

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ ИМ. Р.Е. АЛЕКСЕЕВА

ИНСТИТУТ РАДИОЭЛЕКТРОНИКИ И ИНФОРМАЦИОННЫХ

ТЕХНОЛОГИЙ

Кафедра «Информационные радиосистемы»

КУРСОВАЯ РАБОТА

«Рассматривается информация о какой-либо коллекции (марок, монет, и т. д.).

Сведения об отдельном предмете коллекции включают: название предмета, стоимость, год создания. Должна быть реализована операция отбора предметов коллекции, год создания которых не превосходит заданное значение и стоимость которых не меньше чем указанное значение. »

Выполнил:

студент группы **24-Рз И. И. Иванов**

Проверил:

доцент кафедры ИРС С. Б. Сидоров

Нижний Новгород

2025

Содержание

Введение.....	3
1. Постановка задачи.....	4
2. Руководство пользователя.....	5
3 Руководство программиста.....	7
3.1 Структура программы.....	7
3.2 Структура данных.....	7
3.3 Алгоритм отбора монет.....	8
Заключение.....	9
Список литературы.....	10

Введение

В работе рассматривается решение задачи разработки прикладной программы отбора предметов коллекции, год создания которых не превосходит заданное значение и стоимость которых не меньше чем указанное значение.

Одним из способов преодоления сложности данной задачи является использование отдельной функции для выполнения обработки. Этот подход позволяет свести задачу к последовательному решению более простых задач. Его использование позволяет уменьшить затраты на отладку и повысить надёжность программы. Также существенным моментом является использование структурных типов данных для адекватного отображения сущностей проблемной области в область программной реализации.

В начале основной части отчёта приводится точная формализованная постановка задачи с указанием полного набора операций, выполнение которых должна обеспечивать прикладная программа.

В руководстве пользователя раскрывается назначение программы, её возможности и выполняемые операции. Подробно объясняются правила пользования программой и приводятся конкретные примеры диалога с пользователем.

В руководстве программиста рассматриваются вопросы внутренней организации программы, в том числе перечень функций и их взаимодействие. Кроме этого описываются используемые структуры данных и наиболее важные, и интересные с точки зрения их реализации алгоритмы.

В заключении делаются выводы о полноте решения поставленной задачи, а также приводится ряд соображений по возможным направлениям доработки полученной прикладной программы. Также приведена техническая информация, включающая листинги программы.

1. Постановка задачи

Рассматривается модель информации о коллекции монет. Описание монеты представлено совокупностью свойств: название, стоимость, год создания. В области программной реализации модель монеты имеет вид структурного типа данных.

Рассматривается набор монет с конечным количеством элементов. Требуется получить программную реализацию заданной обработки такого набора структурированных данных.

Все данные, необходимые для обработки, запрашиваются у пользователя: значения элементов, входящих в набор, и дополнительная информация, необходимая для выполнения конкретной обработки данных.

Обработка должна быть реализована отдельной функцией. При этом вся необходимая для выполнения обработки информация должна передаваться в функцию через список аргументов. Результат обработки набора структурированных данных должен передаваться из функции также через список аргументов.

Полученные результаты обработки должны быть выданы на монитор, то есть на стандартное устройство вывода.

Задан определенный набор монет. Требуется найти все монеты, год создания которых, не превосходит заданное значение и стоимость которых не меньше чем указанное значение, и вывести их параметры на стандартное устройство.

2. Руководство пользователя

Программа предназначена для определения точек, выходящих за предел окружности, в режиме диалога с пользователем.

Программа позволяет пользователю задать параметры монет, и их количество. Результаты обработки выводятся на экран монитора. Через консоль (терминал).

Запуск программы осуществляется либо через командную строку с вводом имени исполняемого файла, полученного в результате компиляции, с последующим нажатием клавиши *Enter*, либо иным способом в зависимости от операционной системы.

Программа является интерактивным консольным приложением. Весь диалог с пользователем осуществляется в текстовом режиме.

Сначала пользователю предлагается указать количество монет в коллекции:

- Enter number of coins:

После этого у пользователя запрашиваются параметры каждой монеты из коллекции, а именно название, цена, год выпуска:

- Enter name for the coin #:
- Enter price for the coin #:
- Enter year for the coin #:

После ввода параметров каждой монеты, у пользователя запрашиваются критерии поиска, а именно год создания, который не должны превосходить искомые монеты, и стоимость, которая должна быть не меньше чем указанное значение.

- Enter a year (coins from this year or earlier will be considered):
- Enter a price (coins with this price or higher will be considered):

После ввода набора монет, и критериев поиска, будет осуществлен поиск по коллекции. По завершении обработки выводится результат следующего вида:

- Applicable coins (# Name Price Year):
 - (*#. Name Price Year*)
 - (*#. Name Price Year*)

Если подходящих монет не найдено, выводится соответствующее сообщение:

- No coins are applicable.

По завершении выдачи результатов программа завершает свою работу.

3 Руководство программиста

3.1 Структура программы

Прикладная программа разработана с использованием принципов императивного программирования. Она является совокупностью взаимодействующих функций. Структура программы представлена на рис. 1.



Рисунок 1. Структура программы

Программа состоит из двух функций, назначение каждой из которых приведено ниже:

1. `main` – основная функция приложения
2. `select_coins` – функция поиска монет в коллекции
3. `print_coins` – функция вывода монет в коллекции

3.2 Структура данных

Для описания монеты определяем структурный тип данных `Coin`. По условию задачи, для описания монеты, используется имя, цена и год выпуска, так же, включим текущий порядковый номер монеты. Таким образом получаем следующее определение структурного типа данных, для описания понятия “Монета”:

```
typedef struct {  
    int index;  
  
    char name[NAME_LEN];
```

```

        int price;

        int year;

    } Coin;

```

Для представления коллекции монет используем массив из элементов `Coin coins_collection[coin_count]`. Максимально допустимое количество элементов в массиве ограничим целочисленной переменной `coin_count`, ее значение запрашивается у пользователя.

Для представления набора выбранных монет используем массив из элементов `Coin coins_selected[coin_count]`, так как количество выбранных монет не может превышать общее количество монет, то количество элементов так же, ограничим переменной `coin_count`.

3.3 Алгоритм отбора монет

Отбор монет из коллекции сводится к проверке логического выражения для каждой монеты из коллекции:

$$(год_монеты[i] \leq (год_введенный_пользователем) \ \&\& \ (цена_монеты[i] \geq (цена_введенная_пользователем)))$$

Если текущая монета удовлетворяет условию, значение логического выражения – ИСТИНА, в таком случае данные об этой монете добавляются в вспомогательный массив `coins_selected[]`.

Заключение

В данной работе задача разработки прикладной программы отбора монет из коллекции, была решена с использованием императивного программирования. На основании проведённой отладки и испытаний с помощью контрольных примеров можно сделать вывод, что полученная прикладная программа решает поставленную задачу правильно и в полном объёме.

Список литературы

1. Керниган, Брайан У., Ритчи, Деннис М. Язык программирования С, 2-е издание. [пер. с англ.] / Б.У. Керниган, Д.М. Ритчи – М.: Вильямс, 2007.
2. Павловская, Т.А. С/С++. Программирование на языке высокого уровня: учебник для ВУЗов / Т.А. Павловская. – СПб.: Питер, 2009.
3. Орлов, С.А. Технологии разработки программного обеспечения. учеб. пособие. 2-е изд./ С.А. Орлов, – СПб.: Питер, 2003. – 480 с.: ил.
4. Борисенко, В.В. Основы программирования / В.В.Борисенко, – Интернет-университет информационных технологий – ИНТУИТ.ру, 328 стр. – 2005 г.
5. Шилдт, Г. Полный справочник по С: учеб. пособие / Г. Шилдт. – 4-е изд. – М.: Изд. дом "Вильямс", 2008.
6. Костюкова, Н.И. Язык Си и особенности работы с ним / Н.И. Костюкова, Н.А. Калинина – Интернет-университет информационных технологий – ИНТУИТ.ру, 208 стр. – 2006 г.

Приложение А

Заголовочный файл

```
#ifndef COIN_H
#define COIN_H

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define NAME_LEN 50

typedef struct {
    int index;
    char name[NAME_LEN];
    int price;
    int year;
} Coin;

int select_coins(Coin coins_collection[], Coin coins_selected[], int coin_count,
int year, int price);
void print_coins(Coin coins_selected[], int selected_num);

#endif // COIN_H
```

Основная программа

```
#include "coin.h"

int select_coins(Coin coins_collection[], Coin coins_selected[], int coin_count,
int year, int price) {
    int found = 0;
    for (int i = 0; i < coin_count; ++i) {
        if (coins_collection[i].year <= year && coins_collection[i].price >=
price) {
            coins_selected[found].index = coins_collection[i].index;
            strcpy(coins_selected[found].name, coins_collection[i].name);
            coins_selected[found].price = coins_collection[i].price;
            coins_selected[found].year = coins_collection[i].year;
            found++;
        }
    }
    return found;
}

void print_coins(Coin coins_selected[], int selected_num) {
    if (selected_num == 0) {
        printf("No coins are applicable.\n");
        return;
    }
    printf("Applicable coins:\n");
    for (int i = 0; i < selected_num; ++i) {
        printf("%d. %s %d %d\n", coins_selected[i].index,
coins_selected[i].name, coins_selected[i].price, coins_selected[i].year);
    }
}

int main() {
    int coin_count;
    printf("Enter number of coins: ");
    scanf("%d", &coin_count);

    Coin *coins_collection = malloc(sizeof(Coin) * coin_count);
    if (!coins_collection) {
        printf("Memory allocation failed.\n");
        return 1;
    }

    for (int i = 0; i < coin_count; ++i) {
        coins_collection[i].index = i + 1;
        printf("Enter name for the coin %d: ", i + 1);
        scanf("%s", coins_collection[i].name);
        printf("Enter price for the coin %d: ", i + 1);
        scanf("%d", &coins_collection[i].price);
        printf("Enter year for the coin %d: ", i + 1);
        scanf("%d", &coins_collection[i].year);
    }

    int year, price;
    printf("Enter a year (coins from this year or earlier will be considered):
");
    scanf("%d", &year);
    printf("Enter a price (coins with this price or higher will be considered):
");
}
```

```

scanf("%d", &price);

Coin *coins_selected = malloc(sizeof(Coin) * coin_count);
if (!coins_selected) {
    printf("Memory allocation failed.\n");
    free(coins_collection);
    return 1;
}

int selected_num = select_coins(coins_collection, coins_selected,
coin_count, year, price);
print_coins(coins_selected, selected_num);

free(coins_collection);
free(coins_selected);

return 0;
}

```