

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ



УТВЕРЖДАЮ
Декан РТФ

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

01.03.2023 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
Семестр 4

Распределение учебного времени

| | | |
|---|---------|-----------------------|
| Трудоемкость по учебному плану | 180 / 5 | часов/зачетных единиц |
| Лекции | 18 | часов |
| Лабораторные работы | 36 | часов |
| Практические занятия | 36 | часов |
| Иная контактная работа | - | часов |
| Всего контактной работы (без учета экз.) | 90 | часов |
| Контактная работа по экзамену | 6 | часов |
| Курсовой проект (работа) | - | семестр |
| Самостоятельная работа обучающихся (без учета экз.) | 54 | часов |
| Самостоятельная работа по подготовке к экзамену | 30 | часов |
| Экзамен | 4 | семестр |
| Зачет | - | семестр |
| БРК, ДЗ | - | 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 (должность) | ПиП ЭВС (кафедра) | СОГЛАСОВАНО | Д.О. Глухов (И.О. Фамилия) |
| преподаватель (должность) | ПиП ЭВС (кафедра) | СОГЛАСОВАНО | Д.О. Глухов (И.О. Фамилия) |

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

| | | | |
|------------------------|-------------|---------------------------------|--|
| (наименование кафедры) | | | |
| 16.01.2023 (дата) | протокол № | 8 | |
| 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 06.03.2023 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 | знания: Знания принципов работы современных информационных технологий умения: Умеет разрабатывать технологическую документацию навыки: 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 | знания: Знает оптимальные способы решения задач умения: Умеет решать задачи профессиональной деятельности навыки: 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 | знания: Знает современные информационные технологии для проектирования и разработки программного обеспечения с целью решения задач профессиональной деятельности умения: Использует современные информационные технологии для проектирования и разработки программного обеспечения с целью решения задач профессиональной деятельности навыки: 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 | знания: Знает прикладные программы решения практических задач для электронно-вычислительных средств, пригодные для задач в профессиональной деятельности умения: Умеет разрабатывать прикладные программы решения практических задач для электронно-вычислительных средств, пригодные для задач в профессиональной деятельности навыки: 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 | знания: Знает методы алгоритмизации, языки и технологии программирования, пригодные для практического применения в области информационных систем и технологий умения: навыки: |
| ОПК-5.2 Умеет применять методы алгоритмизации, языки и технологии программирования при решении профессиональных задач в области информационных систем и технологий | знания: умения: Умеет применять методы алгоритмизации, языки и технологии программирования при решении профессиональных задач в области информационных систем и технологий навыки: |
| ОПК-5.3 Владеет навыками программирования, отладки и тестирования прототипов программно-технических | знания: умения: навыки: 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-5), Учебная практика (ознакомительная) (ОПК-4)

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

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: задания, классическая лекция

Раздел 4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

4 семестр

| Виды и темы занятий | Количество часов | Формируемые компетенции |
|--|------------------|-------------------------|
| Язык программирования Си++ | 144 | ОПК-4, ОПК-5 |
| Лекция. Введение. Языки программирования. Основные понятия и определения. | 2 | |
| Лекция. Объектно-ориентированное программирование и процедурное программирование - отличия и преимущества | 4 | |
| Лекция. Типы данных и переменные. Вывод на экран ввод данных. Функции в С++ | 4 | |
| Лекция. Классы в С++. Методы и интерфейсы. Абстрактные классы. | 4 | |
| Лекция. Структуры и шаблоны. Многопоточное программирование. | 4 | |
| Лабораторная работа. Работа с функциями в концепции чистого кода | 6 | |
| Лабораторная работа. Объектно-ориентированное программирование на С++ с использованием классов | 6 | |
| Лабораторная работа. Методы и интерфейсы в С++ | 6 | |
| Лабораторная работа. Абстрактные классы в С++ | 6 | |
| Лабораторная работа. Шаблоны и структуры в С++ | 6 | |
| Лабораторная работа. Многопоточное программирование в С++ | 6 | |
| Практическое занятие. Паттерны проектирования | 6 | |
| Практическое занятие. Концепции чистого кода | 6 | |
| Практическое занятие. Многофайловые проекты | 6 | |
| Практическое занятие. Поддержка версий | 6 | |
| Практическое занятие. Сопровождение кода и отладка | 6 | |
| Практическое занятие. Работа с ошибками | 6 | |
| 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. | 54 | |
| 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.

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. | Рацеев, С. М. Программирование на языке Си. [Электронный ресурс] / Рацеев С. М. Санкт-Петербург: Лань, 2023. - 332 с. ISBN 978-5-8114-8585-7. | https://e.lanbook.com/book/351863 |
| 2. | Барков, И. А. Объектно-ориентированное программирование [Электронный ресурс] : учебник / Барков И. А. Санкт-Петербург: Лань, 2023. - 700 с. ISBN 978-5-8114-3586-9. | https://e.lanbook.com/book/329549 |
| 3. | Федоров, Дмитрий Юрьевич. Программирование на языке высокого уровня Python [Текст : Электронный ресурс] : учебное пособие для вузов / Д. Ю. Федоров. 5-е изд. Москва: Юрайт, 2023. - 210 с ISBN 978-5-534-14638-7. | https://urait.ru/book/programirovanie-na-yazyke-vysokogo-urovnya-python-532868 |
| 4. | 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. | https://e.lanbook.com/book/297002 |
| ЭЛЕКТРОННЫЕ ОБРАЗОВАТЕЛЬНЫЕ РЕСУРСЫ | | |
| 1. | Научная электронная библиотека eLIBRARY.RU | http://elibrary.ru |

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 200x200 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 |

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 затруднения в выполнении практических заданий. | удовлетворительно |
| 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 | отлично |

| | | |
|--|--|--|
| | 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 | |
|--|--|--|

7.1. Текущий контроль успеваемости

Текущий контроль успеваемости обеспечивает оценивание хода освоения дисциплины (модуля) и производится с применением технологии рейтингового контроля в соответствии с технологической картой дисциплины. Порядок составления технологической карты и алгоритм проведения процедуры оценивания видов деятельности обучающихся, направленных на освоение знаний, умений, навыков и/или опыта деятельности, по накопительной системе в баллах устанавливается положением о системе РИТМ в ФГБОУ ВО «ПГТУ»

7.2. 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. Какое из следующих утверждений верно для этого кода?

```
#include<stdio.h>
int main()
{
    int x = 10, y = 100%90, i;
    for(i=1; i<10; i++)
        if(x != y);
    printf("x = %d y = %d\n", x, y);
    return 0;
}
```

1 : Функция `printf()` будет выполнена 10 раз.

2 : Выход программы будет `x = 10 y = 10`

3 : `;` след `if(x!=y)` не приведет к ошибке.

4 : Программа не будет иметь никакого выхода.

1

2,3

3,4

4

2. К чему приведет программа ниже?

```
#include<stdio.h>
int main()
{
    int k, num=30;
    k = (num>5 ? (num <=10 ? 100 : 200) : 500);
    printf('%d\n', num);
    return 0;
}
```

30

100

500

200

3. Какая ошибка в следующей программе:

```
#include<stdio.h>
int main()
{
    struct emp
    {
        char name[20];
        float sal;
    };
    struct emp e[10];
    int i;
    for (i=0; i<=9; i++)
        scanf('%s %f', e[i].name, &e[i].sal);
    return 0;
}
```

Strings cannot be nested inside structures

Cannot use `scanf()` for structures

Suspicious pointer conversion

Floating point formats not linked (Run time error)

4. Команда, которая используется для передачи управления из функции к вызывающей функции, является:

return

switch

goto

go back

5. Какое из следующих утверждений верно для функции?

```
long fun(int num)
{
    int i;
    long f=1;
    for (i=1; i<=num; i++)
        f = f * i;
    return f;
}
```

Функция вычисляет факториал целого числа

Ни один из ответов не является правильным

Функция вычисляет квадратный корень из целого числа

Функция вычисляет 1 на n -й степени

6. Какое из следующих утверждений верно для этой программы?

```
#include<stdio.h>

int main()
{
    float a=3.14;
    char *j;
    j = (char*)&a;
    printf("%d\n", *j);
    return 0;
}
```

Покажет 3

Покажет ASCII двоичный разряд, расположенный в битовой области процессора типа float a

Покажет значение неинициализированной переменной

Принесет символьный код, равный двоичному коду, который находится в первом байте переменной типа float a

7. Будет ли следующая программа компилироваться в Turbo C?

```
#include<stdio.h>
int main()
{
    int a=10, *j;
    void *k;
    j=k=&a;
    j++;
    k++;
    printf("%u %u\n", j, k);
    return 0;
}
```

Нет

Да

8. Какие из следующих утверждений верно для типа массива(array)?

Массив `int num[26];` может сохранить 26 элементов.

2 : `num[1]` обозначается первый элемент в массиве.

3 : Нужно инициализировать массив во время декларации.

4 : Декларация `num[SIZE]` позволена, если SIZE макрос.

2, 3

2, 4

1

1, 4

9. Стандартная функция, которая используется, чтобы найти последние соответствия символа в строке, это:

`strnstr()`

`strrchr()`

`aststr()`

`strstr()`

10. Если размер `char=1`, `int=4` и `float=4` байтов, каким будет результат следующей программы?

```
#include<stdio.h>

int main()
{
    char ch = 'A';
    printf("%d, %d, %d", sizeof(ch), sizeof('A'), sizeof(3.14f));
    return 0;
}
```

1, 4, 4

2, 4, 8

2, 2, 4

1, 2, 4

11. К чему приведет программа ниже?

```
#include<stdio.h>
```

```
int main()
{
    char str[] = "London";
    str[1]='i';
    printf("%s, ", str);
    str = "Bristol%s", str+1);
    return 0;
}
```

London, Bristol

Lindon, istol

Lindon, Bristol

Error

12. К чему приведет программа ниже?

```
#include<stdio.h>
```

```
int main()
{
    int i=4, j=8;
    printf("%d, %d, %d\n", i|j&j|i, i|j&j|i, i^j);
    return 0;
}
```

32, 1, 12

12, 12, 12

112, 1, 12

-64, 1, 12

13. Битовые поля могут использоваться в **union**

Не

Да

14. Какое из следующих утверждений верно для следующей программы?

```
#include<stdio.h>
```

```
int main()
{
    FILE *fp;
    char ch;
    int i=1;
    fp = fopen("myfile.c", "r");
    while( (ch=getc(fp)) !=EOF)
    {
        if(ch == '\n')
            i++;
    }
    fclose(fp);
    return 0;
}
```

Вычисляет количество символов в файле.

Вычисляет количество слов в файле.

Вычисляет количество пустых строк в файле
Вычисляет количество строк в файле

15. Первый аргумент в командной строке всегда должен показывать общее количество аргументов.
Нет
Да

16. Каким будет результат программы?

```
#include<stdio.h>

int main()
{
    char c=48;
    int i, mask=01;
    for (i=1; i<=5; i++)
    {
        printf("%c", c|mask);
        mask = mask<<1;
    }
    return 0;
}
```

12400
12500
12480
12556

17. Каким будет результат программы?

```
typedef struct data;
{
    int x;
    sdata *b;
}sdata;
```

Никто из указанных
Error: Declaration missing ';'
Error: in typedef.
No error

18. Какой header файла нужно включить, если будете использовать функцию, которая может принимать разное количество аргументов?

stdlib.h
stdio.h
stdarg.h
vararg.h

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.