под списки, и прощается с пользователем.

2.2. Чтение данных из файла(рис. 2)

В данный алгоритм передаются списки, а также параметр, который указывает, были ли данные уже прочтены из файла. Так же алгоритм проверяет, есть ли в списке на момент выполнения функции. Если данные не были прочтены или пользователь желает перезаписать имеющиеся данные, то далее вводится директория к папке, которая содержит типизированные файлы с данными, затем вводится имя папки. При отсутствии в папке необходимых файлов выводится соответствующее сообщение, и управление передаётся обратно, иначе - данные прочтены и алгоритм завершает работу.

2.3. Выбор списка(рис. 3)

Функция возвращает номер выбранного списка.

2.4. Просмотр списков(рис. 4)

Пользователь вводит номер списка, который далее выводиться на экран.

2.5. Сортировка списков(рис. 5)

Пользователь вводит номер списка, в процедуру передаётся список, далее вводится номер поля, по которому идёт сортировка. Если список пуст или содержит 1 элемент, то сортировка не производится. Алгоритм сортировки заключается в поиске минимального элемента списка в промежутке от границы до конца списка и вставке его после границы, затем граница двигается на 1 элемент вправо. Алгоритм заканчивается, когда граница достигает конца списка.

Для списка комплектующих доступны следующие поля:

1. Код комплектующего
2. Код типа комплектующего
3. Производитель
4. Имя модели
5. Цена
6. Количество

Для списка типов комплектующих доступны следующие поля:

1. Код типа комплектующего
2. Название

Для списка совместимых комплектующих доступны следующие поля:

1. Код первого комплектующего
2. Код второго комплектующего

2.6. Поиск в списке(рис. 6)

Пользователь вводит номер списка, в процедуру передаётся список, вводится номер поля, по которому идёт поиск. Алгоритм проходиться по всему списку и, найдя элементы, выводит их, иначе - сообщение о то, что записи не найдены.

Для списка комплектующих доступны следующие поля:

1. Код комплектующего
2. Код типа комплектующего
3. Производитель
4. Имя модели

Для списка типов комплектующих доступны следующие поля:

1. Код типа комплектующего
2. Название

2.7. Добавление в список(рис. 7)

Пользователь вводит номер списка, далее:

* Для списка комплектующих передаются списки комп. и типов комп., выводится список типов комплектующих, чтобы пользователь мог выбрать нужный тип, выделяется память под новую запись, затем вводится тип комплектующего(если введён несуществующий, то алгоритм попросит ввести снова), производитель, имя модели(если будет введена уже имеющаяся модель, то алгоритм попросит ввести снова), параметры, цена и количество(если цена или количество будет меньше нуля, то алгоритм попросит ввести снова).
* Для списка типов комплектующих передаётся список типов комп., выделяется память под новую запись, вводится название(если такой тип уже есть, то алгоритм попросит ввести снова).
* Для списка совместимых комплектующих передаются списки комп. и совм. комп., выводится список комплектующих, чтобы пользователь выбрал нужные, выделяется память под новую запись, далее вводится код первого и второго комплектующего(если введённая запись уже существует, либо 1 или 2 коды введены некорректно, то алгоритм попросит ввести снова).

2.8. Удаление из списков(рис. 8)

Пользователь вводит номер списка, далее:

* Для списка комплектующих передаются списки комп. и совм. комп., выводится список комплектующих, чтобы пользователь выбрал нужные записи для удаления, далее вводится код комплектующего(если введён несуществующий код, то алгоритм попросит ввести снова), запись удаляется, память очищается. При этом происходит поиск удалённого комплектующего в списке совм. комп., и ,если таковые найдены, то в данном списке удаляются записи и очищается память.
* Для списка типов комплектующих передаются списки комп., типов комп. и совм. комп., выводится список типов комплектующих, чтобы пользователь выбрал нужные записи для удаления, далее вводиться код типа комплектующего(если введён несуществующий код, то алгоритм попросит ввести снова), запись удаляется, память очищается. При этом идёт поиск всех комплектующих в списке комплектующих, которые имеют удалённый тип, и, если такие найдены, записи удаляются, память очищается, и по алгоритму для списка комплектующих идёт поиск и удаление(если найдены) записей в списке совместимых комплектующих.
* Для списка совместимых комплектующих передаётся данный список, далее список выводится на экран, чтобы пользователь мог выбрать нужные записи для удаления, далее вводятся коды записи(если введена несуществующая запись, то алгоритм попросит ввести снова), запись удаляется, память очищается.

2.9. Редактирование в списках(рис. 9)

Пользователь вводит номер списка, далее в процедуру передаются списки(редактируемый и связные с ним), на экран выводится список для выбора записи для редактирования, далее вводится код записи(если введён несуществующий, то алгоритм попросит ввести снова). Далее пользователь редактирует поля(чтобы оставить поле неизменным, нужно нажать enter), проверка на ввод осуществляется так же, как и при добавлении элемента.

2.10. Специальные функции

2.10.1. Меню спецфункций(рис 10.1)

Функция возвращает код выбранной спецфункции.

2.10.2. Подбор всех возможных комбинаций компьютера в заданном ценовом диапазоне(рис 10.2)

2.10.2.1. Создание массива совместимых комплектующих(рис 10.2.1)

В функцию передаётся список совместимых комплектующих, далее внутри идёт проверка на совместимость всех имеющихся комплектующих друг с другом. Функция возвращает массив комплектующих, которые совместимы друг с другом и дают возможность собрать компьютер.

2.10.2.2 Получение всех комбинаций индексов(рис. 10.2.2)

В функцию передаётся длина массива совместимых комплектующих. Алгоритм находит все возможные сочетания без повторений. Функция возвращает массивы, которые представляют собой комбинации индексов.

2.10.2.3 Сортировка комбинаций(рис. 10.2.3)

В функцию передаётся массив с комбинациями и список комплектующих. Алгоритм сортирует все подобранные комбинации, а так же проверяет возможность приобретения той или иной комбинации(зависит, есть ли в наличии данные комплектующие).

В начале алгоритма проверяется, не были ли до этого подобраны комбинации, и, если были, то пользователю предлагается перезаписать, или оставить как есть. Далее пользователь вводит ценовой диапазон(если введено меньше нуля, то алгоритм попросит ввести снова). Если комбинации не были подобраны, выводится соответствующее сообщение, иначе - алгоритм возвращает массив всех возможных комбинаций сборки.

2.10.3. Оформление заказа(рис. 10.3)

В процедуру передаётся массив комбинаций. Далее на экран выводятся все возможные варианты для заказа, где пользователь решает, хочет ли он что-либо заказать или нет. Если да, то вводится директория, и создаётся текстовый файл с заказом.

2.10.4. Просмотр подобранных вариантов(рис. 10.4)

В процедуру передаётся массив комбинаций. Далее они выводятся на экран. Если пользователь желает сохранить в файл, то в начале вводит директорию, и данные сохраняются в файл.

2.11. Выход из программы

2.11.1 Выход без сохранения(рис 11.1)

Функция удаляет записи из списков и очищает память, результатом является код выхода из программы.

2.11.2 Выход с сохранением(рис 11.2)

Пользователь вводит директорию и имя папки, далее данные сохраняются в файл, удаляются записи из списков и очищается память. Результатом является код выхода из программы.

.

3. Выделение основных структур данных

|  |  |  |
| --- | --- | --- |
| Имя идентификатора структуры | Назначение структуры | Тип структуры |
| TCompPartsArr | Массив совместимых комплектующих | array of integer |
| TcompPartsMtx | Массив комбинаций сборки | array of TCompPartsArr |
| TcompPartsArrPrice | Массив стоимости комбинаций с соот. индексом | array of real |
| TCompPartsArrAvaliable | Массив с данными о возможности покупки комбинации с соот. индексом | array of boolean |
| TCombsFile | Текстовый файл для оформления заказа или записи комбинаций в файл | TextFile |
| PartListDataType | Структура списка комплектующих | packed record  partCode: integer;  partTypeCode: integer;  manufacturer: TString;  modelName: TString;  parameters: TString;  price: real;  availability: integer;  end; |
| PartListType | Список комплектующих | ^PartListPointer |
| PartListPointer | Указатель на запись списка комплектующих | packed record  lastID: integer;  partListInfo: PartListDataType;  partListNextElement: PartListType;  end; |
| PartListFileType | Файл со структурой списка комплектующих | file of PartListDataType |
| PartTypeListDataType | Структура списка типов комплектующих | packed record  partTypeCode: integer;  partTypeName: TString;  end; |
| PartTypeListType | Список типов комплектующих | ^PartTypeListPointer |
| PartTypeListPointer | Указатель на запись списка типов комплектующих | packed record  lastID: integer;  partTypeListInfo: PartTypeListDataType;  partTypeListNextElement: PartTypeListType;  end; |
| PartTypeListFileType | Файл со структурой списка типов комплектующих | file of PartTypeListDataType |
| CompatiblePartListDataType | Структура списка совместимых комплектующих | packed record  firstPartCode: integer;  secondPartCode: integer;  end; |
| CompatiblePartListType | Список совместимых комплектующих | ^CompatiblePartListPointer |
| CompatiblePartListPointer | Указатель на запись списка совместимых комплектующих | packed record  compatiblePartListInfo: CompatiblePartListDataType;  compatiblePartListNextElement: CompatiblePartListType;  end; |
| CompatiblePartListFileType | Файл со структурой списка типов комплектующих | file of CompatiblePartListDataType |

4. Описание структур в виде таблиц

4.1. Список комплектующих

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Код комп. | Код типа комп. | Изгото-витель | Имя  модели | Пара-метры | Цена | Кол-во |
| 1 | 1 | Intel | Core i9 14900H | Frec 4.9 Ghz | 799 | 1 |
| 2 | 2 | Asus | B550HM | Socket lga 1700 | 599 | 2 |
| 3 | 3 | HyperX | Fury | Frec 5200 Mhz | 300 | 3 |
| 4 | 4 | Nvidia | RTX 4090 | 8 Gb VRAM | 1200 | 4 |

4.2. Список типов комрлектующих

|  |  |
| --- | --- |
| Код типа комплектующего | Название |
| 1 | Central Processing Unit |
| 2 | MotherBoard |
| 3 | Random access memory |
| 4 | Graphical Processing Unit |

4.3. Список совместимых комплектующих

|  |  |
| --- | --- |
| Код первого комплектующего | Код второго комплектующего |
| 1 | 2 |
| 1 | 3 |
| 1 | 4 |
| 2 | 3 |

4.4. Массив совместимых комплектующих

[1, 2, 3, 4, 5, 6], [1, 2, 3], [1, 3, 6, 8]

4.5. Массив комбинаций

[[1, 2], [1, 3], [1, 2, 3, 4], [1, 2, 3, 4, 5, 6]

4.6. Массив сумм комбинаций

[3, 4, 10, 21]

4.7. Массив с информацией о наличии

[true, true, false, true]

5. Определение подпрограмм и их описание

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя подпрограммы | Назначение подпрограммы | Заголовок подпрограммы | Имя параметра | | Назначение параметра |
| ClearScreen | Очистка консоли | procedure ClearScreen(); | - | | - |
| ReadFromFiles | Чтение данных из типизированных файлов | procedure ReadFromFiles(list1: PartListType; list2: PartTypeListType;  list3: CompatiblePartListType; var isReadFromFile: boolean); | List1 | | Список комплектующих |
| List2 | | Список типов комплектующих |
| List3 | | Список совместимых  комплектующих |
| isReadFromFile | | Пометка, были ли данные уже прочтены |
| GetList | Возвращает номер выбранного списка | function ShowListMenu(): integer;  function SortListMenu(): integer;  function FindInListMenu(): integer;  function AddToListMenu(): integer;  function DeleteFromListMenu(): integer;  function EditInListMenu(): integer; | - | | - |
| ShowList | Вывод списка на экран | procedure ShowPartList(list: PartListType);  procedure ShowPartTypeList(list: PartTypeListType);  procedure ShowCompatiblePartList(list: CompatiblePartListType); | List | | Список, который выводится |
| SortList | Сортировка списка | procedure SortPartList(list: PartListType);  procedure SortPartTypeList(list: PartTypeListType);  procedure SortCompatiblePartList(list: CompatiblePartListType); | List | | Список, который сортируется |
| FindInList | Поиск в списке | procedure FindInPartList(list: PartListType);  procedure FindInPartTypeList(list: PartTypeListType);  procedure FindInCompatiblePartList(list: CompatiblePartListType) | List | | Список, в котором происходит поиск |
| AddToPartList | Добавление в список комплектующих | procedure AddToPartList(list: PartListType; checkList: PartTypeListType); | List | | Список комплектующих |
| CheckList | | Список типов комплектующих |
| AddToPartTypeList | Добавление в список типов комплектующих | procedure AddToPartTypeList(list: PartTypeListType); | List | | Список типов комплектующих |
| AddToCompatiblePartList | Добавление в список совмести-мых комплектующих | procedure AddToCompatiblePartList(list: CompatiblePartListType;  checkList: PartListType); | List | | Список совмести-мых  комплектующих |
| CheckList | | Список комплектующих |
| DeleteFromPartList | Удаление из списка комплектующих | procedure DeleteFromPartList(list: PartListType;  deleteList1: CompatiblePartListType); | List | | Список комплектующих |
| DeleteList1 | | Список типов комплектующих |
| DeleteFromPartTypeList | Удаление из списка типов комплектующих | procedure DeleteFromPartTypeList(deleteList1: PartListType;  list: PartTypeListType; deleteList2: CompatiblePartListType); | List | | Список типов комплектующих |
| DeleteList1 | | Список комплектующих |
| DeleteList2 | | Список совмести-мых  комплектующих |
| DeleteFromCompatiblePartList | Удаление из списка совмести-мых комплектующих | procedure DeleteFromCompatiblePartList(list: CompatiblePartListType); | List | | Список совмести-мых  комплектующих |
| EditInList | Редактирование в списках | procedure EditInPartList(list: PartListType; checkList: PartTypeListType);  procedure EditInPartTypeList(list: PartTypeListType);  procedure EditInCompatiblePartList(list: CompatiblePartListType; checkList: PartListType); | List | | Список для редактирования |
| SpecFunsMenu | Меню для выбора и исполнения спецфункций | function SpecialFunctionsMenu(isSpecFunCompleted: boolean): integer; | isSpecFunCompleted | | Была ли спецфункция уже выполнена |
| GetCompPrtsArr | Функция возвращает массив совместимых комплектующих | function GetCompatiblePartsArray(list: CompatiblePartListType): TCompPartsArr; | List | | Список совмести-мых  комплектующих |
| GetIndexArrCombs | Создание массива всевозможных комбинаций сборки | procedure GetAllCombsIndex(var IndexArr: TcompPartsMtx; n, m: integer); | IndexArr | | Массив комбинаций индексов |
| N | | Значение N в сочетаниях |
| m | | Значение m в сочетаниях |
| SortCombs | Сортирует все комбинации и обозначает те, которые можно заказать | function SortCombs(list: PartListType; var IndexMtx: TcompPartsMtx;  var avaliable: TCompPartsArrAvaliable): TcompPartsArrPrice; | List | | Список комплектующих |
| avaliable | | Массив с информацией о возможности заказа комбинации |
| IndexMtx | | Массив комбинаций |
| ShowCombs | Вывод на экран комбинаций и сохранение данных в текстовый файл | function ShowAllCombs(list: PartListType; mtx: TcompPartsMtx;  sum: TcompPartsArrPrice; price: real): integer; | List | | Список комплектующих |
| Mtx | | Массив комбинаций |
| Sum | | Массив стоимостей комбинаций |
| Price | | Ценовой диапазон |
| MakeOrder | Запись заказа в текстовый файл | procedure MakeOrder(list: PartListType; mtx: TcompPartsMtx;  sum: TcompPartsArrPrice; avaliable: TCompPartsArrAvaliable; size: integer;  price: real); | List | | Список комплектующих |
| Mtx | | Массив комбинаций |
| Sum | | Массив стоимостей комбинаций |
| avaliable | | Массив с инфор. о возможности покупки |
| size | | Кол-во комбинаций |
| Price | | Ценовой диапазон |
| ExitFuns | Выход из программы(без сохранения, либо с сохранением в файл) | function SaveWithoutChanges(list1: PartListType; list2: PartTypeListType;  list3: CompatiblePartListType): boolean;  function SaveWithChanges(list1: PartListType; list2: PartTypeListType;  list3: CompatiblePartListType): boolean; | List1 | Список комплектующих | |
| List2 | Список типов комплектующих | |
| List3 | Список совместимых  комплектующих | |
| MainMenu | Главное меню программы | function MainMenu(): integer; | - | - | |

6. Тестирование и отладка программы

|  |  |  |  |
| --- | --- | --- | --- |
| № теста | Условие теста | Ожидаемый результат | Результат теста |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |