

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**  
**ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ**



УТВЕРЖДАЮ  
Декан РТФ

УТВЕРЖДАЮ /А.Н. Дедов/  
(Ф.И.О. декана (директора института))

30.06.2021 г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)**

Б.1.1.23 Язык программирования Си++

*(код и наименование дисциплины по учебному плану)*

Направление подготовки  
(специальность)

11.03.03 Конструирование и технология электронных средств

Квалификация выпускника

Бакалавр

(бакалавр/магистр/специалист)

Направленность

Проектирование и технология электронно-  
вычислительных средств

Курс 2, 3

Семестр 4, 5

**Распределение учебного времени**

Трудоемкость по учебному плану	<u>180 / 5</u>	часов/зачетных единиц
Лекции	<u>4</u>	часов
Лабораторные работы	<u>8</u>	часов
Практические занятия	<u>8</u>	часов
Иная контактная работа	<u>-</u>	часов
Всего контактной работы (без учета экз.)	<u>20</u>	часов
Контактная работа по экзамену	<u>6</u>	часов
Курсовой проект (работа)	<u>-</u>	семестр
Самостоятельная работа обучающихся (без учета экз.)	<u>124</u>	часов
Самостоятельная работа по подготовке к экзамену	<u>30</u>	часов
Экзамен	<u>5</u>	семестр
Зачет	<u>-</u>	семестр
БРК, ДЗ	<u>-</u>	семестр

\_\_\_\_\_  
(год)

Программа составлена в соответствии с требованиями ФГОС ВО направления подготовки (специальности) 11.03.03 Конструирование и технология электронных средств

Программу составили:

доцент, кандидат наук	ПиП ЭВС	СОГЛАСОВАНО	Д.О. Глухов
(должность)	(кафедра)		(И.О. Фамилия)

РАССМОТРЕНА и ОДОБРЕНА на заседании кафедры, за которой закреплена дисциплина  
Кафедра проектирования и производства электронно-вычислительных средств

(наименование кафедры)		
17.05.2021	протокол №	19
(дата)		

Заведующий кафедрой	СОГЛАСОВАНО	Т.С. Буканова
		(И.О. Фамилия)

Рабочая программа СОГЛАСОВАНА с факультетом (институтом), выпускающей(ими) кафедрой(ами).

СООТВЕТСТВУЕТ действующей ОП.

Заведующий кафедрой	СОГЛАСОВАНО	Т.С. Буканова
		(И.О. Фамилия)

Председатель методической комиссии факультета (института), в который входит выпускающая кафедра

СОГЛАСОВАНО	А.Н. Дедов
	(И.О. Фамилия)

Эксперт(ы): Семенов Владимир Дмитриевич, заместитель директора ООО "Технотех"

Рабочая программа проверена и зарегистрирована в УМЦ 01.07.2021 г.

Специалист учебно-методического центра СОГЛАСОВАНО /Т.А. Смирнова/

## Раздел 1. ЦЕЛЬ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Целью освоения дисциплины является достижение планируемых результатов обучения, соответствующих установленным в ОПОП индикаторам достижения компетенций:

Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Результаты обучения
1. ОПК-4 Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности	ОПК-4.1 Использует информационно-коммуникационные технологии для подготовки документации	<b>знания:</b> Знания принципов работы современных информационных технологий <b>умения:</b> Умеет разрабатывать технологическую документацию <b>навыки:</b> Владеет навыками решения задач профессиональной деятельности
	ОПК-4.2 Проектирует решение конкретной задачи проекта, выбирая оптимальный способ ее решения, исходя из действующих правовых норм и имеющихся ресурсов и	<b>знания:</b> Знает оптимальные способы решения задач <b>умения:</b> Умеет решать задачи профессиональной деятельности <b>навыки:</b> Владеет навыками решения задач профессиональной деятельности
	ОПК-4.6 Использует современные информационные технологии для проектирования и разработки программного обеспечения с целью решения задач профессиональной деятельности	<b>знания:</b> Знает современные информационные технологии для проектирования и разработки программного обеспечения с целью решения задач профессиональной деятельности <b>умения:</b> Использует современные информационные технологии для проектирования и разработки программного обеспечения с целью решения задач профессиональной деятельности <b>навыки:</b> Владеет навыками решения задач профессиональной деятельности
2. ОПК-5 Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	ОПК-5.4 Разрабатывает прикладные программы на языке высокого уровня для электронно-вычислительных средств, пригодные для решения практических задач в профессиональной деятельности	<b>знания:</b> Знает прикладные программы решения практических задач для электронно-вычислительных средств, пригодные для задач в профессиональной деятельности <b>умения:</b> Умеет разрабатывать прикладные программы решения практических задач для электронно-вычислительных средств, пригодные для задач в профессиональной деятельности <b>навыки:</b> Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения в составе электронно-вычислительных средств

ОПК-5.1 Знает методы алгоритмизации, языки и технологии программирования, пригодные для практического применения в области информационных систем и технологий	<b>знания:</b> Знает методы алгоритмизации, языки и технологии программирования, пригодные для практического применения в области информационных систем и технологий <b>умения:</b> <b>навыки:</b>
ОПК-5.2 Умеет применять методы алгоритмизации, языки и технологии программирования при решении профессиональных задач в области информационных систем и технологий	<b>знания:</b> <b>умения:</b> Умеет применять методы алгоритмизации, языки и технологии программирования при решении профессиональных задач в области информационных систем и технологий <b>навыки:</b>
ОПК-5.3 Владеет навыками программирования, отладки и тестирования прототипов программно-технических	<b>знания:</b> <b>умения:</b> <b>навыки:</b> Владеет навыками программирования, отладки и тестирования прототипов программно-технических комплексов задач

## Раздел 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина относится к обязательной части ОПОП.

Дисциплина является обязательной

Для продолжения формирования заявленных компетенций необходимы знания предшествующих дисциплин: Информационные технологии (ОПК-4); практик: Учебная практика (ознакомительная) (ОПК-4), Учебная практика (ознакомительная) (ОПК-5)

Изучаемая дисциплина является основой для продолжения формирования указанных компетенций в следующих дисциплинах: Микропроцессорные устройства (ОПК-5), Вычислительные машины, системы и сети (ОПК-5); практиках: Преддипломная практика (ОПК-4), Преддипломная практика (ОПК-5); государственной итоговой аттестации в форме: Подготовка к сдаче и сдача государственного экзамена (ОПК-4), Выполнение и защита выпускной квалификационной работы (ОПК-4), Выполнение и защита выпускной квалификационной работы (ОПК-5)

## Раздел 3. ОПИСАНИЕ ОБРАЗОВАТЕЛЬНЫХ ТЕХНОЛОГИЙ

Для формирования заявленных компетенций используются методологические технологии, реализующие деятельностный, личностно-ориентированный, практико-ориентированный подходы.

Основными стратегическими технологиями являются: лекционные занятия, практические занятия

На достижение конкретных целей обучения направлены применяемые тактические технологии: классическая лекция

## Раздел 4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

### 4 семестр

Виды и темы занятий	Количество часов	Формируемые компетенции
<b>5 семестр</b>	<b>72</b>	ОПК-4, ОПК-5
Лекция. Введение. Языки программирования. Основные понятия и определения.	2	
Практическое занятие. Концепции чистого кода	2	
Лабораторная работа. Работа с функциями в концепции чистого кода	2	
Задания для самостоятельной работы, в том числе выполнение Изучение лекционного материала. Изучение дополнительной литературы. Оформление лабораторных и практических отчетов. Подготовка к защите лабораторных работ. Подготовка к промежуточным аттестациям и контрольным работам. Подготовка докладов.	66	
Иная контактная работа:	0	

### 5 семестр

Виды и темы занятий	Количество часов	Формируемые компетенции
<b>6 семестр</b>	<b>72</b>	ОПК-4, ОПК-5
Лекция. Объектно-ориентированное программирование и процедурное программирование - отличия и преимущества	2	
Практическое занятие. Шаблоны и структуры в C++	2	
Практическое занятие. Многопоточное программирование в C++	2	
Практическое занятие. Абстрактные классы в C++	2	
Лабораторная работа. Объектно-ориентированное программирование на C++ с использованием классов	2	
Лабораторная работа. Методы и интерфейсы в C++	2	
Лабораторная работа. Абстрактные классы в C++	2	
Задания для самостоятельной работы, в том числе выполнение Изучение лекционного материала. Изучение дополнительной литературы. Оформление лабораторных и практических отчетов. Подготовка к защите лабораторных работ. Подготовка к промежуточным аттестациям и контрольным работам. Подготовка докладов.	58	
Иная контактная работа: консультации	0	
Подготовка к экзамену	30	
Проведение экзамена	6	

## Раздел 5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Изучение дисциплины рекомендуется начать с ознакомления с рабочей программой, ее структурой и содержанием разделов. Учебный материал структурирован, изучение дисциплины осуществляется в тематической последовательности.

**Занятия лекционного типа** дают систематизированные знания по дисциплине, концентрируют внимание на наиболее сложных и важных вопросах. Во время лекционных занятий рекомендуется вести конспектирование учебного материала; обращать внимание на формулировки и категории, раскрывающие суть проблемы, явления или процесса; зафиксировать выводы и практические рекомендации. Подготовка к занятиям **семинарского типа** включает ознакомление с планом **практического и лабораторного** занятия; работу с конспектом лекций, выполнение домашнего задания, работу с учебной и учебно-методической литературой, научными изданиями и электронными образовательными ресурсами, рекомендованными рабочей программой дисциплины.

Содержание **самостоятельной работы** определяется рабочей программой дисциплины, оценочными и методическими материалами, заданиями и указаниями преподавателя. Самостоятельная работа может осуществляться в аудиторной и внеаудиторной формах. Эффективным средством осуществления самостоятельной работы является электронная информационно-образовательная среда университета, которая обеспечивает доступ к образовательной программе, рабочей программе дисциплины, к электронным библиотечным системам, профессиональным базам данных и информационным справочным системам. Изучение дисциплины включает выполнение **лабораторной работы, практической работы, написание доклада.**

Периодичность проведения, формы текущего контроля успеваемости, система оценивания хода освоения дисциплин представлены в рабочей программе. Формой промежуточной аттестации по дисциплине является **экзамен.**

## Раздел 6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

### 6.1. Учебно-методическое обеспечение

№№ п/п	Список используемой литературы	Количество экземпляров печатных изданий, имеющих в библиотеке, или электронный адрес издания (ресурса) в сети Интернет
УЧЕБНЫЕ, УЧЕБНО-МЕТОДИЧЕСКИЕ И НАУЧНЫЕ ИЗДАНИЯ		
1.	Барков, И. А. Объектно-ориентированное программирование [Электронный ресурс] : учебник / Барков И. А. Санкт-Петербург: Лань, 2023. - 700 с. ISBN 978-5-8114-3586-9.	<a href="https://e.lanbook.com/book/329549">https://e.lanbook.com/book/329549</a>
2.	Конова, Е. А. Алгоритмы и программы. Язык C++ [Электронный ресурс] : учебное пособие для вузов / Конова Е. А., Поллак Г. А.; Конова Е. А. 7-е изд., стер. Санкт-Петербург: Лань, 2023. - 384 с. ISBN 978-5-507-46070-0.	<a href="https://e.lanbook.com/book/297002">https://e.lanbook.com/book/297002</a>
3.	Иванов, Б. Н. Дискретная математика. Алгоритмы и программы. Расширенный курс [Электронный ресурс] : учебное пособие для вузов / Иванов Б. Н. Санкт-Петербург: Лань, 2023. - 668 с. ISBN 978-5-507-45685-7.	<a href="https://e.lanbook.com/book/356132">https://e.lanbook.com/book/356132</a>
ЭЛЕКТРОННЫЕ ОБРАЗОВАТЕЛЬНЫЕ РЕСУРСЫ		

1.	Научная электронная библиотека eLIBRARY.RU	<a href="http://elibrary.ru">http://elibrary.ru</a>
2.	Научная электронная библиотека «Киберленинка»	<a href="http://cyberleninka.ru">http://cyberleninka.ru</a>

## 6.2. Материально-техническая база и программное обеспечение

№№ п/п	Аудитории для проведения учебных занятий, самостоятельной работы и проведения государственной итоговой аттестации	Перечень основного оборудования	Программное обеспечение
1.	502 (III)	Каркас КИП 1000*500*400 с подставкой (1), Персональный компьютер в сборе PowerCool(Core i3-8100/H310/16GbDDR4/HDD 0.5Tb/23"6 АОС/кл.мышь/пач-корд 3м) (12), Принтер HP Laser Jet 1020 (1), Проектор мультимедийный Sanuo PLC-XD2600 (1), Стенд измерит. параметров потоков многофазных сред (1), Стенд лабораторный "ПЛИС" (1), Экран настенный рулонный 200х200 см (1), Комплект учебной мебели (1)	Microsoft Windows Enterprise, Справочная правовая система "Консультант Плюс", Microsoft Office Standard, Агент Dr.Web, Комплект ГАРАНТ-Мастер, Microsoft Access, Microsoft Visio Professional, Microsoft Project Professional, Microsoft Visual Studio Enterprise, Комплект ПО для решения основных пользовательских задач, Mathcad University Classroom Perpetual - 40, Платформа nanoCAD, Altium Designer Perpetual EDU v15, LABVIEW, КОМПАС-3D V19, nanoCAD Инженерный BIM

## Раздел 7. ФОРМЫ КОНТРОЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ/ ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

Критерии оценивания индикаторов достижения компетенций направлены на:

- усвоение теоретического материала (объем знаний, глубина усвоения), предусмотренного рабочей программой;
- умение излагать материал (четкость, грамотность изложения материала, точность и полнота воспроизведения учебного материала);
- умение применять теоретические знания при решении практических заданий.

Шкала оценивания представлена ниже.

Уровень сформированности элементов компетенции	Критерии оценивания	Шкала оценивания
Пороговый уровень	Обучающийся имеет знания основного материала, проявляет умение логично его излагать, но может допускать неточности в изложении материала, недостаточно правильные формулировки, испытывает	удовлетворительно

	затруднения в выполнении практических заданий.	
Продвинутый уровень	Обучающийся твердо знает программный материал, излагает его грамотно и по существу, не допускает существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении практических вопросов и задач, владеет необходимыми навыками и приемами их выполнения	хорошо
Высокий уровень	Обучающийся глубоко и прочно усвоил программный материал, грамотно и логически стройно его излагает, дает исчерпывающие ответы на поставленные вопросы. В ответе тесно увязывается теория с практикой, при этом обучающийся не затрудняется с ответом при видоизменении задания, свободно справляется с задачами, вопросами и другими видами применения знаний, показывает знакомство с монографической литературой, периодическими изданиями, правильно обосновывает принятые решения, свободно владеет разносторонними навыками, приемами выполнения практических работ	отлично

### 7.1. Промежуточная аттестация обучающихся

Промежуточная аттестация обучающихся направлена на оценивание результатов обучения по дисциплине (модулю) и проводится с использованием фондов оценочных средств.

Примеры типовых контрольных заданий из базы фонда оценочных средств по образовательной программе.

1. Класс - это:

- любой тип данных, определяемый пользователем
- \* тип данных, определяемый пользователем и сочетающий в себе данные и функции их обработки
- структура, для которой в программе имеются функции работы с нею

5. Тест. Членами класса могут быть

- \* как переменные, так и функции, могут быть объявлены как `private` и как `public`
- только переменные, объявленные как `private`
- только функции, объявленные как `private`
- только переменные и функции, объявленные как `private`
- только переменные и функции, объявленные как `public`

6. Что называется конструктором?

- \* метод, имя которого совпадает с именем класса и который вызывается автоматически при создании объекта класса
- метод, имя которого совпадает с именем класса и который вызывается автоматически при объявлении класса (до создания объекта класса)
- метод, имя которого необязательно совпадает с именем класса и который вызывается при



создании объекта класса

- метод, имя которого совпадает с именем класса и который необходимо явно вызывать из головной программы при объявлении объекта класса

4. Объект - это

- переменная, содержащая указатель на класс
- \* экземпляр класса
- класс, который содержит в себе данные и методы их обработки

8. Отметьте правильные утверждения

- \* конструкторы класса не наследуются
- конструкторов класса может быть несколько, их синтаксис определяется программистом
- \* конструкторов класса может быть несколько, но их синтаксис должен подчиняться правилам перегрузки функций
- конструктор возвращает указатель на объект
- \* конструктор не возвращает значение

9. Что называется деструктором?

- метод, который уничтожает объект
- метод, который удаляет объект
- \* метод, который освобождает память, занимаемую объектом
- системная функция, которая освобождает память, занимаемую объектом

7. Выберите правильные утверждения

- \* у конструктора могут быть параметры
- конструктор наследуется, но должен быть перегружен
- конструктор должен явно вызываться всегда перед объявлением объекта
- \* конструктор вызывается автоматически при объявлении объекта
- объявление каждого класса должно содержать свой конструктор
- \* если конструктор не создан, компилятор создаст его автоматически

10. Выберите правильные утверждения

- деструктор - это метод класса, применяемый для удаления объекта
- \* деструктор - это метод класса, применяемый для освобождения памяти, занимаемой объектом
- деструктор - это отдельная функция головной программы, применяемая для освобождения памяти, занимаемой объектом
- \* деструктор не наследуется

- деструктор наследуется, но должен быть перегружен

Тест - 11. Что называется наследованием?

- \* это механизм, посредством которого производный класс получает элементы родительского и может дополнять либо изменять их свойства и методы
- это механизм переопределения методов базового класса
- это механизм, посредством которого производный класс получает все поля базового класса
- это механизм, посредством которого производный класс получает элементы родительского, может их дополнить, но не может переопределить

12. Выберите правильное объявление производного класса

- class MoreDetails:: Details;
- class MoreDetails: public class Details;
- \* class MoreDetails: public Details;
- class MoreDetails: class(Details);

13. Выберите правильные утверждения:

- если элементы класса объявлены как private, то они доступны только наследникам класса, но не внешним функциям
- \* если элементы класса объявлены как private, то они недоступны ни наследникам класса, ни внешним функциям
- если элементы объявлены как public, то они доступны наследникам класса, но не внешним функциям
- \* если элементы объявлены как public, то они доступны и наследникам класса, и внешним функциям

14. Возможность и способ обращения производного класса к элементам базового определяется

- ключами доступа: private, public, protected в теле производного класса
- только ключом доступа protected в заголовке объявления производного класса
- \* ключами доступа: private, public, protected в заголовке объявления производного класса
- ключами доступа: private, public, protected в теле базового класса

15. Выберите правильные соответствия между спецификатором базового класса, ключом доступа в объявлении производного класса и правами доступа производного класса к элементам базового

- ключ доступа - public; в базовом классе: private; права доступа в производном классе - protected
- \* ключ доступа - любой; в базовом классе: private; права доступа в производном классе - нет прав
- \* ключ доступа - protected или public ; в базовом классе: protected; права доступа в

производном классе - protected

- ключ доступа - private; в базовом классе: public; права доступа в производном классе - public

- \* ключ доступа – любой; в базовом классе: public; права доступа в производном классе – такие же, как ключ доступа

16. Дружественная функция - это

- функция другого класса, среди аргументов которой есть элементы данного класса

- \* функция, объявленная в классе с атрибутом friend, но не являющаяся членом класса;

- функция, являющаяся членом класса и объявленная с атрибутом friend;

- функция, которая в другом классе объявлена как дружественная данному

17. Выберите правильные утверждения:

- \* одна функция может быть дружественной нескольким классам

- дружественная функция не может быть обычной функцией, а только методом другого класса

- \* дружественная функция объявляется внутри класса, к элементам которого ей нужен доступ

- дружественная функция не может быть методом другого класса

18. Шаблон функции - это...

- \* определение функции, в которой типу обрабатываемых данных присвоено условное обозначение

- прототип функции, в котором вместо имен параметров указан условный тип

- определение функции, в котором указаны возможные варианты типов обрабатываемых параметров

- определение функции, в котором в прототипе указан условный тип, а в определении указаны варианты типов обрабатываемых параметров

2. Выберите правильные утверждения:

- \* по умолчанию члены класса имеют атрибут private

- по умолчанию члены класса имеют атрибут public;

- члены класса имеют доступ только к элементам public;

- \* элементы класса с атрибутом private доступны только членам класса

3. Переопределение операций имеет вид:

- имя\_класса, ключевое слово operation, символ операции

- \* имя\_класса, ключевое слово operator, символ операции, в круглых скобках могут быть указаны аргументы

- имя\_класса, ключевое слово operator, список аргументов

- имя\_класса, два двоеточия, ключевое слово operator, символ операции

Тест - 19. Для доступа к элементам объекта используются:

- \* при обращении через имя объекта – точка, при обращении через указатель – операция «->»
- при обращении через имя объекта – два двоеточия, при обращении через указатель – операция «точка»
- при обращении через имя объекта – точка, при обращении через указатель – два двоеточия
- при обращении через имя объекта – два двоеточия, при обращении через указатель – операция «->»

20. Полиморфизм – это :

- \* средство, позволяющее использовать одно имя для обозначения действий, общих для родственных классов
- средство, позволяющее в одном классе использовать методы с одинаковыми именами;
- средство, позволяющее в одном классе использовать методы с разными именами для выполнения одинаковых действий
- средство, позволяющее перегружать функции для работы с разными типами или разным количеством аргументов.

21. Полиморфизм реализован через механизмы:

- \* перегрузки функций, виртуальных функций, шаблонов
- перегрузки функций, наследования методов, шаблонов;
- наследования методов, виртуальных функций, шаблонов
- перегрузки функций, наследования, виртуальных функций.

22. Виртуальными называются функции:

- \* функции базового класса, которые могут быть переопределены в производном классе
- функции базового класса, которые не используются в производном классе;
- функции базового класса, которые не могут быть переопределены в базовом классе;
- функции производного класса, переопределенные относительно базового класса

23. Выберите правильный вариант выделения динамической памяти под переменную X типа float:

- \* float \*ptr = new float; X = \*ptr;
- float & ptr = new float; X = & ptr;
- float \* ptr = &X; X = new float;

24. Полиморфизм в объектно-ориентированном программировании реализуется:

- \* через механизмы перегрузки (функций и операций), виртуальные функции и шаблоны

- через механизмы перегрузки (функций и операций) и шаблоны;
- через виртуальные функции и шаблоны;
- через механизмы перегрузки (функций и операций) и виртуальные функции

25. Дано определение класса

```
class monstr {
    int health, armo;

    monstr(int he, int arm);

public:
    monstr(int he=50, int arm=10);
    int color;
}
```

26.Тест. Укажите свойства и методы, доступные внешним функциям

- health, armo  
monstr(int he, int arm);  
monstr(int he=50, int arm=10);
- \* int color;  
monstr(int he=50, int arm=10);
- health, armo, color  
monstr(int he=50, int arm=10);
- int color;  
monstr(int he, int arm);

Перечень вопросов для проведения промежуточной аттестации

1. Спецификация const для данных. Назначение.2. Спецификация inline для функций. Назначение.3. Модульность в C++. Что пишется в файле .h, что в файле .cpp? Как использовать модуль, разбитый на эти 2 файла, в своих программах?4. Ссылки. Назначение, обращение к данным по ссылке, использование ссылок для параметров функции и возвращаемого значения.5. Динамическое создание и уничтожение объектов. Можно ли операции new и delete использовать вместе с malloc и free?6. Динамическое создание и уничтожение массивов объектов.7. Определение классов (синтаксис без наследования) в C++. Спецификаторы доступа.8. Использование операции :: для доступа к элементам класса и глобальным функциям и переменным.9. Спецификации const для методов, не изменяющих объект. Спецификация mutable для элементов данных.10. Дружественные функции и классы.11. Статические переменные класса. Определение и инициализация.12. Статические методы класса.13. Указатель this. Назначение. 14. Конструктор: назначение, объявление (синтаксис), момент выполнения.15. Конструктор: инициализация базовых классов и данных объекта через список инициализации в конструкторе.16. Конструктор по умолчанию: объявление (синтаксис), назначение.17. Конструктор копий: объявление (синтаксис), назначение.18. Деструктор: назначение, объявление (синтаксис), момент выполнения.19. Конструкторы-преобразователи и операции для преобразования, способ вызова. Спецификация explicit для конструкторов-преобразователей.20. Шаблоны функций: определение (синтаксис), назначение. Как вызвать функцию-шаблон?21. Шаблоны классов (параметризация): определение (синтаксис), назначение. Как создать объект шаблонного класса?22. Специализация шаблонов.23. Библиотека STL. Общая характеристика.24. Перегрузка функций и методов, основные правила связывания.25. Правила перегрузки операций. 26. Формат перегрузки унарных и бинарных операций как методов и [дружественных] функций.27. Перегрузка операций << и >> для ввода-вывода: назначение, объявление (синтаксис).28. Перегрузка операций присваивания: назначение, объявление (синтаксис),

действия выполняемые в методе.29. Перегрузка операций [] и () : назначение, объявление (синтаксис).30. Перегрузка операций \* и -> : назначение, объявление (синтаксис).31. Перегрузка операций new и delete в классе: назначение, объявление (синтаксис).32. Исключительные ситуации: назначение и стандартные искл. ситуации (кроме STL).33. Исключительные ситуации: порождение и перехват (синтаксис).34. Исключительные ситуации: спецификация порождаемых исключительных ситуаций в заголовке функции.35. Полиморфизм в C++: виртуальные функции, их назначение, объявление.36. Чисто виртуальные функции и абстрактные классы.37. Простое наследование: определение, синтаксис. Порядок выполнения конструкторов и деструкторов. Вызов методов, переопределенных в производном классе, из базового класса.38. Множественное наследование: определение, виртуальные классы. Изменения в порядке выполнения конструкторов и деструкторов для виртуальных классов. Вызов методов с одинаковыми сигнатурами из разных базовых классов.39. Операция typeid. RTTI.40. Операция безопасного преобразования данных const\_cast. Назначение, синтаксис вызова.41. Операция безопасного преобразования данных static\_cast. Назначение, синтаксис вызова.42. Операция безопасного преобразования данных dynamic\_cast. Назначение, синтаксис вызова.43. Операция безопасного преобразования данных reinterpret\_cast. Назначение, синтаксис вызова.44. Пространства имен: назначение, определение (синтаксис), варианты использования имен из namespace в своей программе.45. Какие методы должны быть определены в классе с динамическим выделением памяти для некоторых элементов данных?46. Сложность программного обеспечения.47. Пять свойств сложной системы.48. Основные методы при создании сложных систем.49. Основные положения ОО подхода.50. Концепции ОО подхода: Абстрагирование.51. Концепции ОО подхода: Ограничение доступа.52. Концепции ОО подхода: Модульность.53. Концепции ОО подхода: Иерархия.54. Концепции ОО подхода: Типизация.55. Концепции ОО подхода: Параллелизм.56. Концепции ОО подхода: Устойчивость (сохраняемость).57. Объекты в ООП: Определение объекта.58. Объекты в ООП: Состояние.59. Объекты в ООП: Поведение. Операции.60. Объекты в ООП: Уникальность (идентичность).61. Объекты в ООП: Отношения между объектами.62. Классы в ООП: Понятие класса, связь между объектами и классами.63. Отношения между классами: Ассоциации.64. Отношения между классами: Наследование.65. Отношения между классами: Агрегация.66. Отношения между классами: Использование.67. Отношения между классами: Конкретизация (параметризованные классы).68. Отношения между классами: Метаклассы.69. Паттерны проектирования: Абстрактная фабрика.70. Паттерны проектирования: Одиночка.61. Паттерны проектирования: Прототип (виртуальный конструктор).72. Паттерны проектирования: Адаптер.73. Паттерны проектирования: Заместитель.74. Паттерны проектирования: Компоновщик.75. Паттерны проектирования: Декоратор.76. Паттерны проектирования: Итератор.77. Паттерны проектирования: Шаблонный метод.