

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ИМЕНИ В.Ф. УТКИНА»**

Кафедра «Вычислительная и прикладная математика»

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ
«Программирование»**

Направление подготовки
09.03.04 «Программная инженерия»

Направленность (профиль) подготовки
«Программная инженерия систем искусственного интеллекта»

Уровень подготовки – бакалавриат

Квалификация выпускника – бакалавр

Форма обучения – очная

Срок обучения – 4 года

1. ОБЩИЕ ПОЛОЖЕНИЯ

Оценочные материалы - это совокупность учебно-методических материалов и процедур, предназначенных для оценки качества освоения обучающимися данной дисциплины как части основной образовательной программы.

Цель - оценить соответствие знаний, умений и уровня приобретенных компетенций, обучающихся целям и требованиям основной образовательной программы в ходе проведения текущего контроля и промежуточной аттестации.

Основная задача - обеспечить оценку уровня сформированности компетенций, приобретаемых обучающимися в соответствии с этими требованиями.

Контроль знаний обучающихся проводится в форме зачета в 8-м семестре.

2. ОПИСАНИЕ ПОКАЗАТЕЛЕЙ И КРИТЕРИЕВ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ

Сформированность каждой компетенции в рамках освоения данной дисциплины оценивается по трехуровневой шкале:

- 1) пороговый уровень является обязательным для всех обучающихся по завершении освоения дисциплины;
- 2) продвинутый уровень характеризуется превышением минимальных характеристик сформированности компетенций по завершении освоения дисциплины;
- 3) эталонный уровень характеризуется максимально возможной выраженностью компетенций и является важным качественным ориентиром для самосовершенствования.

Уровень освоения компетенций, формируемых дисциплиной

а) описание критериев и шкалы оценивания тестирования:

Шкала оценивания	Критерий
3 балла (эталонный уровень)	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 85 до 100%
2 балла (продвинутый уровень)	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 75 до 84%
1 балл (пороговый уровень)	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 60 до 74%
0 баллов	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 0 до 59%

б) описание критериев и шкалы оценивания теоретического вопроса:

Шкала оценивания	Критерий
3 балла (эталонный уровень)	выставляется студенту, который дал полный ответ на вопрос, показал глубокие систематизированные знания, смог привести примеры, ответил на дополнительные вопросы преподавателя.

2 балла (продвинутый уровень)	выставляется студенту, который дал полный ответ на вопрос, но на некоторые дополнительные вопросы преподавателя ответил только с помощью наводящих вопросов.
1 балл (пороговый уровень)	выставляется студенту, который дал неполный ответ на вопрос в билете и смог ответить на дополнительные вопросы только с помощью преподавателя.
0 баллов	выставляется студенту, который не смог ответить на вопрос.

в) описание критериев и шкалы оценивания практического задания:

Шкала оценивания	Критерий
3 балла (эталонный уровень)	Задание решено верно.
2 балла (продвинутый уровень)	Задание решено верно, но имеются технические неточности в выполнении.
1 балл (пороговый уровень)	Задание решено верно, с дополнительными наводящими вопросами преподавателя.
0 баллов	Задание не решено.

На экзамен выносятся: тестовое задание, 1 практическое задание и 1 теоретический вопрос. Студент может набрать максимум 9 баллов. Итоговый суммарный балл студента, полученный при прохождении промежуточной аттестации, переводится в традиционную форму по системе «отлично», «хорошо», «удовлетворительно», «неудовлетворительно». Обязательным условием получения оценок «отлично», «хорошо» и «удовлетворительно» является выполнение всех предусмотренных в течение семестра практических заданий и лабораторных работ. Студент не выполнивший всех предусмотренных в течение семестра текущих заданий получает оценку «неудовлетворительно»

Шкала оценивания	Критерий
отлично (эталонный уровень)	8 - 9 баллов (все задания и лабораторные работы выполнены)
хорошо (продвинутый уровень)	6 - 7 баллов (все задания и лабораторные работы выполнены)
удовлетворительно (пороговый уровень)	4 - 5 баллов (все задания и лабораторные работы выполнены)
неудовлетворительно	0 - 3 баллов (студент не выполнил все задания и лабораторные работы)

3. ПАСПОРТ ОЦЕНОЧНЫХ МАТЕРИАЛОВ ПО ДИСЦИПЛИНЕ

Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции (или её части)	Наименование оценочного средства
Тема 1. Введение в программирование и среду разработки	ОПК-3, ОПК-6, ОПК-7, ПК-15	Экзамен
Тема 2. Основы алгоритмизации и линейные программы	ОПК-3, ОПК-6, ОПК-7, ПК-15	Экзамен
Тема 3. Разветвления и условия.	ОПК-3, ОПК-6, ОПК-7, ПК-15	Экзамен
Тема 4. Циклы и их применение.	ОПК-3, ОПК-6, ОПК-7, ПК-15	Экзамен
Тема 5. Структуры данных: списки, кортежи, множества.	ОПК-3, ОПК-6, ОПК-7, ПК-15	Экзамен
Тема 6. Работа с матрицами и строками.	ОПК-3, ОПК-6, ОПК-7, ПК-15	Экзамен
Тема 7. Функции, подпрограммы и рекурсия.	ОПК-3, ОПК-6, ОПК-7, ПК-15	Экзамен
Тема 8. Работа с файлами и исключениями.	ОПК-3, ОПК-6, ОПК-7, ПК-15	Экзамен
Тема 9. Модульное программирование и библиотеки.	ОПК-3, ОПК-6, ОПК-7, ПК-15	Экзамен
Тема 10. Графические интерфейсы и событийная модель.	ОПК-3, ОПК-6, ОПК-7, ПК-15	Экзамен
Тема 11. Введение в C++: основы и конструкции.	ОПК-3, ОПК-6, ОПК-7, ПК-17	Экзамен
Тема 12. ООП, шаблоны и работа с памятью	ОПК-3, ОПК-6, ОПК-7, ПК-17	Экзамен

4. ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ

4.1 Промежуточная аттестация

ОПК-3: Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности;
ОПК-6: Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов;
ОПК-7: Способен применять в практической деятельности основные концепции, принципы, теории и факты, связанные с информатикой;
ПК-15: Способен применять язык программирования Python для решения задач в области ИИ

Закрытые вопросы (с вариантами ответов)

1. Что выведет следующий код?

```
print(2 ** 3 + 5)
```

- a) 11
- b) 13
- c) 16
- d) Ошибка

Правильный ответ: b) 13

2. Какой метод списка используется для добавления элемента в конец?

- a) append()
- b) insert()
- c) add()
- d) extend()

Правильный ответ: a) append()

3. Для генерации случайного целого числа в диапазоне [1, 10] используется:

- a) random.random()
- b) random.randint(1, 10)
- c) random.uniform(1, 10)
- d) random.choice([1, 10])

Правильный ответ: b) random.randint(1, 10)

4. Какой оператор позволяет досрочно завершить выполнение цикла?

- a) exit
- b) stop
- c) break
- d) continue

Правильный ответ: c) break

5. Что такое "list comprehension" (генератор списка)?

- a) Функция для чтения списка из файла
- b) Синтаксис для создания списка на основе итерируемого объекта
- c) Метод для сортировки списка
- d) Оператор для объединения списков

Правильный ответ: b) Синтаксис для создания списка на основе итерируемого объекта

6. Какой метод строки возвращает новый строковый объект, где все символы в нижнем регистре?

- a) lower()
- b) casefold()
- c) toLower()
- d) small()

Правильный ответ: a) lower()

7. Как открыть текстовый файл 'data.txt' для дозаписи?

- a) open('data.txt', 'r')
- b) open('data.txt', 'w')
- c) open('data.txt', 'a')
- d) open('data.txt', 'x')

Правильный ответ: c) open('data.txt', 'a')

8. Какая функция используется для сериализации объекта Python в бинарный файл с помощью модуля pickle?

- a) save()
- b) dump()
- c) write()
- d) serialize()

Правильный ответ: b) dump()

9. Какой из этих алгоритмов сортировки относится к обменным методам?

- a) Сортировка выбором
- b) Сортировка вставками
- c) Сортировка пузырьком
- d) Бинарная сортировка

Правильный ответ: c) Сортировка пузырьком

10. Что такое lambda-функция?

- a) Рекурсивная функция
- b) Анонимная функция, определяемая одним выражением
- c) Функция из модуля lambda
- d) Функция, которая выполняется только в определенных условиях

Правильный ответ: b) Анонимная функция, определяемая одним выражением

Открытые вопросы (со свободным ответом)

1. Перечислите три ключевых слова, используемых в условных операторах Python.

(if, elif, else)

2. Какой оператор в цикле for используется для пропуска текущей итерации и перехода к следующей?

(continue)

3. Как называется метод, позволяющий получить из списка часть его элементов (например, каждый второй)?

(Срез / Slicing)

4. Какой метод словаря возвращает все его ключи в виде специального объекта?

(keys())

5. С помощью какого оператора можно проверить, принадлежит ли элемент множеству?

(in)

6. Какой модуль необходимо импортировать для использования математических функций, таких как sin и sqrt?

(math)

7. Какой метод списка удаляет последний элемент и возвращает его?

(pop())

8. Как называется структура данных в Python, которая хранит пары "ключ-значение" и является неупорядоченной?

(Словарь / Dictionary)

9. Какой метод строки используется для разделения строки на список подстрок по заданному разделителю?

(split())

10. Какой параметр функции print() используется для указания символа-разделителя между выводимыми аргументами?

(sep)

ОПК-3: Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности;
ОПК-6: Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов;
ОПК-7: Способен применять в практической деятельности основные концепции, принципы, теории и факты, связанные с информатикой;
ПК-17: Способен применять языки программирования C/C++ для решения задач в области ИИ

Закрытые вопросы (с вариантами ответов)

1. Какой тип данных в C++ используется для хранения целых чисел двойной точности?

- a) int
- b) float
- c) double
- d) long long

Правильный ответ: c) double

2. Какой оператор используется для динамического выделения памяти под экземпляр класса?

- a) malloc
- b) new
- c) alloc
- d) create

Правильный ответ: b) new

3. Что такое nullptr?

- a) Макрос, обозначающий ноль
- b) Ключевое слово, обозначающее нулевой указатель
- c) Функция для освобождения памяти
- d) Синоним для void*

Правильный ответ: b) Ключевое слово, обозначающее нулевой указатель

4. Какой из способов передачи параметра в функцию позволяет изменить значение исходной переменной?

- a) По значению
- b) По константной ссылке
- c) По указателю
- d) По константному указателю

Правильный ответ: c) По указателю

5. Что такое перегрузка функций?

- a) Изменение реализации существующей функции
- b) Создание нескольких функций с одним именем, но разными параметрами
- c) Ускорение работы функции
- d) Объявление функции внутри другой функции

Правильный ответ: b) Создание нескольких функций с одним именем, но разными параметрами

6. Какой модификатор доступа используется для членов класса, которые должны быть доступны только из методов этого же класса?

- a) public
- b) private
- c) protected
- d) internal

Правильный ответ: b) private

7. Какой метод вызывается автоматически при уничтожении объекта?

- a) Конструктор
- b) Деструктор
- c) Финализатор
- d) delete

Правильный ответ: b) Деструктор

8. Что позволяет сделать ключевое слово virtual?

- a) Сделать функцию константной
- б) Запретить переопределение функции
- в) Реализовать полиморфизм
- г) Объявить статическую функцию

Правильный ответ: в) Реализовать полиморфизм

9. Для чего используется ключевое слово const в объявлении метода класса?

- a) Чтобы метод не мог изменять поля объекта
- б) Чтобы метод возвращал константное значение
- в) Чтобы метод был статическим
- г) Чтобы метод нельзя было переопределить

Правильный ответ: а) Чтобы метод не мог изменять поля объекта

10. Что такое шаблон (template) в C++?

- a) Готовый класс для работы с контейнерами
- б) Механизм для написания кода, который работает с разными типами данных
- в) Файл с заголовочными объявлениями
- г) Схема наследования классов

Правильный ответ: б) Механизм для написания кода, который работает с разными типами данных

Открытые вопросы (со свободным ответом)

1. Как называется этап компиляции, на котором работает препроцессор, обрабатывая директивы #include и #define?

(Препроцессинг / Preprocessing)

2. Какая стандартная библиотека C++ предоставляет потоки ввода-вывода (например, cout, cin)?

(iostream)

3. Какой оператор используется для получения адреса переменной?

(& / амперсанд)

4. Что такое "утечка памяти" (memory leak)?

(Ситуация, когда память, выделенная динамически, не освобождается после того, как перестала быть нужной)

5. Как называется специальный метод класса, который вызывается при создании нового объекта?
(Конструктор)
6. Какое ключевое слово используется для явного указания, что производный класс переопределяет виртуальный метод базового?
(override)
7. Какой тип приведения используется для безопасного приведения типов в иерархии наследования?
(dynamic_cast)
8. Как называется класс, который содержит хотя бы одну чистую виртуальную функцию и не может иметь экземпляров?
(Абстрактный класс)
9. Какой механизм в C++ предназначен для обработки ошибок времени выполнения?
(Исключения / Exceptions)
10. Как называется связь между объектами, когда один объект является частью другого и не может существовать без него?
(Композиция)

4.2 Типовые контрольные вопросы

Семестр 1.

1. Язык python. Основные определения. Ключевые слова.
2. Операторы языка. Виды операторов. Приоритеты операторов.
3. Встроенные функции языка python.
4. Модуль random. Работа со случайными числами.
5. Функции ввода и вывода.
6. Функция вывода. Форматирование вывода.
7. Условные операторы. Полные условные операторы. Неполные условные операторы. Тернарный оператор условия. Примеры использования.
8. Условные операторы. Множественный выбор. Вложенные операторы условия. Примеры использования.
9. Операторы цикла. Оператор цикла с условием. Операторы break и continue. Примеры использования.
10. Операторы цикла. Оператор цикла с итератором. Примеры использования.
11. Модуль math. Основные функции модуля. Примеры использования функций.
12. Строки. Основные функции и методы работы со строками. Пример программы.
13. Списки. Создание списков. Генераторы списков.
14. Списки. Основные методы для работы с элементами списка. Добавление элемента, вставки, удаление, поиск.
15. Списки. Основные операции со списками. Поиск минимального элемента. Поиск максимального элемента. Нахождение количества элементов. Нахождение суммы и произведения элементов.
16. Списки. Использование срезов при обработке списков.
17. Списки. Сортировка элементов списка.
18. Списки. Сортировка. Сортировка вставками. Сортировка выбором.

19. Списки. Сортировка вставками. Метод простых вставок. Метод вставок с бинарным поиском. Вставки с барьером. Метод Шелла.
20. Списки. Сортировка. Обменные методы сортировки. Сортировка пузырьком. Сортировка пузырьком с флагом. Метод шейкер-сортировки.
21. Списки. Сортировка. Метод быстрой сортировки.
22. Множества. Определение. Основные операции.
23. Словари. Понятие ключей и значений. Создание словарей. Основные методы словарей.
24. Матрицы. Создание матрицы. Ввод и вывод матрицы. Выполнение операций с элементами матрицы.
25. Матрицы. Квадратные матрицы. Обработка верхне- и нижнетреугольных матриц. Работа с диагональными элементами матрицы.
26. Функции. Создание функции. Аргументы функции. Возвращаемое значение.
27. Функции. Lambda-функции. Рекурсивные функции.
28. Файлы. Текстовые файлы. Открытие файла. Режимы доступа к файлу.
29. Файлы. Текстовые файлы. Чтение файла. Запись в файл. Поиск в файле.
30. Файлы. Текстовые файлы. Итерационное чтение содержимого файла.
31. Файлы. Бинарные файлы. Сериализация данных. Модуль pickle.
32. Модуль numpy. Обработка массивов с использованием данного модуля.
33. Модуль numpy. Работа с числами и вычислениями.
34. Модуль matplotlib. Построение графиков в декартовой системе координат. Управление областью рисования.
35. Модуль matplotlib. Построение гистограмм и круговых диаграмм.

Семестр 2.

1. Чем характеризуется язык C++? Для чего он применяется?
2. Что такое статическая и динамическая типизация?
3. Что такое уровневость языка? В чем разница между низко- и высокоуровневыми языками?
4. Чем отличаются компилируемые языки от интерпретируемых?
5. Для чего нужна функция main? Что значит её возвращаемое значение?
6. Формат объявления констант и переменных?
7. Что такое область видимости? Какие области видимости бывают?
8. Какие есть целочисленные и вещественные типы данных в языке C++? Перечислите их в порядке возрастания размера.
9. Что такое автоматическое приведение типов? Каковы правила автоматического приведения типов?
10. Что такое явное преобразование типа и как оно осуществляется?
11. Для чего нужны пространства имен и как их объявлять? Для чего используется ключевое слово using и какой формат использования?
12. Что такое потоки ввода-вывода и как с ними работать в C++?
13. Какие есть операции в C++? Какие есть математические функции? Приведите примеры.
14. Для чего нужны функции rand() и srand()? Каков их формат?
15. Укажите формат условного оператора и оператора выбора.
16. Укажите формат операторов цикла.
17. Что такое массив? Как происходит обращение к элементам массива? Какими понятиями характеризуется массив?
18. Формат объявления и инициализации массивов? Приведите пример объявления и инициализации одномерного и двумерного массива.
19. Что такое функция? Формат объявления и вызова функции?

20. Для чего нужно предварительное объявление функций? Как объявить функцию, не принимающую аргументов или не возвращающую значение?
21. Что такое константные параметры функции и для чего применяются?
22. Как передаётся массив в функцию через параметр? Приведите примеры.
23. Что такое перегрузка функций? Как компилятор определяет какая функция вызывается? Приведите примеры.
24. Что такое указатель? Как объявляется указатель? Какие операции определены для указателей?
25. Что такое нулевой указатель и для чего он применяется? Зачем был введен `nullptr` и в чём его отличие от `NULL`?
26. Какие существуют способы передачи параметров функции? В каких случаях какой способ необходимо выбрать?
27. Что такое константный указатель и для чего он применяется? Какие есть виды константного указателя?
28. Как связаны массивы и указатели? Что такое индексация массива с точки зрения указателей?
29. Как представляются одномерные и двумерные массивы в памяти компьютера? Как передавать массив в функцию через указатель?
30. Что такое пустой указатель и для чего он применяется?
31. Как объявить указатель на функцию? Для чего он может применяться?
32. На какие области делится память, выделяемая для процесса?
33. Какие способы хранения объектов в памяти существуют?
34. Что такое динамическая память? Какие преимущества даёт динамическое выделение памяти?
35. Что такое статическая память? Какие преимущества существуют у статического выделения памяти?
36. Формат и назначение функции `malloc` и `free`?
37. Формат и назначение функции `calloc` и `realloc`?
38. Формат и назначение функции `memcpy` и `memset`?
39. Как создать одномерный и двумерный динамический массив указанного размера? Что такое «утечка памяти»?
40. Что такое строковый тип данных? Как хранится строка в памяти компьютера?
41. Как происходит компиляция программ? Что такое этап препроцессинга и зачем он нужен?
42. Как происходит компиляция программ? Что такое этап компиляции и зачем он нужен?
43. Как происходит компиляция программ? Что такое этап ассемблирования и зачем он нужен?
44. Как происходит компиляция программ? Что такое этап компоновки и зачем он нужен?
45. Для чего нужны директивы `#include` и `#define` и каков их формат?
46. Что такое условная компиляция? Какие директивы препроцессора используются для нее?
47. Что такое модуль и из каких частей он состоит?
48. Что такое статические и глобальные переменные?
49. Что такое структура и для чего она нужна? Как происходит объявление структуры? Как происходит копирование структуры?
50. Какими способами можно передавать структуру в качестве аргумента функции и в чём отличие между ними?
51. Что такое перечисление и для чего оно нужно? Чем отличается `enum` от `enum class`?

52. Что такое побитовая операция? Какие побитовые операции существуют?
53. Что такое битовые флаги и маски?
54. Что такое ООП? Какова цель внедрения ООП? Что такое объект и что такое класс?
55. Что такое объект и что такое класс? Дайте определение принципа ООП абстракции и инкапсуляции.
56. Что такое объект и что такое класс? Дайте определение принципа ООП наследования и полиморфизма.
57. Какие уровни доступа есть при объявлении класса? Формат объявления класса?
58. Что такое конструктор и деструктор? Формат объявления конструктора и деструктора?
59. Зачем нужна интерфейсная часть и часть реализации при объявлении класса в программе? Что такое указатель `this` и в каких случаях он применяется?
60. Какие рекомендации существуют по объявлению классов?
61. Что такое юнит-тестирование? На какие части обычно разделен код теста?
62. Что такое конструктор копирования? В каких случаях он объявляется явно? Приведите его формат.
63. Что такое ссылки и в чём их отличие от указателей? В каких случаях надо применять ссылки, а в каких указатели?
64. Зачем используются ссылки как параметры функции? А в качестве возвращаемого значения? Приведите примеры.
65. Что такое статические поля и методы класса? Зачем они используются? Каковы правила работы с ними?
66. Что такое перегрузка операторов и какие её виды бывают? Как она объявляется? Какие операторы нельзя перегружать?
67. Что такое константные методы класса и для чего они нужны?
68. Что такое наследование и для чего оно нужно? Что такое суперкласс, подкласс, базовый класс?
69. В каком порядке при наследовании вызываются конструкторы и деструкторы? Какие типы наследования бывают?
70. Что такое переопределение метода и зачем оно используется? Зачем нужно ключевое слово `auto`?
71. Что такое множественное наследование и какие проблемы с ним связаны?
72. С помощью каких операторов происходит выделение\освобождение динамической памяти в C++? В чём их преимущество?
73. Что такое дружественная функция и класс? Когда их стоит использовать?
74. Что такое полиморфизм? Какие виды полиморфизма бывают и какие механизмы языка C++ их реализуют?
75. Что такое полиморфизм? Что такое полиморфизм подтипов? Какой класс называют полиморфным?
76. Что такое виртуальная функция? В чём отличие виртуальных функций от обычного переопределения? Для чего используется виртуальный деструктор?
77. Для чего нужны модификаторы `override` и `final`? Что такое чистые виртуальные функции? Что такое абстрактный класс?
78. Что такое интерфейсный класс и для чего он используется? Что такое контейнерные классы и какие виды их бывают?
79. Какие типы связей между объектами вы знаете? Что такое тип связи композиция? Приведите пример.
80. Какие типы связей между объектами вы знаете? Что такое тип связи агрегация? Приведите пример.

81. Какие типы связей между объектами вы знаете? Что такое тип связи ассоциация? Приведите пример.
82. Какие типы связей между объектами вы знаете? Что такое тип связи зависимость? Приведите пример.
83. Какие типы связей между объектами вы знаете? Что такое тип связи наследование? Приведите пример.
84. Что такое шаблоны и для чего они применяются? Как объявить шаблон функции и шаблон класса? Приведите примеры.
85. Что такое инстанцирование шаблона и в какой момент оно происходит? Какие преимущества и недостатки есть у шаблонов?
86. Что такое non-type параметры шаблона, какого типа они могут быть? Что такое явная и частичная специализация шаблона и каким образом она объявляется?
87. Что такое `static_cast` и `dynamic_cast`?
88. Что такое `const_cast` и `reinterpret_cast`?
89. Что такое исключения? В чем преимущество исключений перед возвратом кодов ошибок?
90. Что такое исключения? Какова общая структура выбрасывания и обработки исключения? Из каких блоков она состоит?
91. Что такое исключения? Для чего нужны классы-исключений?
92. Что такое исключения? Каковы недостатки исключений? В каких случаях стоит использовать исключения?

4.3 Типовые задачи по дисциплине

Семестр 1.

1. Напишите программу, которая запрашивает у пользователя три числа и определяет, существует ли треугольник с такими сторонами. Если существует, определите его тип (равносторонний, равнобедренный, разносторонний).
2. Напишите программу, которая принимает строку и подсчитывает, сколько раз в ней встречается каждый символ (игнорируя регистр). Выведите результат в виде: "a: 5, b: 3, ...".
3. Напишите программу, которая находит все совершенные числа в диапазоне от 1 до N. Совершенное число — это число, равное сумме всех своих собственных делителей (например, $6 = 1 + 2 + 3$).
4. Дан список целых чисел. Создайте новый список, который содержит только уникальные элементы исходного списка, сохраняя их порядок.
5. Дан список целых чисел. Отсортируйте его по возрастанию последней цифры каждого числа. Например, список [101, 22, 43] должен превратиться в [101, 22, 43] (так как последние цифры 1, 2, 3).
6. Реализуйте алгоритм сортировки списка "пузырьком" без использования встроенной функции `sort()`.
7. Дан список. Создайте новый список, который является "сдвигом" исходного списка на K позиций вправо (элементы, выходящие за границу, появляются в начале). Используйте срезы.

8. Напишите функцию, которая принимает на вход список чисел и возвращает кортеж из двух элементов: минимального и максимального числа в списке. Не используйте встроенные функции `min()` и `max()`.
9. Создайте программу для учета оценок студентов. Используйте словарь, где ключ — имя студента, а значение — список его оценок. Реализуйте функции для добавления студента, добавления оценки и вычисления среднего балла для каждого студента.
10. Даны два списка. Напишите программу, которая находит все элементы, которые встречаются в обоих списках (пересечение), и все элементы, которые уникальны для каждого списка (симметрическая разность).
11. Дана квадратная матрица (список списков). Проверьте, является ли она симметричной относительно главной диагонали.
12. Напишите программу, которая читает текстовый файл, подсчитывает частоту встречаемости каждого слова (игнорируя регистр и знаки препинания) и записывает результат в новый файл в формате "слово: количество".
13. Напишите рекурсивную функцию для вычисления чисел Фибоначчи. Функция должна принимать номер числа N и возвращать N -ное число Фибоначчи.
14. Сгенерируйте два массива с помощью `numpy`: массив X из 100 точек в диапазоне от 0 до 10 и массив $Y = \sin(X)$. Постройте график Y от X с помощью `matplotlib`, добавьте подписи осей и заголовок.
15. Реализуйте программу "Шифр Цезаря". Она должна запрашивать у пользователя строку, шаг сдвига и направление (шифрование/дешифрование). Программа должна возвращать преобразованную строку, где каждая буква сдвинута в алфавите на указанный шаг.

Семестр 2.

1. Напишите функцию, которая принимает указатель на массив целых чисел и его размер. Функция должна возвращать сумму и произведение элементов массива. Используйте параметры по указателю или ссылке для возврата двух значений.
2. Напишите функцию, которая принимает строку (массив `char`) и определяет, является ли она палиндромом (читается одинаково слева направо и справа налево), игнорируя пробелы и регистр букв.
3. Напишите программу, которая создает одномерный динамический массив целых чисел заданного размера, заполняет его случайными значениями, находит минимальный элемент, а затем освобождает память.
4. Напишите рекурсивную функцию для бинарного поиска в отсортированном массиве целых чисел. Функция должна принимать массив, левую и правую границы и искомое значение, и возвращать индекс элемента или -1.
5. Напишите функцию `swap`, которая принимает два указателя на целые числа и обменивает значениями переменные, на которые они указывают.

6. Объявите структуру Student, содержащую поля: имя (массив char), фамилия, номер группы, массив оценок. Напишите функцию, которая принимает массив таких структур и находит студента с наивысшим средним баллом.
7. Создайте класс Rectangle (Прямоугольник). Закрытые поля: width, height. Публичные методы: конструктор, геттеры/сеттеры, методы для вычисления площади и периметра.
8. Для класса Rectangle из предыдущей задачи перегрузите оператор +, чтобы он возвращал новый прямоугольник с шириной и высотой, равными сумме соответствующих сторон двух других прямоугольников.
9. Модифицируйте класс Rectangle, добавив статическое поле count, которое подсчитывает количество созданных объектов этого класса.
10. Создайте класс Library (Библиотека), который содержит в себе массив объектов класса Book (Книга). Класс Book должен содержать название, автора и год издания. Реализуйте в Library методы для добавления книги и поиска книги по автору.
11. Создайте базовый класс Shape (Фигура) с чисто виртуальными методами area() и perimeter(). Создайте производные классы Circle (круг) и Square (квадрат), реализующие эти методы.
12. Создайте массив указателей типа Shape*. Заполните его адресами объектов Circle и Square. Напишите функцию, которая в цикле вызывает для каждого элемента массива метод area() и выводит результат, демонстрируя полиморфное поведение.
13. Продемонстрируйте необходимость виртуального деструктора. Создайте базовый класс с неvirtуальным деструктором и производный класс, в деструкторе которого выводится сообщение. Убедитесь, что при удалении объекта производного класса через указатель на базовый деструктор производного класса не вызывается. Затем сделайте деструктор базового класса виртуальным и покажите разницу.
14. Напишите шаблонную функцию sortThree, которая принимает три параметра одного типа по ссылке и сортирует их по возрастанию.
15. Модифицируйте класс Rectangle так, чтобы в сеттерах и конструкторе выбрасывалось исключение std::invalid_argument, если ширина или высота задаются отрицательным числом. Напишите код в main, который обрабатывает это исключение.