

## Программа курса «Программирование на С++ (базовый уровень)»

Номер	Название темы	Количество часов	Описание темы
1	Вступление	4	Краткая история, предназначение и ниша языка С++. Демонстрация примера «Hello, world». Описание инструментов разработчика. Процесс сборки программ. Правила оформления кода. Где искать информацию. Проработка материала дополнительной литературы по теме
2	Основные понятия	10	Переменные. Выражения. Декларация и инициализация. Точка входа. Аргументы функций и возврат значений. Директива препроцессора #include. Директива using. Пространства имён. Основы потоков ввода/вывода. Хранение данных в ОЗУ. Целочисленные переменные и литералы. Встроенные целочисленные типы. Целочисленные типы фиксированной длины. Инициализация переменных. Квалификатор const. Неявное приведение типов. Математические операторы. Операторы приведения типа c-style cast, static_cast. Оператор sizeof(). Псевдотип auto. Перечисления. Статические переменные и функции
3	Функции и управление потоком	10	Функции и процедуры. Разные способы указания возвращаемого типа. Значения аргументов по умолчанию. Перегрузка функций. Контейнер std::string. Интерактивное взаимодействие с программой. Операторы if и switch.

			Циклы for, while и do-while. Области видимости. Логические операторы и операторы сравнения. Тернарный оператор. Оператор goto. Рекурсия
4	Указатели и ссылки	10	Массивы. Строковый тип данных. Указатели, типизированные и обобщённые. Нулевой указатель. Операции взятия адреса и разыменования. Арифметика указателей. Операторы new и delete, понятия стека и кучи. Ссылки. Оператор приведения типа reinterpret_cast. Контейнеры unique_ptr и shared_ptr. Указатели на функции. Оператор decltype. Передача функций как аргументов.
5	Классы	10	Классы и структуры. Спецификаторы доступа, инкапсуляция. Время жизни объекта. Способы инициализации членов класса. Методы. Конструктор и деструктор. Указатель this. Оператор «стрелка». Структуры хранения данных: стек, очередь, односвязный и двусвязный список, двоичное дерево, красно-чёрное дерево. Контейнеры std::vector, std::array, std::list, std::set, std::map.
6	Наследование	10	Делегаты, события и лямбды. Как делегаты хранят ссылки на методы: через события / анонимные методы / лямбда-выражения / LINQ показана истинная сила делегатов.
7	Итоговая аттестация	2	Зачёт

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение высшего образования  
«Московский государственный технический университет имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)



УТВЕРЖДАЮ  
Первый проректор –  
проректор по учебной работе  
МГТУ им. Н.Э. Баумана  
Б.В. Падалкин  
29 февраля 2024 г.

Дополнительное профессиональное образование

ДОПОЛНИТЕЛЬНАЯ ПРОФЕССИОНАЛЬНАЯ ПРОГРАММА  
ПРОГРАММА ПОВЫШЕНИЯ КВАЛИФИКАЦИИ  
«Программирование на C++ (базовый уровень)»

Регистрац. № 05.22.23.02.96

Москва, 2024

**АВТОРЫ ПРОГРАММЫ:**

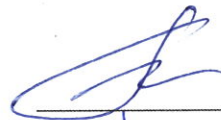
Ст. преподаватель каф. ИУ4



В.В. Терехов


**СОГЛАСОВАНО:**

Начальник УСП



Т.А. Гузева

Директор  
Центра дополнительного образования



М.В. Стоянова

## Оглавление

<b>1. ОБЩАЯ ХАРАКТЕРИСТИКА ДПП.....</b>	<b>4</b>
1.1. Цель ДПП.....	4
1.2. Планируемые результаты обучения.....	4
1.3. Дополнительные характеристики ДПП.....	4
1.4. Перечень профессиональных компетенций в рамках имеющейся квалификации, качественное изменение которых осуществляется в результате обучения.....	4
1.5. Соответствие видов деятельности профессиональным компетенциям и их составляющих.....	5
<b>2. УЧЕБНЫЙ ПЛАН ДПП .....</b>	<b>6</b>
2.1. Категория слушателей ДПП.....	6
2.2. Общая трудоёмкость программы, аудиторная и самостоятельная работа.....	6
2.3. Форма обучения.....	6
2.4. Учебный план.....	6
<b>3. КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК .....</b>	<b>6</b>
<b>4. РАБОЧАЯ ПРОГРАММА ДПП .....</b>	<b>8</b>
<b>5. УСЛОВИЯ РЕАЛИЗАЦИИ ДПП.....</b>	<b>14</b>
5.1. Организационные условия реализации ДПП.....	14
5.2. Педагогические условия реализации ДПП.....	14
5.3. Учебно-методическое обеспечение ДПП.....	14
5.4. Методические рекомендации.....	15
<b>6. ФОРМЫ ИТОГОВОЙ АТТЕСТАЦИИ ДПП .....</b>	<b>17</b>
<b>7. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ИТОГОВОЙ АТТЕСТАЦИИ.....</b>	<b>18</b>
7.1. Паспорт комплекта оценочных средств.....	18
7.2. Комплект оценочных средств.....	18

## **1. ОБЩАЯ ХАРАКТЕРИСТИКА ДПП**

Программа подготовлена на основе:

- Федерального закона от 29 декабря 2012 года № 273-ФЗ «Об образовании в Российской Федерации»;
- требований Приказа Минобрнауки России от 01.07.2013 года № 499 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам»;
- методических рекомендаций-разъяснений Минобрнауки России по разработке дополнительных профессиональных программ на основе профессиональных стандартов от 22 апреля 2015 года № ВК-1030/06.

Реализация программы ДПП направлена на получение новой компетенции, необходимой для профессиональной деятельности.

### **1.1. Цель ДПП**

Сформировать у обучающихся знания, навыки и умения в области разработки, отладки, проверки работоспособности, модификации компьютерного программного обеспечения.

### **1.2. Планируемые результаты обучения**

Планируемые результаты обучения по ДПП:

- освоение профессиональных компетенций в процессе изучения перечисленных тем в учебном плане;
- успешное освоение программы повышения квалификации;
- успешное прохождение итоговой аттестации (зачет).

Обучающимся, успешно прошедшим обучение, выполнившим текущие контрольные задания и выдержавшим предусмотренное учебным планом зачет, выдается удостоверение о повышении квалификации по ДПП «Программирование на C++ (базовый уровень)».

### **1.3. Дополнительные характеристики ДПП**

Характеристики новой квалификации определены в приказе Минтруда России от 20.07.2022 №424н «Об утверждении профессионального стандарта «Программист».

Вид профессиональной деятельности:

- Разработка компьютерного программного обеспечения (Код 06.001).

Трудовые функции:

- Проектирование компьютерного программного обеспечения (D/03.6).

#### 1.4. Перечень профессиональных компетенций в рамках имеющейся квалификации, качественное изменение которых осуществляется в результате обучения

Получаемые компетенции базируются на основании Приказа Минобрнауки России от 19 сентября 2017 г. № 929 «Об утверждении федерального государственного образовательного стандарта высшего образования - бакалавриат по направлению подготовки 09.03.01 Информатика и вычислительная техника».

Перечень компетенций:

ОПК-8. Способен разрабатывать алгоритмы и программы, пригодные для практического применения.

#### 1.5. Соответствие видов деятельности профессиональным компетенциям и их составляющих

Профессиональные компетенции	Практический опыт	Умения	Знания
Проектирование компьютерного программного обеспечения (D/03.6)			
ОПК-8. Способен разрабатывать алгоритмы и программы, пригодные для практического применения	Проектирование структур данных; Проектирование баз данных; Проектирование программных интерфейсов	Применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов	Типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке компьютерного программного обеспечения

## 2. УЧЕБНЫЙ ПЛАН ДПП

### 2.1. Категория слушателей ДПП

Имеющаяся квалификация (требования к слушателям) – к освоению ДПП допускаются лица, имеющие среднее профессиональное и/или высшее образование.

### 2.2. Общая трудоёмкость программы, аудиторная и самостоятельная работа

Общая трудоёмкость программы 56 академических часов, из них 42 академических часа аудиторной работы, 12 академических часов самостоятельной работы и 2 академических часов итоговой аттестации.

### 2.3. Форма обучения

Форма обучения по ДПП – очная с применением дистанционных образовательных технологий.

### 2.4. Учебный план

ДПП «Программирование на C++ (базовый уровень)» реализуется одним модулем.

№ п/п	Наименование темы, модуля	Форма контроля	Всего, час	В том числе			
				Лекции	Практ. занятия	Самост. работа	Итоговая аттестация
1.	Вступление	Практ. задание	4	2	-	2	-
2.	Основные понятия	Практ. задание	10	4	4	2	-
3.	Функции и управление потоком	Практ. задание	10	4	4	2	-
4.	Указатели и ссылки	Практ. задание	10	4	4	2	-
5.	Классы	Практ. задание	10	4	4	2	-
6.	Наследование	Практ. задание	10	4	4	2	-
7.	Итоговая аттестация	Зачет	2	-	-	-	2
	ИТОГО	-	56	22	20	12	2



### 3. КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК

№ п/п	Наименование темы, модуля	1 день	2 день	3 день	4 день	5 день	6 день	7 день
1	Вступление	■						
2	Основные понятия	■	■					
3	Функции и управление потоком		■	■				
4	Указатели и ссылки				■	■		
5	Классы					■	■	
6	Наследование						■	■
7	Итоговая аттестация							Зачет

Минимальный срок освоения ДПП – 7 дней.

## 4. РАБОЧАЯ ПРОГРАММА ДПП

### 4.1. Рабочая программа модуля «Программирование на C++ (базовый уровень)»

4.1.1. Цель изучения модуля: сформировать у обучающихся знания, навыки и умения в области разработки, отладки, проверки работоспособности, модификации компьютерного программного обеспечения.

4.1.2. Задачи изучения модуля:

1. Знакомство с синтаксисом, семантикой и паттернами программирования на языке C++.
2. Получение практических навыков разработки консольных приложений на C++.
3. Получение теоретических и практических знаний по применению инструментов разработчика C++.

4.1.3. Планируемые результаты обучения

Процесс изучения раздела направлен на формирование следующих компетенций

Код компетенции	Перечень планируемых результатов обучения по модулю	Формы и методы обучения, способствующие формированию и развитию компетенции
ОПК-8	<b>Знать:</b> Типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке компьютерного программного обеспечения. <b>Уметь:</b> Применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов. <b>Владеть:</b> Проектирование структур данных; Проектирование баз данных; Проектирование программных интерфейсов.	Формы обучения: Фронтальная. Методы обучения: Лекция; Практическая работа; Самостоятельная работа.

4.1.4 Содержание курса

#### Тема 1. Вступление (4 часа)

Лекции (2 часа). Краткая история, предназначение и ниша языка C++. Демонстрация примера «Hello, world». Описание инструментов разработчика. Процесс сборки программ. Правила оформления кода. Где искать информацию.

Самостоятельная работа (2 часа). Проработка материала дополнительной литературы по теме.

Наименование темы	Дидактические единицы, вынесенные на самостоятельное изучение	Формы самостоятельной работы	Учебно-методическое обеспечение	Форма контроля
Вступление	Синтаксис, оформление кода	Проработка дополнительной литературы	Прата С. Язык программирования C++. Лекции и упражнения / Прата С.; пер. с англ. Корниенко Ю.И., Моргунов А.А.; ред. пер. Артеменко Ю.Н. – 6-е изд. – М.: Вильямс, 2018. – 1244 с.	Практ. задание

## Тема 2. Основные понятия (10 часов)

Лекции (4 часа). Переменные. Выражения. Декларация и инициализация. Точка входа. Аргументы функций и возврат значений. Директива препроцессора `#include`. Директива `using`. Пространства имён. Основы потоков ввода/вывода. Хранение данных в ОЗУ. Целочисленные переменные и литералы. Встроенные целочисленные типы. Целочисленные типы фиксированной длины. Инициализация переменных. Квалификатор `const`. Неявное приведение типов. Математические операторы. Операторы приведения типа `c-style cast`, `static_cast`. Оператор `sizeof()`. Псевдотип `auto`. Перечисления. Статические переменные и функции.

Практические занятия (4 часа). Ответы на вопросы. Разбор задания для самостоятельной работы.

Самостоятельная работа (2 часа). Проработка материала дополнительной литературы по теме.

Наименование темы	Дидактические единицы, вынесенные на самостоятельное изучение	Формы самостоятельной работы	Учебно-методическое обеспечение	Форма контроля
Основные понятия	Синтаксис, оформление кода	Проработка дополнительной литературы	Лафоре Р. Объектно-ориентированное программирование в C++. Классика Computer Science. 4-е изд. – (Серия «Классика computer science»). / Лафоре Р. – Санкт-Петербург: Питер, 2021. – 928 с.	Практ. задание

### Тема 3. Функции и управление потоком (10 часов)

Лекции (4 часа). Функции и процедуры. Разные способы указания возвращаемого типа. Значения аргументов по умолчанию. Перегрузка функций. Контейнер `std::string`. Интерактивное взаимодействие с программой. Операторы `if` и `switch`. Циклы `for`, `while` и `do-while`. Области видимости. Логические операторы и операторы сравнения. Тернарный оператор. Оператор `goto`. Рекурсия.

Практические занятия (4 часа). Ответы на вопросы. Разбор задания для самостоятельной работы.

Самостоятельная работа (2 часа). Проработка материала дополнительной литературы по теме.

Наименование темы	Дидактические единицы, вынесенные на самостоятельное изучение	Формы самостоятельной работы	Учебно-методическое обеспечение	Форма контроля
Функции и управление потоком	Функции	Проработка дополнительной литературы	Страуструп, Б. Дизайн и эволюция C++ / Б. Страуструп. – Москва: ДМК Пресс, 2007. – 448 с.	Практ. задание

### Тема 4. Указатели и ссылки (10 часов)

Лекции (4 часа). Массивы. Строковый тип данных. Указатели, типизированные и обобщённые. Нулевой указатель. Операции взятия адреса и разыменования. Арифметика указателей. Операторы `new` и `delete`, понятия стека и кучи. Ссылки. Оператор приведения типа `reinterpret_cast`. Контейнеры `unique_ptr` и `shared_ptr`. Указатели на функции. Оператор `decltype`. Передача функций как аргументов.

Практические занятия (4 часа). Ответы на вопросы. Разбор задания для самостоятельной работы.

Самостоятельная работа (2 часа). Проработка материала дополнительной литературы по теме.

Наименование темы	Дидактические единицы, вынесенные на самостоятельное изучение	Формы самостоятельной работы	Учебно-методическое обеспечение	Форма контроля
Указатели и ссылки	Ссылки	Проработка дополнительной литературы	Страуструп Б. Язык программирования С++ / Страуструп Б.; ред. пер. с англ. Мартынов Н. Н. – Спец. изд. – М.: Бином, 2011. – 1135 с.	Практ. задание

### Тема 5. Классы (10 часов)

Лекции (4 часа). Классы и структуры. Спецификаторы доступа, инкапсуляция. Время жизни объекта. Способы инициализации членов класса. Методы. Конструктор и деструктор. Указатель this. Оператор «стрелка».

Структуры хранения данных: стек, очередь, односвязный и двусвязный список, двоичное дерево, красно-чёрное дерево. Контейнеры `std::vector`, `std::array`, `std::list`, `std::set`, `std::map`.

Практические занятия (4 часа). Ответы на вопросы. Разбор задания для самостоятельной работы.

Самостоятельная работа (2 часа). Проработка материала дополнительной литературы по теме.

Наименование темы	Дидактические единицы, вынесенные на самостоятельное изучение	Формы самостоятельной работы	Учебно-методическое обеспечение	Форма контроля
Классы	Классы, структуры	Проработка дополнительной литературы	Р. Седжвик. Алгоритмы на С++ : курс лекций / Р. Седжвик. – Москва: Национальный Открытый Университет «ИНТУИТ», 2016. – 1773 с.	Практ. задание

### Тема 6. Наследование (10 часов)

Лекции (4 часа). Делегаты, события и лямбды. Как делегаты хранят ссылки на методы: через события / анонимные методы / лямбда-выражения / LINQ показана истинная сила делегатов.

Практические занятия (4 часа). Ответы на вопросы. Разбор задания для самостоятельной работы.

Самостоятельная работа (2 часа). Проработка материала дополнительной литературы по теме.

Наименование темы	Дидактические единицы, вынесенные на самостоятельное изучение	Формы самостоятельной работы	Учебно-методическое обеспечение	Форма контроля
Наследование	Делегаты, события, лямбды	Проработка дополнительной литературы	Мейерс, С. Наиболее эффективное использование C++. 35 новых рекомендаций по улучшению ваших программ и проектов: учебное пособие / С. Мейерс. – Москва: ДМК Пресс, 2007. – 294 с.	Практ. задание

4.1.5. Оценочное средство для текущего контроля (формулировка практических заданий):

*Тема 1. Формулировка практического задания:*

Настроить рабочее окружение, собрать тестовую программу. Инструментарий: WSL2/Linux, VS Code, gcc.

Критерии оценивания: наличие рабочего окружения, возможность сборки тестовой программы.

*Тема 2. Формулировка практического задания:*

Разработать программу для вывода таблицы основных типов данных C++.

Критерии оценивания: соответствие программы заданию; оформление кода.

*Тема 3. Формулировка практического задания:*

Разработать программу, которая в зависимости от аргументов командной строки будет либо возвращать размер введённого пользователем типа данных, либо выполнять преобразование входных данных в заданный тип. Предусмотреть защиту от неправильного ввода данных.

Критерии оценивания: соответствие программы заданию; оформление кода; наличие защиты от неправильного ввода данных.

*Тема 4. Формулировка практического задания:*

Разработать программы для:

1. Вычисления размера строки с использованием указателей.
  2. Замены значений двух переменных одного типа двумя способами: передачей переменных по указателю и по ссылке. Способ необходимо выбирать на основании ввода пользователя.
- Предусмотреть защиту от неправильного ввода данных.

Критерии оценивания: соответствие программы заданию; оформление кода; наличие защиты от неправильного ввода данных.

*Тема 5. Формулировка практического задания:*

Разработать программу, которая будет сохранять введённые пользователем в виде пар «ключ — значение» данные в контейнеры. Использовать разные типы контейнеров. Для последовательных контейнеров — сохранять классы. Для ассоциативных контейнеров — сохранять пары «ключ — значение». Предусмотреть вывод данных по запросу пользователя. Предусмотреть защиту от неправильного ввода данных.

Критерии оценивания: соответствие программы заданию; оформление кода; наличие защиты от неправильного ввода данных.

*Тема 6. Формулировка практического задания:*

Разработать программу для парсинга текстового файла, содержащего информацию о геометрических фигурах. Создать классы для каждой из фигур. Вывести сводную информацию: количество каждого типа фигур, общую площадь, общий периметр.

Критерии оценивания: соответствие программы заданию; оформление кода; наличие защиты от неправильного ввода данных.

## 5. УСЛОВИЯ РЕАЛИЗАЦИИ ДПП

### 5.1. Организационные условия реализации ДПП

Наименование аудитории	Вид занятия	Наименование оборудования, программного обеспечения
Аудитория для проведения лекций/практических занятий	Лекции	ПК с ОС GNU/Linux, либо Windows 10/11 + WSL, редактор VS Code, с доступом в Интернет и возможностью просмотра файлов в формате *.ppt, *.pptx, *.pdf, проектор/телевизор/монитор. Дополнительное ПО устанавливается по необходимости из репозитория Linux.
Аудитория для проведения лекций/практических занятий	Практические занятия	ПК с ОС GNU/Linux, либо Windows 10/11 + WSL, редактор VS Code, с доступом в Интернет и возможностью просмотра файлов в формате *.ppt, *.pptx, *.pdf, проектор/телевизор/монитор. Дополнительное ПО устанавливается по необходимости из репозитория Linux.
Коворкинги, учебные залы и т.д.	Самостоятельная работа	ПК с ОС GNU/Linux, либо Windows 10/11 + WSL, редактор VS Code, с доступом в Интернет и возможностью просмотра файлов в формате *.ppt, *.pptx, *.pdf, проектор/телевизор/монитор. Дополнительное ПО устанавливается по необходимости из репозитория Linux.
Аудитория для проведения лекций/практических занятий	Итоговая аттестация	ПК с ОС GNU/Linux, либо Windows 10/11 + WSL, редактор VS Code, с доступом в Интернет и возможностью просмотра файлов в формате *.ppt, *.pptx, *.pdf, *.docx, *.xlsx проектор/телевизор/монитор.

### 5.2. Педагогические условия реализации ДПП

Реализация программы обеспечивается преподавательским составом, удовлетворяющим следующим условиям:

- наличие высшего профессионального образования, соответствующее профилю программы, из числа штатных преподавателей, или привлеченных на условиях почасовой оплаты труда;
- значительный опыт практической деятельности в соответствующей сфере из числа штатных преподавателей или привлеченных на условиях почасовой оплаты труда

### 5.3. Учебно-методическое обеспечение ДПП

Основная литература:

1. Прата С. Язык программирования C++. Лекции и упражнения / Прата С.; пер. с англ. Корниенко Ю.И., Моргунов А.А.; ред. пер. Артеменко Ю.Н. – 6-е изд. – М.: Вильямс, 2018. – 1244 с.



2. Лафоре Р. Объектно-ориентированное программирование в С++. Классика Computer Science. 4-е изд. – (Серия «Классика computer science»). / Лафоре Р. – Санкт-Петербург: Питер, 2021. – 928 с.
3. Страуструп, Б. Дизайн и эволюция С++ / Б. Страуструп. – Москва: ДМК Пресс, 2007. – 448 с.
4. Страуструп Б. Язык программирования С++ / Страуструп Б.; ред. пер. с англ. Мартынов Н. Н. – Спец. изд. – М.: Бином, 2011. – 1135 с.
5. Мейерс, С. Наиболее эффективное использование С++. 35 новых рекомендаций по улучшению ваших программ и проектов: учебное пособие / С. Мейерс. – Москва: ДМК Пресс, 2007. – 294 с.

#### Дополнительная литература

1. Р. Седжвик. Алгоритмы на С++ : курс лекций / Р. Седжвик. – Москва: Национальный Открытый Университет «ИНТУИТ», 2016. – 1773 с.

#### Интернет-источники:

1. <https://habr.com/ru/companies/ncloudtech/articles/743930/>
2. <https://habr.com/ru/companies/otus/articles/762548/>
3. <https://habr.com/ru/companies/piter/articles/524882/>
4. <https://habr.com/ru/companies/pvs-studio/articles/715436/>
5. <https://en.cppreference.com/w/cpp>
6. <https://github.com/NekroIm/ubbook>

#### 5.4. Методические рекомендации

ДПП построена по тематическому принципу, каждый раздел представляет собой логически завершённый материал.

Преподавание программы основано на личностно-ориентированной технологии образования, сочетающей два равноправных аспекта этого процесса: обучение и учение. Личностно-ориентированный подход развивается при участии слушателей в активной работе на практических занятиях. Личностно-ориентированный подход направлен, в первую очередь, на развитие индивидуальных способностей обучающихся, создание условий для развития творческой активности слушателя и разработке инновационных идей, а также на развитие самостоятельности мышления при решении учебных задач разными способами, нахождение рационального варианта решения, сравнения и оценки нескольких вариантов их решения и т.п. Это способствует формированию приемов умственной деятельности по восприятию новой информации, ее запоминанию и

осознанию, созданию образов для сложных понятий и процессов, приобретению навыков поиска решений в условиях неопределенности.

Практические занятия проводятся для приобретения навыков решения практических задач в предметной области модуля. Задания, выполняемые на практических занятиях, выполняются с использованием активных и интерактивных методов обучения.

Самостоятельная работа слушателей предназначена для проработки дополнительной литературы. Результаты практических заданий слушателей учитываются на итоговой аттестации.

При изучении курса предусмотрены следующие методы организации и осуществления учебно-познавательной деятельности:

- объяснительно-иллюстративный метод;
- репродуктивный метод;
- частично-поисковый метод.

## 6. ФОРМЫ ИТОГОВОЙ АТТЕСТАЦИИ ДПП

Итоговая аттестация проводится в форме зачета для проверки сформированности компетенций, полученных в рамках ДПП.

Зачет проводится в формате тестирования. Результатом зачета служат правильные ответы на вопросы билета.

По результатам итоговой аттестации обучающемуся выставляется оценка «ЗАЧТЕНО/НЕ ЗАЧТЕНО»:

Оценка «ЗАЧТЕНО» выставляется обучающемуся, который:

- ответил на 21 из 35 вопросов;

Оценка «НЕ ЗАЧТЕНО» ставится обучающемуся, который:

- ответил менее чем на 21 из 35 вопросов теста.

## 7. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ИТОГОВОЙ АТТЕСТАЦИИ

### 7.1. Паспорт комплекта оценочных средств

Предметы оценивания	Объекты оценивания	Показатели оценки
ОПК-8. Способен разрабатывать алгоритмы и программы, пригодные для практического применения	Вопросы теста	Количество правильных ответов

### 7.2. Комплект оценочных средств

#### 7.2.1. Темы для подготовки к зачету:

1. Правила оформления кода.
2. Инструменты разработчика.
3. Процесс сборки программ.
4. Переменные и выражения.
5. Работа с пространствами имён.
6. Работа с памятью.
7. Работа с функциями.
8. Перегрузка функций.
9. Рекурсивные функции.
10. Массивы.
11. Работа с ссылками и указателями.
12. Строковые типы данных.
13. Структуры и классы.
14. Методы классов.
15. Структуры хранения данных.
16. Перегрузка операторов.
17. Наследование и полиморфизм.
18. Работа с файловой системой.

#### 7.2.2. Вопросы теста для проведения зачета:

1. Какой размер имеет тип long int ?
  - А. 2 байта
  - Б. 4 байта
  - В. 8 байтов

Г. зависит от архитектуры и платформы

2. Какому типу данных соответствует литерал 0x1000UL ?

А. int

Б. unsigned int

В. long unsigned int

Г. long long int

3. Какой тип данных возвращает функция со следующим прототипом:

```
auto addStudent(Student& stud, int age) -> decltype(stud)
```

А. auto

Б. Student

В. Student&

Г. int

4. Чему будет равна переменная b ?

```
unsigned char a = 250;
```

```
unsigned char b = a + 10;
```

А. 4

Б. 5

В. 260

Г. код содержит ошибку (неопределённое поведение)

5. Сколько раз выполнится цикл?

```
unsigned int a = 0, b = 1;
```

```
do {
```

```
    b++;
```

```
} while (a);
```

А. ни одного раза

Б. один раз

В. будет работать бесконечно

Г. код содержит ошибку компиляции

6. Чем является переменная `str` ?

```
const char* str;
```

- А. указателем на строку
- Б. изменяемым указателем на неизменяемую строку
- В. неизменяемым указателем на изменяемую строку
- Г. неизменяемым указателем на неизменяемую строку

7. Для преобразования типа `long int` в тип `short int` необходимо использовать?

- А. `const_cast<short int>`
- Б. `static_cast<short int>`
- В. `reinterpret_cast<short int>`
- Г. `dynamic_cast<short int>`

8. Выберите верное утверждение о ссылках и указателях

- А. ссылка и указатель — это одно и то же
- Б. указатель необходимо удалять после использования, а ссылку — нет
- В. указатель может быть нулевым, а ссылка — нет
- Г. ссылка гарантирует отсутствие неопределённого поведения, а указатель — нет

9. Двусвязный список это ...

- А. структура данных, содержащая указатель на предыдущий элемент
- Б. структура данных, содержащая указатель на следующий элемент
- В. структура данных, содержащая указатель на первый элемент
- Г. структура данных, содержащая указатель на предыдущий и следующий элементы

10. Оператор `new[]` нужен для

- А. выделения памяти для объекта на стеке
- Б. выделения памяти для объекта в куче
- В. выделения памяти для массива объектов в куче
- Г. выделения памяти для стандартных контейнеров

11. Какая из перегрузок функции `int foo(int val)` является ошибочной?

- А. `int foo(float val)`
- Б. `int foo(int val, int& out)`
- В. `int foo()`

Г. `int foo(int val, int flag = 0)`

12. Какую структуру данных реализует контейнер `std::list` ?

- А. односвязный список
- Б. двусвязный список
- В. вектор
- Г. двоичное дерево

13. Кто не имеет доступа к полям класса, скрытым спецификатором `protected` ?

- А. наследник со спецификатором `public` ( `class A : public B { }` )
- Б. наследник со спецификатором `protected` ( `class A : protected B { }` )
- В. класс, владеющий указанным ( `class A { class B ; }` )
- Г. функции-друзья

14. Какой оператор не может быть перегружен?

- А. Инкремент и декремент ( `a++` , `a--` )
- Б. Оператор стрелка ( `a->b` )
- В. Оператор доступа к члену структуры или класса ( `a.b` )
- Г. Оператор вызова функции, функтор ( `a()` )

15. Может ли метод класса, объявленный со спецификатором `const` , изменять поля класса?

- А. не может ни в каком случае
- Б. может, если поле имеет квалификатор `mutable`
- В. может, если поле имеет спецификатор `static`
- Г. может, если метод не возвращает никакого значения

16. Выберите верное утверждение о структурах

- А. понятие структуры полностью идентично классу
- Б. структуры способны хранить только встроенные типы данных ( `int` , `float` и т. д.)
- В. структуры по умолчанию имеют спецификатор доступа `public` , а классы — `private`
- Г. структуры работают как классы, но не поддерживают наследование

17. Когда вызовется деструктор класса, созданного в функции foo() ?

- А. после выхода из функции foo()
- Б. после выхода из функции main()
- В. после вызова оператора delete
- Г. деструктор необходимо вызвать вручную

18. Дано перечисление `enum class A : uint32_t { VAL = 10 }`. Выберите фрагмент кода, не содержащий синтаксических ошибок.

- А. `uint32_t a = VAL`
- Б. `uint32_t a = A::VAL`
- В. `auto a = static_cast<uint32_t>(VAL)`
- Г. `auto a = static_cast<uint32_t>(A::VAL)`

19. Чему равно значение переменной a ? `uint32_t* a = 0x1000; a += 2;`

- А. 0x1002
- Б. 0x1004
- В. 0x1008
- Г. 0x100A

20. Какая проверка ptr на нулевой указатель не является правильной?

- А. `if (ptr == nullptr)`
- Б. `if (!ptr)`
- В. `if (ptr == NULLPTR)`
- Г. `if (ptr == NULL)`

21. Что напечатает следующий код, если `int val = 1` ?

```
switch(val) {  
    case 1: std::cout << "1" << std::endl;  
    case 2: std::cout << "2" << std::endl;  
    case 3: std::cout << "3" << std::endl;  
    default: std::cout << "4" << std::endl;  
}
```

- А. 1
- Б. 234
- В. 123



Г. 1234

22. Какая из строк кода содержит ошибку в использовании типа void ?

- A. int foo(void);
- Б. void a;
- В. void\* a;
- Г. void foo(int a);

23. Для немедленного перехода к следующей итерации любого цикла используется оператор

- A. break
- Б. continue
- В. return
- Г. goto

24. Какой прототип должна иметь функция, вызываемая следующим образом:

```
int* ptr = nullptr;  
foo(&ptr);
```

- A. void foo(int ptr)
- Б. void foo(int\* ptr)
- В. void foo(int\*\* ptr)
- Г. void foo(int& ptr)

25. Чем будет равна переменная c ?

```
int a = 5, b = 10;  
int c = (a > 4) ? ((b < 10) ? 100 : 200) : 300;
```

- A. 100
- Б. 200
- В. 300
- Г. ошибка компиляции

26. Какое из объявлений цикла range-based for для массива `int a[10]`; приведёт к ошибке компиляции?

- А. `for(int val : a)`
- Б. `for(auto val : a)`
- В. `for(auto& val : a)`
- Г. `for(auto* val : a)`

27. Чему будут равны значения массива `int a[3] = { 1 } ?`

- А. `a[0] == a[1] == a[2] == 1`
- Б. `a[0] == 1`, `a[1]` и `a[2]` содержат мусор
- В. `a[0] == 1`, `a[1] == a[2] == 0`
- Г. зависит от контекста

28. Чему равно значение `arrSize` ?

```
uint32_t arr[5] = {};  
uint32_t* ptr = arr;  
arrSize = sizeof(ptr);
```

- А. заданному размеру массива (5)
- Б. сумме размеров всех элементов массива ( `sizeof(arr)` )
- В. размеру типа массива ( `sizeof(uint32_t)` )
- Г. размеру обобщённого указателя ( `sizeof(void*)` )

29. Какая операция допустима в C++?

- А. многократное изменение одной переменной в пределах одной точки следования
- Б. разыменование нулевого указателя
- В. переполнение беззнаковой переменной
- Г. попытка освободить уже освобождённую память (оператором `delete`)

30. Каким способом можно вернуть из функции больше одного значения?

- А. возвращать класс или структуру, содержащие нужные значения
- Б. передать в функцию изменяемую ссылку на значение, а другое значение вернуть оператором `return`
- В. передать в функцию указатель на изменяемое значение, а другое значение вернуть оператором `return`
- Г. всё вышеперечисленное

31. Может ли деструктор быть перегружен? Объясните ответ (вопрос с открытым ответом).

32. Зачем нужен спецификатор override (вопрос с открытым ответом)?

33. Что напечатает данный код (вопрос с открытым ответом)?

```
#include <iostream>
class A {
public:
    A() = default;
    void print() { std::cout << "1" << std::endl; }
};
class B : public A {
public:
    B() = default;
    void print() { std::cout << "2" << std::endl; }
};
int main() {
    A classA;
    classA.print();
    B classB;
    classB.print();
    A* pointerToB = &classB;
    pointerToB->print();
    return 0;
}
```

34. Что напечатает данный код (вопрос с открытым ответом)?

```
#include <iostream>
bool foo() {
    std::cout << "foo" << std::endl;
    return true;
}
bool bar() {
```

```
std::cout << ""bar"" << std::endl;
return false;
}
bool baz() {
std::cout << ""baz"" << std::endl;
return false;
}
int main() {
if ((foo() || bar()) && baz())
std::cout << ""main"" << std::endl;
return 0;
}
```

35, Что напечатает данный код (вопрос с открытым ответом)?

```
#include <iostream>
void rec(int n) {
if(n > 0)
rec(n - 1);
std::cout << n << std::endl;
}
int main(void) {
rec(4);
return 0; }
```