

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«Московский технологический университет»**

**МИРЭА**

**Институт информационных технологий (ИИТ)**

Кафедра инструментального и прикладного программного обеспечения (ИППО)

**УТВЕРЖДАЮ**

Заведующий кафедрой

В.А. Мордвинов

« » 2017 г.

**ЗАДАНИЕ**

**на выполнение курсовой работы**

**по дисциплине** «Структуры и алгоритмы обработки данных»

«наименование дисциплины»

Студент: Киясов Ислам Русланович Группа: ИКБО-01-16

**1. Тема курсовой работы**

«Конкурс»

**2. Исходные данные**

* + Язык программирования C++
  + Лекции по структурам и алгоритмам обработки данных: Платонова О. В.

**3. Перечень вопросов, подлежащих разработке, и обязательного графического материала**:

* + Сдача собранной информации по теме
  + Сдача плана работ курсовой работы
  + Разработка исходного кода программы Обязательные последние этапы:
  + Разработка пояснительной записки к КР
  + Подготовка к защите КР
  + Защита КР

**4. Срок представления к защите курсовой работы (КР ): до «29» декабря 2017 г.**

Задание на курсовую

работу выдал «08» сентября 2017 г. Платонова О. В.

Задание на курсовую

<подпись руководителя работы>

<Ф.И.О. руководителя работы>

работу получил «08» сентября 2017 г. Киясов И.Р.

<Подпись студента> <Ф.И.О. студента>

**Мониторинг процесса выполнения курсовой работы (КР)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| №  э т а п а | Наименование этапа курсовой работы | Этап курсовой работы выполнил и представил результаты руководителю работы | Работу по этапу курсовой работы принял на рассмотрение | Рекомендации и замечания по этапу курсовой работы выдал исполнителю | Оценка выполнения этапа курсовой работы | Комментарии руководителя курсовой работы |
| 1 | Сдача собранной информации по теме | 15.09.2017 | 15.09.2017 | 15.09.2017 |  |  |
| 2 | Сдача плана работ курсовой работы | 22.09.2017 | 22.09.2017 | 22.09.2017 |  |  |
| 3 | Разработка исходного кода программы | 23.10.2017 | 23.10.2017 | 23.10.2017 |  |  |
| 4 | Составление документации к программе | 6.11.2017 | 6.11.2017 | 6.11.2017 |  |  |
| 5 | Разработка пояснительной записки к курсовой работе | 24.11.2017 | 24.11.2017 | 24.11.2017 |  |  |
| 6 | Подготовка к защите и защита курсовой работы | 01.12.2017 | .12.2017 | .12.2017 |  |  |



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«Московский технологический университет»**

**МИРЭА**

**Институт информационных технологий (ИИТ)**

Кафедра инструментального и прикладного программного обеспечения (ИППО)

**КУРСОВАЯ РАБОТА**

**по дисциплине**

«Структуры и алгоритмы обработки данных »

«наименование дисциплины»

**Тема:**

**«Конкурс»**

Студент группы ИКБО-01-16 Киясов И.Р.

<код группы> <подпись студента> <ФИО студента>

Руководитель курсовой работы к.т.н., доцент

<должность, звание, уч. степень>

Платонова О. В.

<подпись руководителя> <ФИО руководителя>

Рецензент (при наличии)

<должность, звание, уч. степень>

<подпись рецензента> <ФИО рецензента>

Курсовая работа представлена к

защите « » 2017 г.

<подпись студента>

«Допущен к защите» « » 2017 г.

<подпись руководителя>

Москва 2017

**ЛИСТ ЗАМЕЧАНИЙ**

Оглавление

[1. Техническое задание 5](#_Toc501493189)

[1.1 Введение. 6](#_Toc501493190)

[1.2. Основание для разработки. 6](#_Toc501493191)

[1.3. Назначение разработки. 6](#_Toc501493192)

[1.4. Требования, предъявляемые к программе. 6](#_Toc501493193)

[1.5. Требования к программной документации. 7](#_Toc501493194)

[1.6. Этапы разработки. 7](#_Toc501493195)

[2. Описание программы 7](#_Toc501493196)

[2.1. Общие сведения. 7](#_Toc501493197)

[2.2. Функциональное назначение программы: Программа для проведение конкурсов. 7](#_Toc501493198)

[2.3. Технические средства, которые используются при работе 7](#_Toc501493199)

[программы: 2-ядерный процессор Intel Core m3 с тактовой частотой 1,2 ГГц (ускорение Turbo Boost до 3,0 ГГц) и 4 МБ кэш-памяти третьего уровня. 7](#_Toc501493200)

[2.4. Вызов программы: запустить файл soft. 7](#_Toc501493201)

[2.5. Входные данные: данные вводимые с клавиатуры, или из файла (не обязательно) 7](#_Toc501493202)

[2.6. Выходные данные: значения элементов вектора, выводимые на 7](#_Toc501493203)

[экран. данные выводимые на экран. 7](#_Toc501493204)

[2.7 Описание логической структуры программы. 7](#_Toc501493205)

[Приложение 10](#_Toc501493206)

[Меню. Рисунок 1. 10](#_Toc501493207)

[Новый конкурс. Рисунок 2. 10](#_Toc501493208)

[Добавление участника. Рисунок 3. 10](#_Toc501493209)

[Вывод участников. Рисунок 4. 10](#_Toc501493210)

[Исходный код: 12](#_Toc501493211)

[Библиографический список 20](#_Toc501493212)

## **1. Техническое задание**

### 1.1 Введение.

1.1.1. Наименование программы: «Конкурс».

1.1.2. Краткая характеристика области применения.

Программа разрабатывается для проведения конкурсов.

### 1.2. Основание для разработки.

Учебный план по направлению 09.03.04 «Программная инженерия»

(бакалавры).

### 1.3. Назначение разработки.

Программа предназначена для проведения конкурсов.

### 1.4. Требования, предъявляемые к программе.

1.4.1. Требования к функциональным характеристикам программы.

В программе должны быть реализованы следующие операции:

Количество участников конкурса и количество призовых мест, на которые

претендуют участники, ограничено. Участник конкурса имеет

идентификационный номер, имя. Во время проведения конкурса его

участникам присваивают баллы.

Операции конкурса:

\* открытие конкурса;

\* регистрация участника;

\* окончание регистрации участников;

\* проведение конкурса, регистрация балла, полученного участником

конкурса;

\* окончание конкурса;

\* подведение итогов конкурса, сортировка списка участников по

убыванию баллов и вывод упорядоченного списка участников на

экран и в текстовый файл.

Регистрация участников может происходить в течение нескольких

сеансов работы программы, поэтому при завершении работы программы

(до окончания регистрации участников) необходимо участников,

зарегистрированных в текущем сеансе, дописывать в конец текстового

файла. Регистрация результатов конкурса и подведение итогов

выполняется в течение одного сеанса работы программы.

1.4.2. Требования к техническим средствам, используемым при работе

программы.

Работа программы должна осуществляться на macOS.

1.4.3. Требования к языкам программы и среде разработки программы.

Исходный код программы должен быть записан на языке

программирования С++.

1.4.4. Требования к информационным структурам на входе и выходе

программы.

На входе: ввод данных с клавиатуры, чтение файла.

На выходе: вывод данных на экран, запись в файл.

### 1.5. Требования к программной документации.

Состав программной документации должен включать:

\* техническое задание;

\* описание программы;

\* текст программы.

### 1.6. Этапы разработки.

1.6.1. Обзор способов организации данных и обоснование выбора структуры

данных для эффективного выполнения операций 01.10.2017-3.10.2017.

1.6.2. Разработка программы: 4.10.2017-6.10.2011.

1.6.4. Оформление пояснительной записки 26.10.2017-27.10.2017.

1.6.5. Защита курсовой работы: 2.11.2017.

## **2. Описание программы**

### 2.1. Общие сведения.

2.1.2. Наименование программы: «Конкурс».

2.1.3. Программное обеспечение, необходимое для функционирования

программы: macOS.

2.1.4. Язык программирования, на котором написана программа: С++.

### 2.2. Функциональное назначение программы: Программа для проведение конкурсов.

### 2.3. Технические средства, которые используются при работе

### программы: 2-ядерный процессор Intel Core m3 с тактовой частотой 1,2 ГГц (ускорение Turbo Boost до 3,0 ГГц) и 4 МБ кэш-памяти третьего уровня.

### 2.4. Вызов программы: запустить файл soft.

### 2.5. Входные данные: данные вводимые с клавиатуры, или из файла (не обязательно)

### 2.6. Выходные данные: значения элементов вектора, выводимые на

### экран. данные выводимые на экран.

### 2.7 Описание логической структуры программы.

2.7.1 Основные алгоритмы

Основные функция, составляющие логическую структуру программы:

Int main()

Функция main(Блок-схема) имеет switch, который обрабатывает запросы пользователя.

Блок - схема 1 – алгоритм функции main.

../../../../../Downloads/Untitled%20Diagram%20

2.7.2 Структура программы с описанием функций, составных частей, и связей между ними

Таблица 1 - структуры данных.

|  |  |  |
| --- | --- | --- |
| Название структуры | Роль структуры данных в классе | Описание структуры данных |
| Participant | Структура для создания участника | Данные об у частнике |
| Contest | Структура для работы с конкурсом | Управление конкурсом |

Таблица 2 - поля структуры Participant (Участник)

|  |  |
| --- | --- |
| Название поля и его тип | Роль в программе |
| string name | Имя участника |
| Int Id | Уникальный номер |
| double Score | Очки участника |

Таблица 3 - поля структуры Contest (Конкурс)

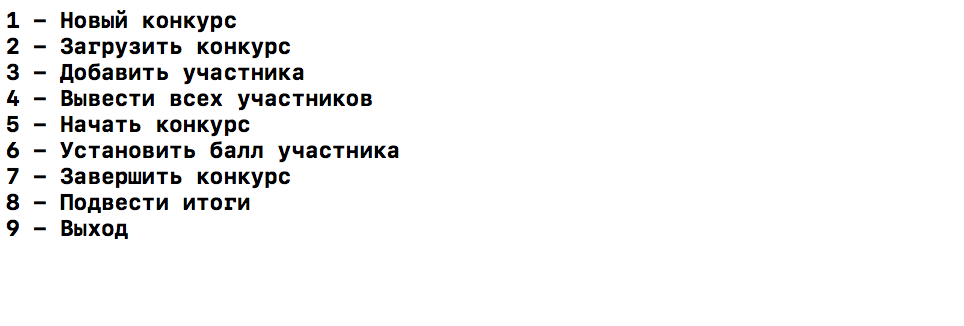
|  |  |
| --- | --- |
| Название поля и его тип | Роль в программе |
| vector<Participant> Participants | Вектор с участниками |
| string Name | Название конкурса |
| int MaxParticipants | Макc. количество участников |
| int PrizePlaceCount; | Количество призов |
| int LoggedParticipantsCount | Количество призовых мест |
| int Status | Статус конкурса |

Таблица 4 - методы класса Contest (Конкурс)

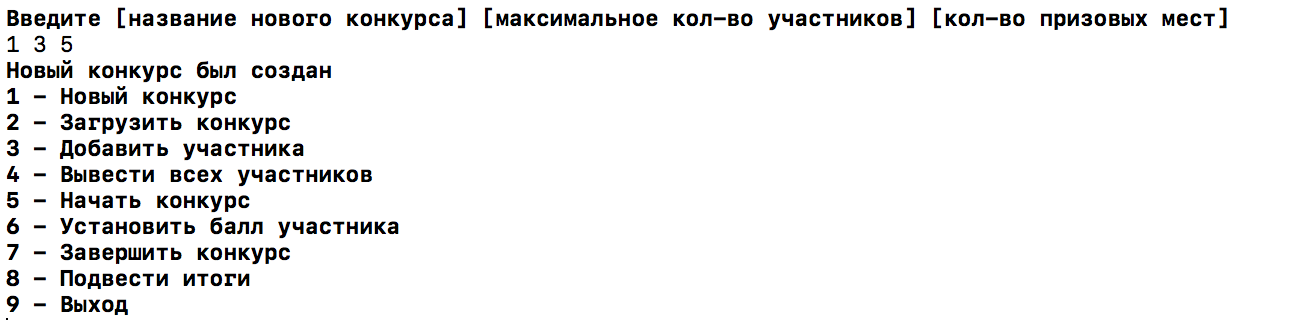
|  |  |
| --- | --- |
| Метод | Роль в программе |
| Contest(string name, int maxParticipants, int prizePlaceCount) | Создаёт новый конкурс |
| Contest(string name, vector<Participant> participants, int maxParticipants, int prizePlaceCount) | Загружает конкурс |
| bool TrySetScore(int id, double score) | Проверят и ставит баллы |
| bool TryAddParticipant(string name) | Проверят и добавляет участника |

## **Приложение**

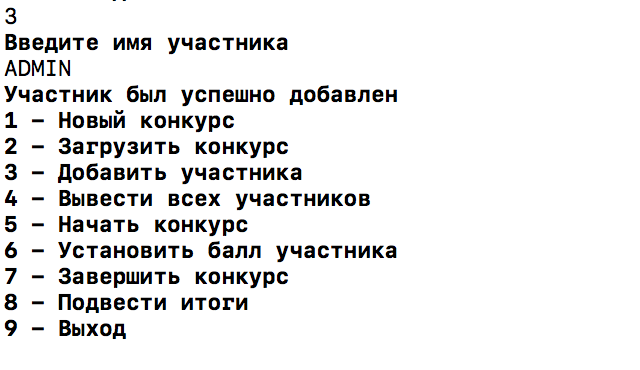
### Меню. Рисунок 1.



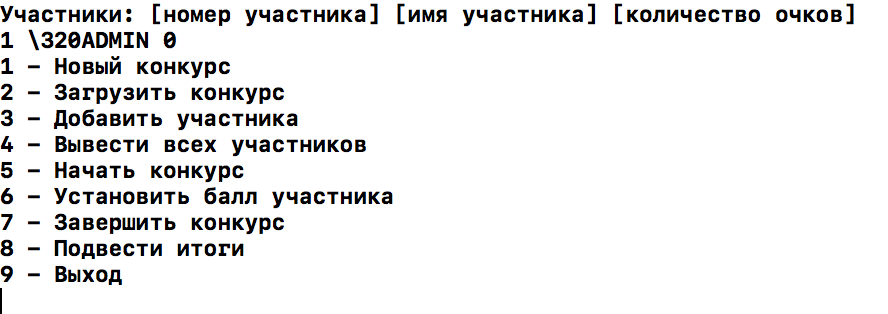
### Новый конкурс. Рисунок 2.



### Добавление участника. Рисунок 3.

****

### Вывод участников. Рисунок 4.

****

### 

### Исходный код:

#include <string>

#include <iostream>

#include <fstream>

#include <vector>

using namespace std;

struct Participant //Участник

{

    string Name;

    int Id;

    double Score;

};

struct Contest //Конкурс

{

    vector<Participant> Participants; //Участники

    string Name;

    int MaxParticipants; //Макc. кол-во участников

    int PrizePlaceCount; //Кол-во призов

    int LoggedParticipantsCount; //Кол-во участников

    int Status;

    Contest(string name, int maxParticipants, int prizePlaceCount) // Новый

    {

        Name = name;

        Participants = vector<Participant>();

        MaxParticipants = maxParticipants;

        PrizePlaceCount = prizePlaceCount;

        LoggedParticipantsCount = 0;

        Status = 0;

    }

    Contest(string name, vector<Participant> participants, int maxParticipants, int prizePlaceCount)

    {

        Name = name;

        Participants = participants;

        MaxParticipants = maxParticipants;

        PrizePlaceCount = prizePlaceCount;

        LoggedParticipantsCount = participants.size();

        Status = 0;

    }

    bool TrySetScore(int id, double score) //Ставим баллы

    {

        if (Participants.size() < id)

        {

            return false;

        }

        else

        {

            Participants[id - 1].Score = score;

            return true;

        }

    }

    bool TryAddParticipant(string name) //Добавляем участника

    {

        if (Participants.size() == MaxParticipants)

            return false;

        else

        {

            Participant temp;

            temp.Id = Participants.size() + 1;

            temp.Name = name;

            temp.Score = 0;

            Participants.push\_back(temp);

            return true;

        }

    }

};

bool compareParticipants(const Participant &first, const Participant &second)

{

    return first.Score > second.Score;

}

Contest ResumeContest(string name)

{

    ifstream file;

    file.open(name + "\_participants.txt");

    int maxParticipants;

    int prizePlaceCount;

    file >> maxParticipants >> prizePlaceCount;

    vector<Participant> result;

    int counter = 1;

    while (!file.eof()) //Конец файла

    {

        Participant temp;

        getline(file, temp.Name);

        if (temp.Name != "")

        {

            temp.Id = counter++;

            temp.Score = 0;

            result.push\_back(temp);

        }

    }

    file.close();

    return Contest(name, result, maxParticipants, prizePlaceCount);

}

bool IsContestExist(string name)

{

    ifstream file(name + "\_participants.txt");

    return file.good();

}

void GetResults(vector<Participant> &participants)

{

    sort(participants.begin(), participants.end(), compareParticipants); //Сортировка по баллам

}

Contest StartNewContest(string name, int maxParticipants, int prizePlaceCount)

{

    return Contest(name, maxParticipants, prizePlaceCount);

}

void PrintParticipants(Contest contest)

{

    cout << "Участники: [номер участника] [имя участника] [количество очков]" << endl;

    for (int i = 0; i < contest.Participants.size(); i++)

    {

        cout << contest.Participants[i].Id << " " << contest.Participants[i].Name << " " << contest.Participants[i].Score << endl;

    }

}

void PrintResult(Contest contest)

{

    cout << "Название сореванования: " << contest.Name << endl;

    cout << "Количество участников " << contest.Participants.size() << endl;

    cout << "Максимальное количество участников в конкурсе " << contest.MaxParticipants << endl;

    cout << "Количество призовых мест " << contest.PrizePlaceCount << endl;

    cout << "Результаты конкурса: [место] [номер участника] [имя участника] [количество очков] " << endl;

    cout << "Победители:" << endl;

    for (int i = 0; i < min((int)contest.Participants.size(), contest.PrizePlaceCount); i++)

    {

        cout << i + 1 << " " << contest.Participants[i].Id << " " << contest.Participants[i].Name << " " << contest.Participants[i].Score << endl;

    }

    cout << "Остальные участники:" << endl;

    for (int i = contest.PrizePlaceCount; i < contest.Participants.size(); i++)

    {

        cout << i + 1 << " " << contest.Participants[i].Id << " " << contest.Participants[i].Name << " " << contest.Participants[i].Score << endl;

    }

}

void PrintResultToFile(Contest contest)

{

    ofstream file;

    file.open(contest.Name + "\_result.txt");

    file << "Название сореванования: " << contest.Name << endl;

    file << "Количество участников " << contest.Participants.size() << endl;

    file << "Максимальное количество участников в конкурсе " << contest.MaxParticipants << endl;

    file << "Количество призовых мест " << contest.PrizePlaceCount << endl;

    file << "Результаты конкурса: [место] [номер участника] [имя участника] [количество очков]"<<endl;

    file << "Победители:" << endl;

    for (int i = 0; i < min((int)contest.Participants.size(), contest.PrizePlaceCount); i++)

    {

        file << contest.Participants[i].Id << " " << contest.Participants[i].Name << " " << contest.Participants[i].Score << endl;

    }

    file << "Остальные участники:" << endl;

    for (int i = contest.PrizePlaceCount; i < contest.Participants.size(); i++)

    {

        file << contest.Participants[i].Id << " " << contest.Participants[i].Name << " " << contest.Participants[i].Score << endl;

    }

    file.close();

}

void PrintParticipantsToFile(Contest contest)

{

    if (!IsContestExist(contest.Name)) // Проверка на налич файла

    {

        ofstream f(contest.Name + "\_participants.txt");

        f << contest.MaxParticipants << " " << contest.PrizePlaceCount << endl;

        f.close();

    }

    ofstream file;

    file.open(contest.Name + "\_participants.txt", std::ios::app); // Добавляем в конец файла

    for (int i = contest.LoggedParticipantsCount; i < contest.Participants.size(); i++)

    {

        file << contest.Participants[i].Name << endl;

    }

    file.close();

}

int main()

{

    setlocale(LC\_ALL, "Russian");

    Contest currentContest("", 0, 0);

    while (true)

    {

        string input;

        cout << "1 - Новый конкурс\n2 - Загрузить конкурс\n3 - Добавить участника\n4 - Вывести всех участников\n";

        cout << "5 - Начать конкурс\n6 - Установить балл участника\n7 - Завершить конкурс\n8 - Подвести итоги\n9 - Выход\n";

        char c;

        cin >> c;

        switch (c)

        {

            case '1':

                if (currentContest.MaxParticipants != 0)

                    PrintParticipantsToFile(currentContest);

                cout << "Введите [название нового конкурса] [максимальное кол-во участников] [кол-во призовых мест]" << endl;

                int maxParticipants, prizePlaces;

                cin >> input >> maxParticipants >> prizePlaces;

                currentContest = StartNewContest(input, maxParticipants, prizePlaces);

                cout << "Новый конкурс был создан" << endl;

                break;

            case '2':

                if (currentContest.MaxParticipants != 0)

                    PrintParticipantsToFile(currentContest);

                cout << "Введите конкурса, который вы хотите загрузить" << endl;

                cin >> input;

                if (!IsContestExist(input))

                {

                    cout << "Такой конкурс не найден" << endl;

                    currentContest = Contest("", 0, 0);

                }

                else

                {

                    currentContest = ResumeContest(input);

                    cout << "Конкурс загружен" << endl;

                }

                break;

            case '3':

                if (currentContest.MaxParticipants == 0)

                {

                    cout << "Нельзя добавить участника пока не выбран конкурс" << endl;

                    break;

                } else if (currentContest.Status != 0)

                {

                    cout << "Нельзя добавить участника в конкурс который уже был начат" << endl;

                    break;

                }

                cout << "Введите имя участника" << endl;

                cin >> input;

                if (currentContest.TryAddParticipant(input))

                    cout << "Участник был успешно добавлен" << endl;

                else

                    cout << "Превышено количество участников конкурса" << endl;

                break;

            case '4':

                if (currentContest.MaxParticipants == 0)

                    cout << "Конкурс не выбран" << endl;

                else

                    PrintParticipants(currentContest);

                break;

            case '5':

                if (currentContest.MaxParticipants == 0)

                    cout << "Конкурс не выбран" << endl;

                else if (currentContest.Status == 1)

                    cout << "Нельзя начать конкурс который уже был начат" << endl;

                else

                {

                    currentContest.Status = 1;

                    cout << "Конкурс начат" << endl;

                }

                break;

            case '6':

                if (currentContest.MaxParticipants == 0)

                    cout << "Конкурс не выбран" << endl;

                else if (currentContest.Status != 1)

                    cout << "Конкурс недоступен для проставки баллов участников" << endl;

                else

                {

                    PrintParticipants(currentContest);

                    cout << "Введите [номер участника] [балл]";

                    int n;

                    double score;

                    cin >> n >> score;

                    if (!currentContest.TrySetScore(n, score))

                        cout << "Участник с таким номером не найден" << endl;

                    else

                        cout << "Балл участника успешно установлен" << endl;

                }

                break;

            case '7':

                if (currentContest.MaxParticipants == 0)

                    cout << "Конкурс не выбран" << endl;

                else if (currentContest.Status != 1)

                    cout << "Невозможно завершить этот конкурс" << endl;

                else

                {

                    cout << "Конкурс завершён, можете подвести итоги" << endl;

                    currentContest.Status = 2;

                }

                break;

            case '8':

                if (currentContest.MaxParticipants == 0)

                    cout << "Конкурс не выбран" << endl;

                else if (currentContest.Status != 2)

                    cout << "Невозможно подвести итоги незавершенного конкурса" << endl;

                else

                {

                    GetResults(currentContest.Participants);

                    PrintResult(currentContest);

                    PrintResultToFile(currentContest);

                }

                break;

            case '9':

                if (currentContest.MaxParticipants != 0)

                    PrintParticipantsToFile(currentContest);

                return 0;

            default:

                cout << "Неизвестная команда" << endl;

        }

    }

    return 0;

}

# Библиографический список

1. Ориг.название: C++. Базовый курс. Автор: Герберт Шилдт. Издательство: Вильямс. Количество страниц: 624. Год выпуска: 2015.
2. Ориг.название: Технология программирования на С++. Автор: Литвиненко Н. А.. Издательство: БХВ-Петербург. Количество страниц: 281. Год выпуска: 2010.
3. Ориг.название: Изучаем Си. Автор: А. Крупник. Издательство: Питер. Количество страниц: 233. Год выпуска: 2001.
4. Ориг.название: Си++. От дилетанта до профессионала. Автор: Романов Е.Л..

Издательство: ФГБОУ ВО "Новосибирский государственный технический университет". Количество страниц: 600. Год выпуска: 2014.