

Объектно-ориентированное программирование

Преподаватель: Рябычина Ольга Петровна

+375291550082

<https://vk.com/ryabychina>

по ООП:

Лекции

Лабораторные работы

Курсовая работа

Зачёт

Вся информация по курсовой работе – см.
в документе:

Методические указания по курсовой работе по дисциплине ООП
для **заочного**

Методические указания и задания по выполнению курсовой работы по дисциплине «ООП»

Цель работы: закрепить на практике навыки объектно-ориентированного проектирования программного обеспечения, а также использования различных средств языка C++ для разработки прикладного программного обеспечения.

Исходные данные к работе

1. Индивидуальное задание выбирается из списка, приведенного в данном файле.
2. Язык программирования C++.
3. Рекомендуемая среда разработки Microsoft VisualStudio.
4. Вид приложения – объектно-ориентированное консольное.
5. Хранение данных в памяти должно быть организовано посредством классов (объектно-ориентированная база данных).

6. Физически данные должны храниться в файлах. При запуске программы происходит загрузка данных из файла в массив, либо другую структуру данных (допускается использование стандартных контейнеров библиотеки STL – vector, list), компонентами которой являются объекты. Перед завершением работы программы, либо при выборе соответствующего пункта меню происходит сохранение данных в текстовый файл (перезапись исходного файла данных).

7. Построение программного кода должно соответствовать правилам, определенным в документе «C++ CodeConvention».

8. Текст пояснительной записки оформляется в соответствии со стандартом предприятия СТП 01–2017 (<http://bsac.by/students>)

Требования к программной разработке

В результате должен быть разработан комплекс программных средств, позволяющих осуществлять обработку данных **согласно варианту задания**.

Программное обеспечение основано на работе с файлами в консольном режиме. **Использование СУБД не допускается!**

Внизу программных форм должна выводиться информация о разработчике, руководителе проекта, год разработки. Предусмотреть возможность вставки на страницу текущей даты и времени.*

*не обязательно, влияет на оценку

Первым этапом работы программы является авторизация.

Данные об учетных записях пользователей хранятся в файле следующей структуры:

-login;

-password;

-role.

Предусмотреть возможность входа в качестве администратора (в этом случае, например, $role = 1$) и в качестве пользователя (в этом случае, например, $role = 0$).

Изначально записать в файл учетную запись администратора.

Предусмотреть возможность шифрования паролей пользователей.*

Вторым этапом работы программы является

собственно работа с файлом данных, которая становится возможной после прохождения авторизации.

Комплекс программных средств включает две прикладных программы: **«Администратор»** и **«Пользователь»**.

***Программа «Администратор»* должна обеспечивать выполнение следующих функциональных возможностей:**

1. Управление учетными записями пользователей:

- просмотреть все учетные записи;
- добавить учетную запись;
- отредактировать учетную запись;
- удалить учетную запись.

2. *Работа с файлом данных:

- создать файл;
- открыть файл;
- удалить файл.

3. Работа с данными:

а) режим редактирования:

- просмотреть все данные;
- добавить запись;
- удалить запись (для этого необходимо ввести порядковый номер конкретной записи);
- редактировать запись (для этого необходимо ввести порядковый номер конкретной записи);

Программа «Пользователь» должна обеспечивать выполнение следующих функциональных возможностей:

- просмотреть все данные;
- выполнить задачу (задачи), указанную в индивидуальном задании;
- выполнить поиск данных;
- выполнить сортировку по различным полям в алфавитном порядке / в порядке убывания.

После авторизации пользователю должно быть выведено меню, из которого, путем ввода номера соответствующего пункта меню, пользователь переходит к выполнению нужной ему операции. Примеры пользовательских интерфейсов приведены на рис. 1-3 на следующих слайдах.

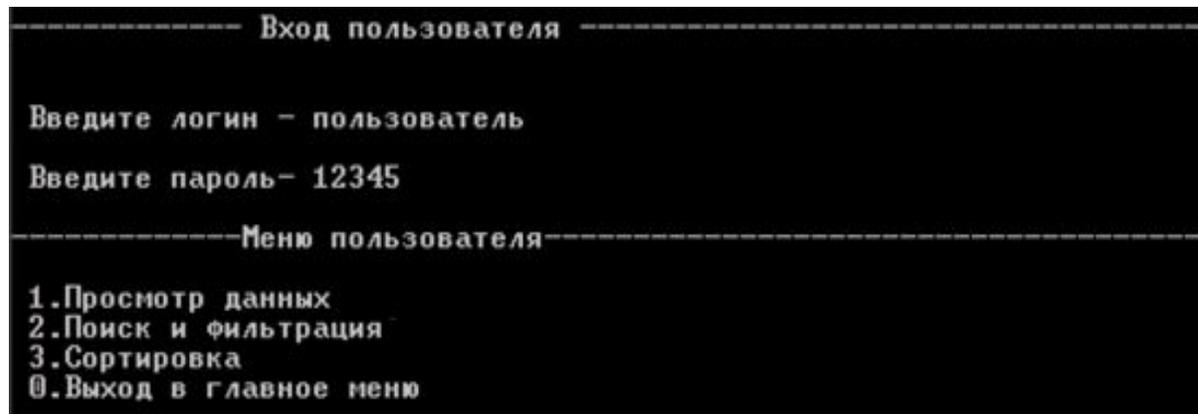


Рис.1 – Пример интерфейса для авторизации и меню для пользователя

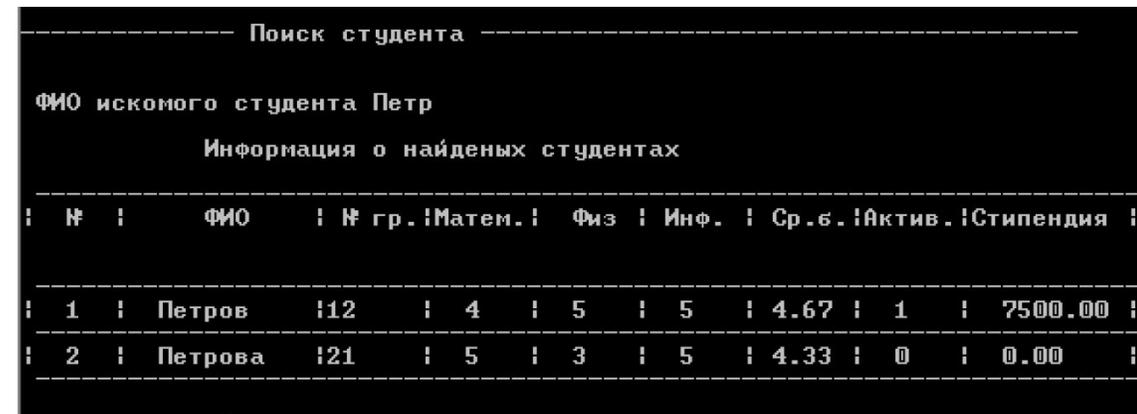


Рис.2 – Пример интерфейса для функциональной возможности «выполнить поиск»

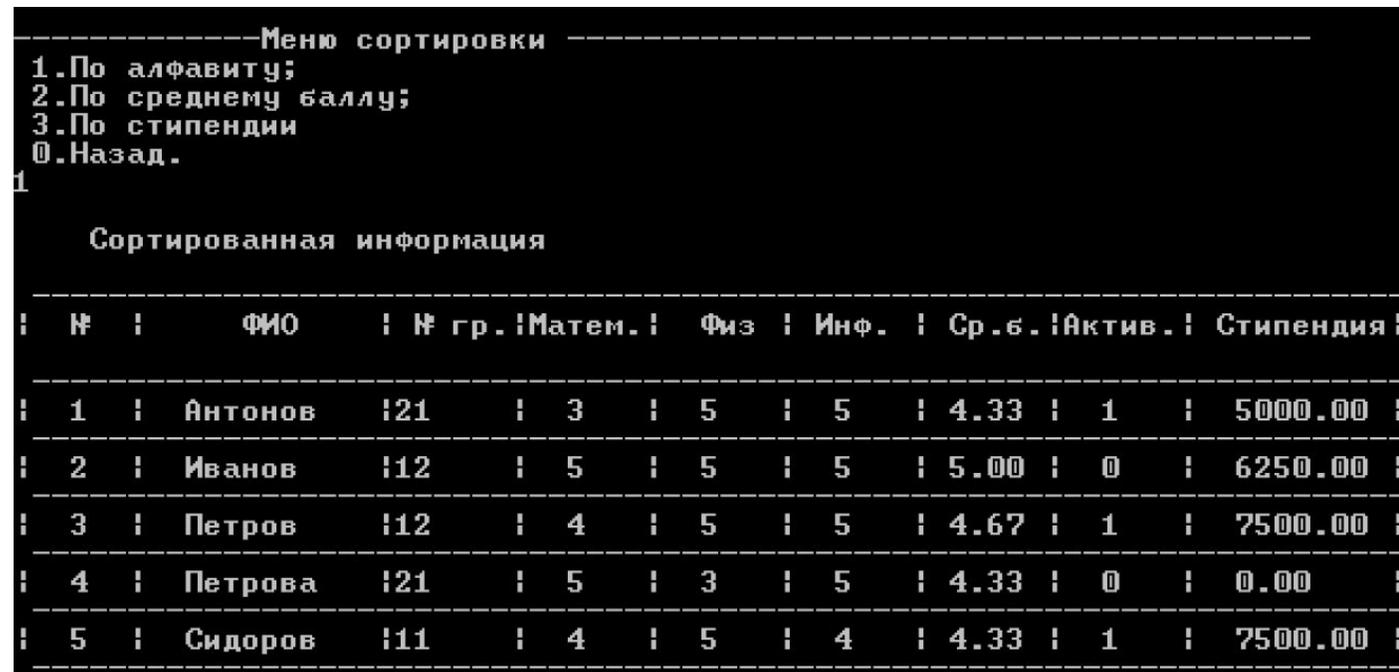


Рис. 3 – Пример интерфейса для функциональной возможности «выполнить сортировку»

Предусмотреть:

- обработку исключительных ситуаций (файл с данными не найден, ввод некорректных данных) данный пункт может быть дополнен по согласованию с руководителем проекта;
- возможность возврата назад (навигация);
- в случае отсутствия результатов поиска должно быть выведено сообщение «По Вашему запросу ничего не найдено.».
- запрос на подтверждение удаления вида «Вы действительно хотите удалить файл (запись)?»;
- вывод сообщения о результате выполнения каждой из операций (например, «Запись успешно создана»). Перечень выводимых сообщений должен быть согласован с руководителем проекта.

Работа меню должна быть реализована в виде класса.

Каждая курсовая работа, включая программную разработку, должна быть уникальной, т.к. подразумевает индивидуальную работу над своим заданием.

За две недели до даты защиты пояснительная записка должна быть представлена на рецензирование руководителю курсовой работы.

На защите курсовой работы студент демонстрирует презентацию программного продукта, которая включает следующую информацию:

- Тема и данные о разработчике
- Предметная область
- Перечень функциональных возможностей в привязке к предметной области. (Не просто поиск записей, а конкретно, например, поиск билетов на поезд по дате)
- Среда разработки.
- Блок-схемы.
- Демонстрация функциональных возможностей в порядке приоритета их важности.
- Выводы о проделанной работе.

При выставлении отметки за курсовую работу учитывается:

- Качество разработанной проектной, а также сопроводительной документации (пояснительная записка)
- Объем реализованных функциональных требований
- Качество работы представляемого ПО (наличие ошибок и неправильных действий во время работы)
- Качество используемых алгоритмов
- Качество оформления программного кода
- Качество презентации, ее соответствие требованиям, изложенным в данных рекомендациях
- Ответы на вопросы преподавателя на защите

Структура пояснительной записки

Титульный лист

Задание по курсовой работе (*выполненное в соответствии с требованиями стандарта предприятия СТП 01–2017*
заполненное и подписанное студентом, преподавателем и заведующим кафедрой)

Содержание

Введение

1. Теоретическое обоснование

2. Анализ и разработка программного обеспечения

2.1 Анализ предметной области и формулирование функциональных требований

2.2 Разработка структуры программы и алгоритмов работы модулей программы

2.3. Описание программного кода

3. Демонстрация работы

3.1. Руководство пользователя

3.2. Тестирование программы

Заключение

Литература

Приложения

А Блок-схемы алгоритмов

Б Листинг программного кода

В Презентация

Во **введении** должна быть четко прописана цель разработки данного ПО, а также актуальность разработки. Цель должна быть сформулирована с точки зрения предметной области. Записать в качестве цели закрепление навыков программирования на языке C++ **нельзя**.

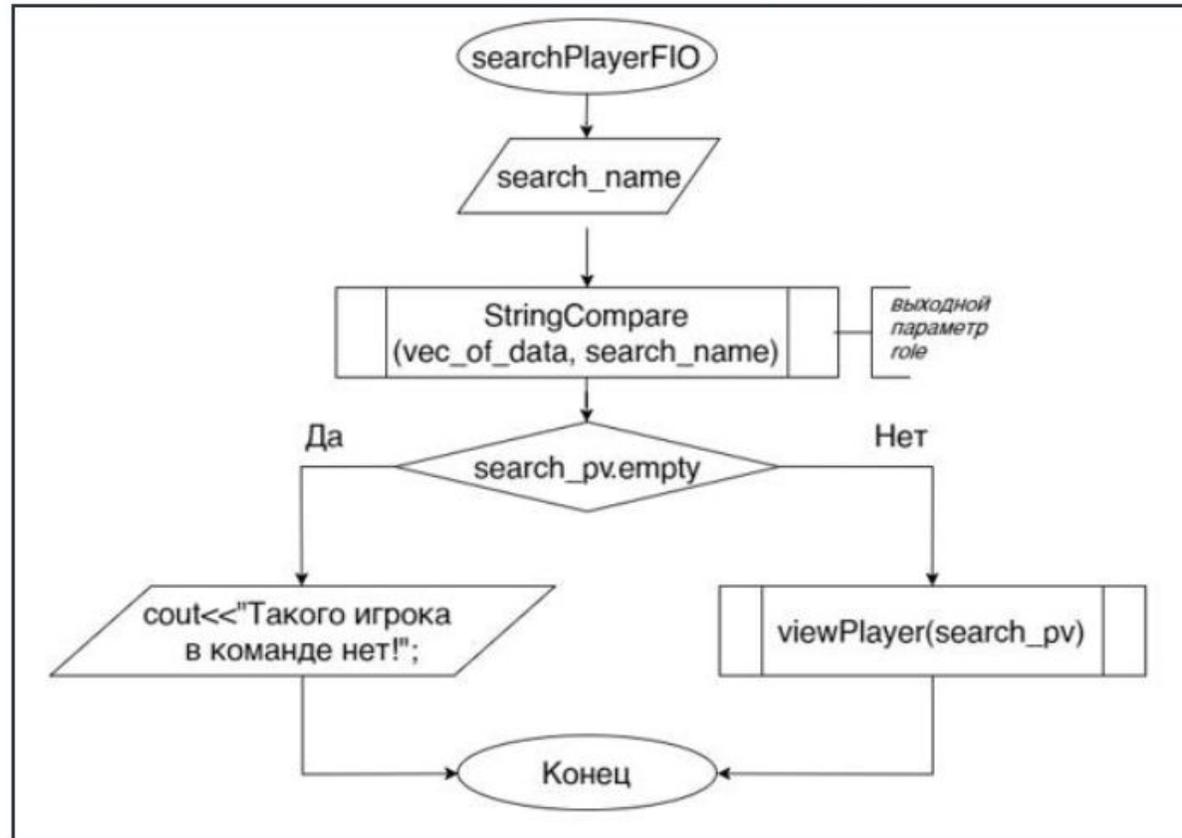
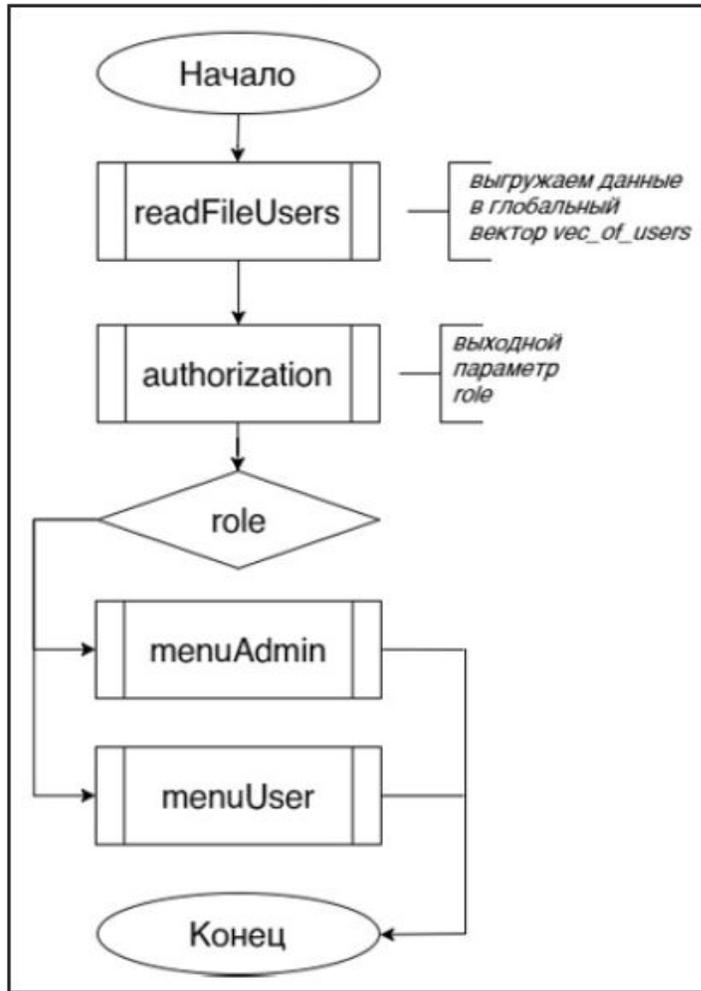
Теоретическое обоснование. Здесь дается полный ответ на теоретический вопрос (согласно Вашему варианту задания).

Анализ предметной области и формулирование функциональных требований. Здесь приводится полный текст Вашего варианта задания. Четко обозначается предметная область, кратко описываются правила и закономерности ее функционирования, на основании которых позднее формулируются функциональные требования к разрабатываемому ПО. (В качестве модели можно привести UML-диаграмму вариантов использования.)

Разработка структуры программы подразумевает перечисление и краткое описание свойств и методов для классов. Код методов **НЕ ПРИВОДИТСЯ**, т.к. на этом этапе он еще не существует. (В качестве модели можно привести UML-диаграммы классов.)

Разработка алгоритмов работы модулей программы включает блок-схемы алгоритмов с описанием их работы для **трех** пользовательских функций. Обратите внимание, что так как алгоритмы разрабатываются до непосредственного кодирования программы, то они **не могут содержать просто копии строчек кода**. Алгоритм может содержать словесные инструкции с упоминанием имен структур, массивов/векторов, модулей, описанных выше.

Разработка алгоритмов работы программы



Online: <https://www.draw.io/>

Описание программного кода дается краткое словесное описание всех модулей, функций и т.д., разработанной программы.

Руководство пользователя подразумевает подробное пошаговое описание работы программы.

Тестирование программы здесь приводятся тестовые случаи работы программы, проиллюстрированные программными формами консоли. Допускается помещать программные формы в отдельное приложение.

В **заключении** прописываются выводы о проделанной работе. Что было конкретно сделано студентом. Далее можно перечислить основные функциональные возможности, область применения, а также слабые места и пути дальнейшего развития ПО.

Текст должен быть составлен не общими фразами, а конкретно в привязке в вашему варианту задания. Объем 1 страница. **Писать, что работа выполнена в полном объеме в соответствии с требованиями нельзя.** Это определит руководитель при рецензировании, а также комиссия на защите.

В качестве достоинств разработанного ПО нельзя писать про качественный интерфейс, т.к. это не является предметом изучения в данном учебном заведении, следовательно, не формулировались критерии как это доказать.

Литература. Приводится список печатных, а также интернет-ресурсов, используемых при выполнении работы. Оформляется в соответствии с требованиями стандарта предприятия.

← → ↻ Ненадежный | bsac.by/students#Какоформитьдиплом,курсовой,другиедокументы

БЕЛОРУССКАЯ ГОСУДАРСТВЕННАЯ АКАДЕМИЯ СВЯЗИ

Витебский филиал

3 учебная неделя

Академия Образование Наука Поступающим **Обучающимся**

Обучающимся

Как оформить диплом, курсовой, другие документы

- Бланк заявления на утверждение темы ДП
- Титульный лист ПЗ ДП (ДР)
- Бланк задания КП (КР)
- Бланк задания на ДП
- Титульный лист ПЗ КП ССО
- Титульный лист ПЗ КР ВО
- Стандарт предприятия. Курсовое и дипломное проектирование. Общие требования**

- Как оформить диплом, курсовой, другие документы**
- Учащемуся отделения заочного и дистанционного образования
- Студенту факультета заочного и дистанционного образования
- Стоимость обучения
- Расписания групп дневной формы обучения

Приложения. Графический материал, приведенный в приложениях, должен быть распечатан как чертеж на листах с рамками. Программный код также должен быть распечатан на листах с рамкой в соответствии с требованиями стандарта предприятия.

Варианты заданий к курсовой работе по дисциплине «ООП» берем в Методических указаниях по курсовой работе по дисциплине ООП для **заочного**

Например: Вариант 1

1. Структура программы на C++: директивы препроцессора, описание типов пользователя, прототипы функций, глобальные переменные, описание функций (объяснить суть понятий, привести примеры). **ТЕОРЕТИЧЕСКИЙ ВОПРОС**

2. Разработка программы учета товаров на складе

Программа предоставляет сведения о товарах, имеющихся на складе: наименование товара; количество единиц товара; цена единицы товара; дата поступления товара на склад.

Вывести в алфавитном порядке список товаров, хранящихся более x месяцев, стоимость которых превышает y рублей (x, y вводятся с клавиатуры).

ПРАКТИЧЕСКОЕ ЗАДАНИЕ

Сроки:

Сдача студентом законченной работы
1.03.2021г

Литература

- Труханович, Т.Л. Основы алгоритмизации и программирования : учеб. пособие / Т.Л. Труханович, О.П. Рябычина. Минск : Белорусская государственная академия связи, 2017 – 212 с. : ил.
- Лафоре Р. Объектно-ориентированное программирование с C++ / Р. Лафоре. – СПб: Питер, 2018. – 928 с.
- Шилдт, Г. C++. Базовый курс / Г. Шилдт. – М.: Вильямс, 2015. – 700 с.

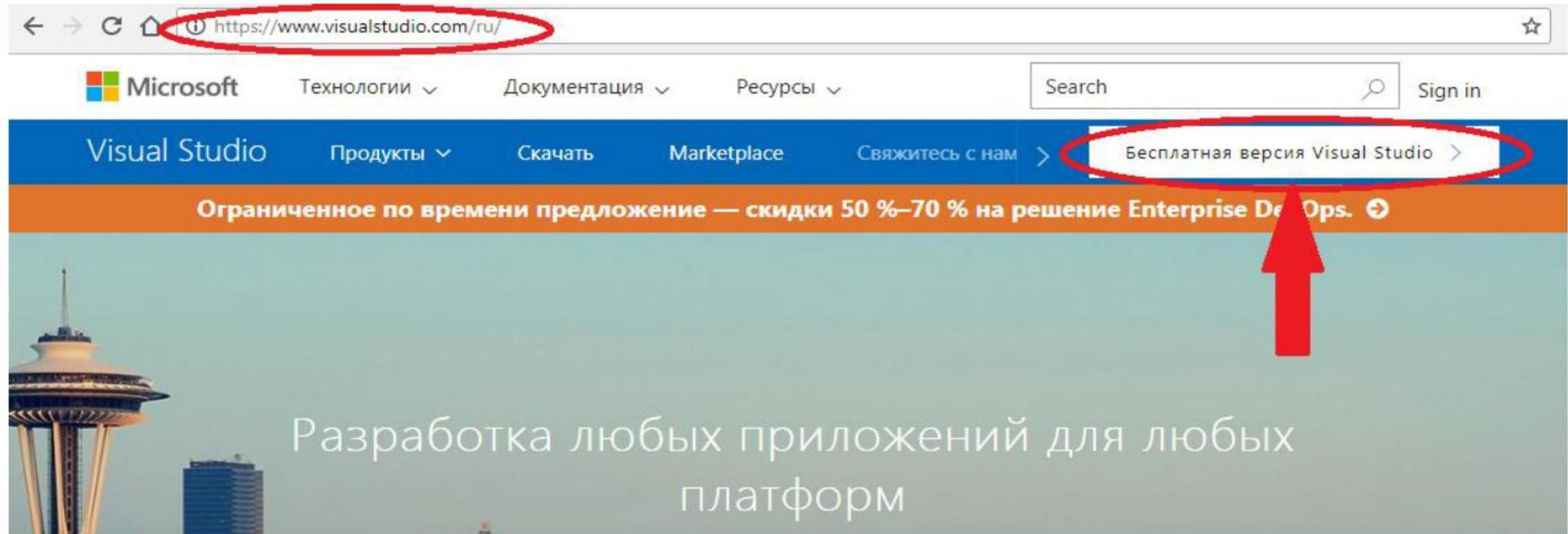
Интернет-ресурсы

1. <https://metanit.com/cpp/>
2. <https://www3.ntu.edu.sg/home/ehchua/programming/index.html>
3. <http://www.c-cpp.ru/books/yazyk> -
4. <http://cpp-reference.ru/patterns/>

Скачиваем и устанавливаем Microsoft Visual Studio

Скачиваем официальную **БЕСПЛАТНУЮ** версию Microsoft Visual Studio с сайта <https://www.visualstudio.com/ru/>

Microsoft Visual Studio **Community** (2013/2015/2017)



The screenshot shows the Microsoft Visual Studio website in Russian. The browser's address bar is circled in red, containing the URL <https://www.visualstudio.com/ru/>. The Microsoft logo is in the top left, followed by navigation links: Технологии, Документация, and Ресурсы. A search bar and a Sign in button are on the right. The main navigation bar is blue and contains: Visual Studio, Продукты, Скачать, Marketplace, Свяжитесь с нами, and **Бесплатная версия Visual Studio** (circled in red). Below this is an orange banner with the text: Ограниченное по времени предложение — скидки 50%–70% на решение Enterprise DevOps. A red arrow points to the highlighted link. The main content area features a background image of the Space Needle and the text: Разработка любых приложений для любых платформ.

Скачиваем и устанавливаем Microsoft Visual Studio

← → ↻ 🏠 <https://www.visualstudio.com/ru/free-developer-offers/> ☆

Visual Studio Community

Полнофункциональная интегрированная среда разработки для создания некорпоративных приложений для Windows, Android и iOS, а также современных веб-приложений и облачных служб.



Скачать

Visual Studio Team Services

Инструменты Agile, Git и непрерывная интеграция для любого языка и любой операционной системы.



Бесплатная учетная запись

Visual Studio Code

Мощный бесплатный редактор на основе исходного кода, работающий на любой платформе.



Visual Studio
Installing - Visual Studio Community 2017 (15.0.26228.4)

Workloads Individual components Language packs

Windows (3)

- Universal Windows Platform development
Create applications for the Universal Windows Platform with C#, VB, JavaScript, or optionally C++.
- Desktop development with C++
Build classic Windows-based applications using the power of the Visual C++ toolset, ATL, and optional features like MFC and...
- .NET desktop development
Build WPF, Windows Forms and console applications using the .NET Framework.

Web & Cloud (5)

- ASP.NET and web development
Build web applications using ASP.NET, ASP.NET Core, HTML, JavaScript, and CSS.
- Azure development
Azure SDK, tools, and projects for developing cloud apps and creating resources.
- Node.js development
Build scalable network applications using Node.js, an asynchronous event-driven JavaScript runtime.
- Data storage and processing
Connect, develop and test data solutions using SQL Server, Azure Data Lake, Hadoop or Azure ML.
- Office/SharePoint development

Location
C:\Program Files (x86)\Microsoft Visual Studio\2017\Community

Install size: 600 MB
Install

Summary
Visual Studio core editor
The Visual Studio core shell experience, including syntax-aware code editing, source code control and work item management.

By continuing, you agree to the license for the Visual Studio edition you selected. We also offer the ability to download other software with Visual Studio. This software is licensed separately, as set out in the 3rd Party Notices or in its accompanying license. By continuing, you also agree to those licenses.

ВОПРОСЫ К ЗАЧЕТУ

1. Объектно-ориентированный подход к организации программ. Характеристики ОО языков программирования.
2. Принципы ООП: инкапсуляция, полиморфизм, наследование.
3. Синтаксис объявления класса. Режимы доступа к элементам класса.
4. Синтаксис описания методов класса.
5. Создание экземпляров класса. Вызов методов.
6. Массивы объектов. Создание. Доступ к элементам. Вызов методов.
7. Указатели на объекты. Ссылки.
8. Конструкторы и деструкторы.
9. Методы доступа к закрытым членам класса (getter и setter –методы).
10. Константные методы.
11. Статические члены класса.
12. Указатель this. Примеры использования.
13. Конструктор копирования.
14. Перегружаемые функции.
15. Перегрузка операций.

ВОПРОСЫ К ЗАЧЕТУ (продолжение)

16. Дружественные функции.
17. Агрегация композиция классов. Примеры.
18. Наследование. Синтаксис объявления класса-потомка. Режимы доступа при наследовании.
19. Переопределение членов базового класса в производном классе.
20. Создание экземпляров классов-потомков. Вызов методов.
21. Наследование. Конструкторы и деструкторы. Передача параметров из производного в базовый класс.
22. Виртуальные функции и полиморфизм.
23. Чисто виртуальные функции и абстрактные классы. Примеры.
24. Интерфейсы.
25. Шаблоны функций и классов.
26. Обработка исключительных ситуаций.
27. Стандартные средства обработки структур данных в ОО языках программирования.
28. Организация файлового ввода-вывода при объектном подходе.
29. Общие сведения о шаблонах проектирования.
30. Приемы разработки многофайлового программного обеспечения.