Согласовано
Начальник отдела эксплуатации и
внедрения информационных систем
ОГАУЗ СОМИАЦ
_____ Я.А.Комиссаров
«31» 08 2021 г.

УТВЕРЖДАЮ
Заместитель директора по
учебной работе
_____ И. В. Иванешко
«31» 08 2021 г.

Контрольно-оценочные средства для промежуточной аттестации
по дисциплине
ОПЦ. 04 ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ
для специальности 09.02.07 Информационные системы и программирование

Экзамен является промежуточной формой контроля, подводит итог освоения дисциплины ОПЦ. 04 ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ.

В результате освоения дисциплины студент должен освоить следующие **общие компетенции:**

ОК 1. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.

ОК 2. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.

ОК 4. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.

ОК 5. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.

ОК 9. Использовать информационные технологии в профессиональной деятельности.

ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языках.

профессиональные компетенции:

ПК 1.1.Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.

ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием.

ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств.

ПК 1.4. Выполнять тестирование программных модулей.

ПК 1.5. Осуществлять рефакторинг и оптимизацию программного кода.

ПК 2.4. Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.

ПК 2.5. Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.

В ходе проведения экзамена проверяется сформированность:

Умений:

У1. Разрабатывать алгоритмы для конкретных задач.

У2. Использовать программы для графического отображения алгоритмов.

У3. Работать в среде программирования.

У4. Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.

У5. Оформлять код программы в соответствии со стандартом кодирования.

У6. Выполнять тестирование, отладку кода программы.

У8. Обрабатывать ошибки и исключительные ситуации;

У10. Проводить рефакторинг и оптимизацию программного кода.

Знаний:

З1. Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции. Этапы решения задачи на компьютере;

З2. Эволюция языков программирования, их классификация, понятие системы программирования.

З3. Основные элементы языка программирования: простые и структурированные типы данных, структуру программы, операторы и операции, управляющие структуры, файлы, указатели.

З4. Основные принципы структурного программирования. Подпрограммы, составление библиотек подпрограмм.

З5. Стили и культура программирования. Стандарты кодирования. Рефакторинг и оптимизация кода.

З6. Объектно-ориентированная модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка.

З7. Структуры данных: стек, очередь, бинарное дерево, списки, множества и пр.

З11. Исключительные ситуации и ошибки;

З12. Базовые понятия отладки и тестирования.

Экзамен по дисциплине ОПЦ. 04 ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ проводится в форме тестирования.

Банк тестов содержит 180 вопросов (90 вопросов с выбором ответа, 90 вопросов с кратким ответом: по 30 вопросов на 3 группы компетенций).

Тест содержит 24 вопроса, выбираемых случайным образом программой из блоков заданий по группам компетенций:

ПК.1.1, ПК.1.2 - 8 вопросов (5 с выбором ответа, 3 с ответом);

ПК.1.3, ПК.1.4, ПК.2.4 – 8 вопросов (5 с выбором ответа, 3 с ответом);

ПК.1.5, ПК.2.5 - 8 вопросов (5 с выбором ответа, 3 с ответом).

Итого будет выбрано 15 вопросов с выбором ответа и 9 вопросов с ответом.

Время тестирования – 57 минут (по 2 минуты на вопрос с выбором ответа, 3 минуты на вопросы с ответом).

Критерии оценивания

«5» - получают студенты, справившиеся с работой 100-90%;

«4» - ставится в том случае, если верные ответы составляют 80-89% от общего количества;

«3» - соответствует работа, содержащая 60-79% правильных ответов;

«2» - соответствует работа, содержащая менее 60% правильных ответов.

Шкала оценивания образовательных результатов:

Оценка	Критерии
«отлично»	Студент набрал 5 баллов
«хорошо»	Студент набрал 4 балла
«удовлетворительно»	Студент набрал 3 балла
«неудовлетворительно»	Студент набрал 0-2 балла

Список вопросов:

Первый блок заданий - вопросы с выбором ответа
Формируемые компетенции: ОК1, ОК2, ОК4, ОК5, ОК9, ОК10, ПК 1.1., ПК 1.2.

1. Что представляет собой программа согласно технологии структурного программирования?
   a) последовательность действий
   b) несколько взаимодействующих объектов
   c) система определений функций, описание того, что нужно вычислить

2. Что представляет собой программа согласно технологии объектно-ориентированного программирования?
   a) последовательность действий
   b) несколько взаимодействующих объектов
   c) система определений функций, описание того, что нужно вычислить

3. Цикл какого типа должен выполниться хотя бы один раз?
   a) Цикл с предусловием;
   b) Цикл с постусловием;
   c) Цикл со счетчиком;
   d) Все циклические конструкции;

4. Какие бывают виды циклов?
   a) С подусловием;
   b) С предусловием;
   c) С заданным количеством повторений;
   d) С постепенным условием;

5. Какова особенность цикла с постусловием?
   a) Должен выполниться хотя бы один раз;
   b) Не выполнится ни разу;
   c) Выполнится один раз;
   d) Выполнится заданное количество раз;

6. Что представляет собой сортировка данных?
   a) Упорядочение данных
   b) Выбор только положительных элементов
   c) Нумерация элементов массива с 1

7. На каком этапе жизненного цикла программы пишется техническое задание?
   a) Маркетинг и спецификация программного продукта
   b) Проектирование структуры программного продукта
   c) Документирование программного продукта
   d) Программирование

8. На каком этапе решения задачи определяется форма выдачи результатов?
   a) Постановка задачи
   b) Анализ и исследование задачи
   c) Разработка алгоритма
   d) Программирование
   e) Тестирование и отладка

9. В чем сущность концепции модульного программирования?
   a) в разбиении программы на отдельные функционально независимые части;
   b) в разбиении программы на отдельные равные части;
   c) в разбиение программы на процедуры и функции;

10. Какой вид цилка присутствует в данном фрагменте программы?

```
int i=0;
int y=0;
do
 {   i++;
    y=y+1/i;}
while (1/i<e);
```

a) Цикл с предусловием;
b) Цикл с постусловием;
c) Цикл со счетчиком;
d) Тут нет циклических алгоритмов;

11. Какой вид цикла присутствует в данном фрагменте программы?

```
y=1; k=0;
while (y<=M)
{
  y=y*3;
  k++;
}
```

a) Цикл с предусловием;
b) Цикл с постусловием;
c) Цикл со счетчиком;
d) Нет циклических конструкций;

12. Сколько раз можно вызвать подпрограмму в основной программе?
a) 1 раз;
b) 10 раз;
c) Столько, сколько необходимо раз;

13. Как вызывается функция в программе?
a) Указанием имени;
b) Указанием и имени и фактических параметров в скобках, если они нужны;
c) Указанием и имени и формальных параметров в скобках;

14. Какие элементы может содержать тип Массив?
a) Только однотипные элементы;
b) Разнотипные элементы;
c) Только строковую информацию

15. Каковы основные принципы объектно-ориентированного программирования?
a) Наследование
b) Полиморфизм
c) Полиформизм
d) Инклюдирование
e) Инкапсуляция

16. Какое понятие более общее?
a) Класс
b) Объект
c) Конструктор
d) Атрибуты класса

17. Как называется свойство родственных классов решать сходные проблемы различными способами (алгоритмами)?
a) Наследование

b) Полиморфизм
c) Полиформизм
d) Инклюдирование
e) Инкапсуляция

18. Как называется свойство объектов порождать своих потомков?
    a) Наследование
    b) Полиморфизм
    c) Полиформизм
    d) Инклюдирование
    e) Инкапсуляция

19. Как называется относительно самостоятельная часть программы, имеющая свое назначение и реализующая некий алгоритм?
    a) Подпрограмма
    b) Массив
    c) Структура
    d) Файл

20. Как называется объединение в единое целое данных и алгоритмов обработки этих данных?
    a) Наследование
    b) Полиморфизм
    c) Полиформизм
    d) Инклюдирование
    e) Инкапсуляция

21. Какой вид цилка присутствует в данном фрагменте программы?

```
int i=0;
int y=0;
do
 {   i++;
    y=y+1/i;}
while (1/i<e);
```

    a) Цикл с предусловием;
    b) Цикл с постусловием;
    c) Цикл со счетчиком;
    d) Тут нет циклических алгоритмов;

22. Какой вид цилка присутствует в данном фрагменте программы?

```
y=1; k=0;
while (y<=M)
{
  y=y*3;
  k++;
}
```

    a) Цикл с предусловием;
    b) Цикл с постусловием;
    c) Цикл со счетчиком;
    d) Нет циклических конструкций;

23. Какие имена могут использоваться в качестве идентификаторов при написании программ в языке C++ (C#)?
    a) x1
    b) 1x
    c) for
    d) AAA

24. Как может выглядеть в языках С-семейства инкремент переменной x?
    a) x:=x+1
    b) x=x+1
    c) x++
    d) x=y+1
    e) x--
    f) ++x

25. Что позволяеть сделать команда continue в языках программирования С-семейства?
    a) Досрочно выйти из цикла
    b) Досрочно выйти из программы
    c) Выйти из текущей итерации цикла и продолжить цикл дальше;
    d) Начать цикл заново

26. Что будет являться атрибутами заданного класса?

```
class Complex
{ int real;
  int im;
  void Add(Complex x);
  void Mult(Complex x);}
```

    a) real
    b) Add
    c) im
    d) Mult

27. Какой доступ дает спецификация доступа protected языках C++/C#, примененная к элементам класса?
    a) Доступ к элементам только в этом классе
    b) Доступ к элементам из всех классов (общий)
    c) Доступ к элементам из производных классов
    d) Доступ к элементам из базового и производных классов

28. Что вычислит заданная функция?

```
int number(int a,int b)
 {int m;
   if (a>b) m=a;
else m=b;
   return m;}
```

    a) максимальное из 2-х чисел;
    b) Вычисляет минимальное из 2-х чисел;

29. Как производится нумерация элементов в массиве в С-семействе языков программирования?
    a) с 1
    b) по желанию программиста
    c) с 0

30. Какая спецификация доступа применяется к элементам класса, которые должны быть доступны в классе и вне класса?
    a) protected
    b) public
    c) private

Формируемые компетенции: ОК1, ОК2, ОК4, ОК5, ОК9, ОК10, ПК 1.3., ПК 1.4., ПК2.4

1. Какова причина синтаксических ошибок при написании программы?
    a) плохое знание языка программирования;
    b) ошибки в исходных данных;
    c) ошибки, допущенные на более ранних этапах;
    d) неправильное применение процедуры тестирования.

2. Когда можно обнаружить синтаксические ошибки в программе?
    a) при компиляции;
    b) при отладке;
    c) при тестировании;
    d) на этапе проектирования;
    e) при эксплуатации.

3. Могут ли проявиться ошибки в работе программы при изменении условий эксплуатации программы?
    a) да;
    b) нет.

4. Могут ли проявиться ошибки в работе программы при изменении в предметной области?
    a) да;
    b) нет.

5. Что представляет собой защитное программирование?
    a) встраивание в программу отладочных средств;
    b) создание задач защищенных от копирования;
    c) разделение доступа в программе;
    d) использование паролей;
    e) оформление авторских прав на программу.

6. Как называется ошибка с неправильным написанием служебных слов (операторов) при написании программы?
    a) синтаксическая;
    b) семантическая;
    c) логическая;
    d) символьная.

7. Как называется ошибка с неправильным использованием управляющих конструкций при написании программы?
    a) семантическая;
    b) синтаксическая;
    c) логическая;
    d) символьная.

8. Какая ошибка может быть при написании программы?
    a) синтаксические;
    b) орфографические;
    c) фонетические;

9. Как называется процедура поиска ошибки в программе, когда известно, что она есть?
    a) отладка;
    b) тестирование;
    c) компоновка;
    d) транзакция;
    e) трансляция.

10. Как называется программа для просмотра значений переменных при выполнении программы?
    a) отладчик;
    b) компилятор;
    c) интерпретатор;
    d) трассировка;
    e) тестирование.

11. Что прооедставляет солбой отладка программы?
    a) процедура поиска ошибок, когда известно, что ошибка есть;
    b) определение списка параметров;
    c) правило вызова процедур (функций);
    d) составление тестов для проверки работы программы.

12. Когда программист может проследить пошаговое выполнение команд программы?
    a) при трассировке;
    b) при тестировании;
    c) при компиляции;
    d) при выполнении программы;
    e) при компоновке.

13. На каком этапе создания программы могут появиться синтаксические ошибки в программе?
    a) программирование;
    b) проектирование;
    c) анализ требований;
    d) тестирование.

14. Существует ли различие между отладкой и тестированием программы?
    a) да;
    b) нет.

15. Чему нужно уделять больше времени, чтобы получить хорошую программу?
    a) тестированию;
    b) программированию;
    c) отладке;
    d) проектированию.

16. Что представляет собой трассировка программы?
    a) проверка пошагового выполнения программы;
    b) тестирование исходного кода;
    c) отладка модуля;
    d) составление блок-схемы алгоритма.

17. Что представляет собой локализация ошибки в программе?
    a) определение места возникновения ошибки;
    b) определение причин ошибки;
    c) обнаружение причин ошибки;
    d) исправление ошибки.

18. Каково назначение отладки программы?
    a) поиск причин существующих ошибок;
    b) поиск возможных ошибок;
    c) составление спецификаций;
    d) разработка алгоритма.

19. Выберите инструментальное средство, не предназначенные для отладки программы?

a) компиляторы;
b) отладчики;
c) трассировка.

20. Что представляет собой отладка программ?
a) локализация и исправление ошибок;
b) алгоритмизация программирования;
c) компиляция и компоновка.

21. Что выполняется раньше, автономная или комплексная отладка программы?
a) автономная;
b) комплексная.

22. Какие действия при отладке и тестировании указывают на наличие синтаксической ошибки в программе?
a) Программа не компилируется
b) Программа запускается, но работает неправильно
c) Программа компилируется, но не запускается

23. Какие действия при тестировании указывают на наличие логической ошибки в программе?
a) Программа не компилируется
b) Программа запускается, но работает неправильно
c) Программа компилируется, но не запускается

24. Какие действия при тестировании программы указывают на ошибку среды выполнения?
a) Программа не компилируется
b) Программа запускается, но работает неправильно
c) Программа компилируется, но не запускается

25. Что позволяет выполнить отладчик ПО?
a) отслеживать, устанавливать или изменять значения переменных в процессе выполнения кода
b) устанавливать и удалять контрольные точки
c) выполнять трассировку программы
d) производить оптимизацию кода
e) компилировать исходный код

26. Что проверятеся при комплексном тестировании программы?
a) согласованность работы отдельных частей программы;
b) правильность работы отдельных частей программы;
c) быстродействие программы;
d) эффективность программы.

27. Как называется процесс исполнения программы с целью обнаружения ошибок?
a) тестирование;
b) кодирование;
c) сопровождение;
d) проектирование.

28. Что представляет сообой автономное тестирование программы?
a) тестирование отдельных частей программы;
b) инструментальное средство отладки;
c) составление блок-схем;
d) пошаговая проверка выполнения программы.

29. Как называется просчитанный вручную пример выполнения программы от исходных данных до ожидаемых результатов расчета?

a) тест
b) отладчик
c) компиляция

30. Как называется процесс исследования, испытания программного продукта, имеющий своей целью проверку соответствия между реальным поведением программы и её ожидаемым поведением на конечном наборе тестов, выбранных определённым образом?
a) Тестированием
b) Оптимизацией
c) Отладкой

Формируемые компетенции: ОК1, ОК2, ОК4, ОК5, ОК9, ОК10, ПК 1.5., ПК2.5

1. Какой этап жизненного цикла программы не является обязательным?
a) оптимизация;
b) проектирование;
c) тестирование;
d) программирование;
e) анализ требований.

2. Существует ли связь между эффективностью и оптимизацией программы?
a) да;
b) нет.

3. Повышает ли качество программы оптимизация?
a) да;
b) нет.

4. Как называется нахождение наилучшего варианта из множества возможных вариантов программного кода?
a) оптимизация;
b) тестирование;
c) автоматизация;
d) отладка;
e) сопровождение.

5. Что представляет собой оптимизация программ?
a) улучшение работы существующей программы;
b) создание удобного интерфейса пользователя;
c) разработка модульной конструкции программы;
d) применение методов объектно-ориентированного программирования.

6. Какова основная цель оптимизации программы?
a) уменьшение время выполнения или размера требуемой памяти;
b) уменьшение размера программы;
c) независимость модулей;
d) качество программы, ее надежность.

7. Какой критерий оптимизации программы является основным?
a) эффективность использования ресурсов;
b) структурирование алгоритма;
c) структурирование программы.

8. Возможна ли оптимизация программ без участия программиста?
a) да;
b) нет.

9. Возможна ли оптимизация циклов?
   a) да;
   b) нет.

10. В чем заключатеся оптимизация циклов?
   a) уменьшении количества повторений тела цикла;
   b) просмотре задачи с другой стороны;
   c) упрощение задачи за счет включения логических операций.

11.Какова суть оптимизации программы?
   a) модификация кода;
   b) отладка;
   c) повышение сложности программы;

12. Каков результат оптимизации программы?
   a) эффективность;
   b) надежность;
   c) машино-независимость;
   d) мобильность.

13. Какова сущность оптимизации циклов?
   a) сокращение количества повторений тела цикла;
   b) сокращение тела цикла;
   c) представление циклов в виде блок-схем;
   d) трассировка циклов;
   e) поиск ошибок в циклах.

14. Каковы основные цели оптимизации програмсы?
   a) Уменьшение объема используемой программой оперативной памяти
   b) ускорение работы программы
   c) увеличение объема кода программы

15. Может ли случиться так, что для оптимизации работы программы часть кода переписывается на другом языке?
a) Да
b) Нет

16. Может ли оптимизация кода оказаться неэффективной?
   a) Да
   b) Нет

17. Может ли оптимизация кода оказаться нерентабельной?
   a) Да
   b) Нет

18. Существуют ли автоматические анализаторы кода?
   a) Да
   b) Нет

19. Тождественны ли понятия оптимизация кода и рефакторинг кода?
   a) Да
   b) Нет

20. Какой ввраиант цикла в коде поиска наличия четного числа в массиве является оптимальным?
   a) evenNumber = false;
      for (int i=0; i< count; i++)

```
if ((input[i] % 2) ==0)
{
        evenNumber = true;
        break;
}
```

b)
```
evenNumber = false;
for (int i=0; i< count; i++)
{
        if ((input[i] % 2) ==0)
        evenNumber = true;
}
```

21. Какая операция из тождественных будет более оптимальна?

a) sqrt(x) < sqrt(y)
b) x < y

22. Какие названия идентификаторов в программе допустимо использовать?
a) TotalSale;
b) Total Sale;
c) 1Sale;
d) Total;

23. Что называют стандартом кодирования?
a) набор правил и соглашений, используемых при написании исходного кода на некотором языке программирования.
b) Набор служебных слов языка программирования
c) Правила написания комментариев

24. Меняет ли рефакторинг кода поведение программы?

a) Да
b) Нет

25. Какие действия могут произойти при выполнении рефакторинга кода программы?

a) Добавление нового функционала
b) Исправление ошибок
c) Повышает читабельность кода

26. Каковы цели рефакторинга кода программы?

a) Повышение читаемости кода
b) Повышение понятности кода
c) Исправление ошибок
d) Изменение алгоритма программы

27. Какое именование переменной относится к стилю CamelCase?
a) ClientName
b) client_name
c) Client_name
d) Clientname

28. Какое именование переменной относится к стилю snake_case?
a) ClientName
b) client_name

c) Client_name
d) Clientname

29. Какое именование переменной относится к стилю CamelCase?
    a) NumberOrder
    b) number_order
    c) Number_order
    d) Nemberorder

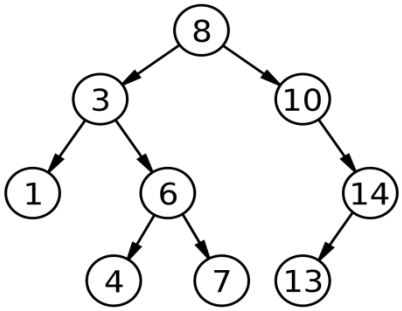
30. Какое именование переменной относится к стилю snake_case?
    a) NumberOrder
    b) number_order
    c) Number_order
    d) Nemberorder

Второй блок заданий – вопросы с ответом
Формируемые компетенции: ОК1, ОК2, ОК4, ОК5, ОК9, ОК10, ПК 1.1., ПК 1.2.

1. Каковы этапы жизненного цикла программного продукта?
2. Каковы основные принципы структурного программирования?
3. Какие вы знаете технологии программирования?
4. Что представляет собой программа, описанная в стиле модульного программирования?
5. Чем отличаются утилитарные программы от программных продуктов?
6. Каковы достоинства структурного программирования?
7. Каковы основные принципы объектно-ориентированного программирования?
8. Что такое инкапсуляция?
9. Что представляет собой полиморфизм?
10. Что представляет собою наследование?
11. Каковы преимущества разделения программы на подпрограммы (модули)?
12. Что такое подпрограмма?
13. Что представляет собой рекурсивная функция?
14. Что такое класс в концепции программы?
15. Каковы составляющие класса программы?
16. Чем объект отличается от класса?
17. Что такое тип данных в программировании?
18. Что такое стек?
19. Что такое очередь?
20. Какая структура данных изображена на рисунке?

21. Какая структура данных изображена на рисунке?

22. Какая структура данных изображена на рисунке?



23. Какая структура данных изображена на рисунке?

24. Даны три целых числа: A, B, C. Как можно записать в виде условного на языках C++/C# выражения следующее утверждение: «Двойное неравенство A < B < C»?

25. Даны три целых числа: A, B, C. Как можно записать на языках C++/C# в виде условия следующее утверждение: «Хотя бы одно из чисел A, B, C положительное»?

26. Какие спецификации доступа к членам класса на языках C++/C# Вы знаете?

27. Дайте определение конструктора класса. Каково назначение конструктора?

28. Сколько конструкторов может быть в классе? Какие виды конструкторов создаются по умолчанию?

29. Что выполнит данная программа?

```
#include <iostream>
using namespace std;
int length(char *s)
{int i;
  for(i=0; *s!='\0';s++)
    i++;
 return(i);
 }
int main()
{ char s[]="lalalalalalalalalala";
  cout<<length(s);
  return 0;}
```

30. Что выполнит данная программа?

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
   ifstream in_file("f.txt");
   char x;
   if (in_file.fail())
   {
      cout << "Error";
```

```
        exit(1);
    }
  int count1 = 0;
    int count2 = 0;
    while (in_file >> x)
    {
      if (x == '{') count1++;
      if (x == '}') count2++;
    }
    in_file.close();
    if (count1 == count2)
      cout << "Program good!";
  }
```

Формируемые компетенции: ОК1, ОК2, ОК4, ОК5, ОК9, ОК10, ПК 1.3., ПК 1.4., ПК2.4

1. Как называется просчитанный вручную пример выполнения программы от исходных данных до ожидаемых результатов расчета?
2. Как называеств пошаговое выполнение программы в режиме отладки?
3. Какие ключевые слова С++, которые используются для обработки исключений?
4. Что обозначается ключевым словом catch?
5. Что обозначается ключевым словом try?
6. Что представляет собой отладка программы?
7. Какие средства облегчают процесс отладки?
8. Что представляет собой аналитический способ обнаружения ошибки?
9. Что представляет собой экспериментальный способ обнаружения ошибки?
10. Что представляет собой отладочная печать?
11. Программа компилируется и работает, но выдает неправильный результат. Сама логика, на которой базируется вся программа, является ущербной. К какому типу ошибок относится подобная ошибка?
12. Иногда, синтаксис исходного кода может быть безупречным, но ошибка все же может произойти. Это может быть связано с проблемами в самом компиляторе. К какому типу ошибок относится подобная ошибка?
13. Какие действия можно отнести к силовым методам отладки?
14. Как можно исправить ошибку в функции вычисления среднего арифметического
```
float average(float a, float b)?
{
  return a + b / 2;
}
```
15. Что представляет собой исключительная ситуация?
16. Если в контролируемом блоке при выполнении программы формируется исключение, что произойдет?
17. Что представляет собой трассировка программы?
18. Что представляет собой точка останова?
19. Программа пошагового выполнения для просмотра значений переменных при выполнении программы называется:
20. В чем разница между отладкой и тестированием?
21. Какие стратегии тестирования Вы знаете?
22. Каким образом составляются тестовые наборы для ручного тестирования?
23. Что такое функциональное тестирование?
24. Что такое ручное тестирование?
25. Что такое автоматизированное тестирование?
26. Как называется стратегия тестирования функционального поведения программы, при которой нет доступа к исходному коду приложения?
27. Как называется тестирование внутренней структуры, дизайна и кодирования программного решения, в котором код виден тестеру?

28. Что такое дефект, или баг?
29. Имеется код:

```
Int a;
Cin>>a;
If (a%2==0)
{
Cout<<a<<”  четное ”;
}
```

Какое минимальное количество тестов нужно сделать, чтобы проверить конструкцию ветвления?
30. Имеется код:

```
Int a;
Cin>>a;
If (a%2==1)
{
Cout<<a<<”  нечетное ”;
}
```

Какое минимальное количество тестов нужно сделать, чтобы проверить конструкцию ветвления?

Формируемые компетенции: ОК1, ОК2, ОК4, ОК5, ОК9, ОК10, ПК 1.5., ПК2.5

1. Как определить «узкие» места программы, требующие оптимизации?
2. Что представляет собой оптимизация кода?
3. Каковы цели оптимизации?
4. Тождественны ли понятия оптимизация кода и рефакторинг кода?
5. Чем отличается оптимизация кода от рефакторинга?
6. Существуют ли автоматизированные средства оптимизации кода?
7. Могут ли автоматизированные средства оптимизации кода кардинально изменить первоначальный алгоритм?
8. Может ли оптимизация кода оказаться неэффективной?
9. Может ли оптимизация кода оказаться нерентабельной?
10. Какова сущность оптимизации циклов?
11. Какая из двух записей условия будет наиболее оптимальной и почему?
    not a and not b
    not (a or b)
12. Может ли операция может помочь оптимизировать цикл?
13. Дан алгоритм пузырьковой сортировки. Есть ли в нем приемы оптимизации кода?

```
ЦИКЛ ДЛЯ J=1 ДО N-1 ШАГ 1
 F=0
 ЦИКЛ ДЛЯ I=0 ДО N-1-J ШАГ 1
    ЕСЛИ A[I] > A[I+1] ТО
      ОБМЕН A[I],A[I+1]
      F=1
 ЕСЛИ F=0 ТО ВЫХОД ИЗ ЦИКЛА
```

14. В целях экономии памяти какие переменные следует предпочитать в программе, глобальные или локальные?
15. Влияют ли характеристики вычислительной машины (например, количество и тактовая частота процессорных ядер, размер кэша, пропускная способность системной шины, объём оперативной памяти) на оптимизацию программы?
16. Как можно оптимизировать код, если два соседних цикла повторяются одно и то же число раз и не влияют друг на друга?
17. Каковы минусы оптимизации?
18. Может ли оптимизация привести к появлению ошибок?
19. Нужно ли оптимизировать весь код программы?
20. При написании программ в целях оптимизации предпочтительнее использовать стандартные библиотечные функции или писать вместо них свои?

21. Данный код заменяет прописные буквы на строчные:

```
void lower(char *s) {
   for (int i = 0; i < strlen(s); i++)
      if (s[i] >= 'A' && s[i] <= 'Z')
         s[i] -= ('A' - 'a');
}
```

Что существенно замедляет работу этой функции?

22. Какая операция выполнится быстрее, деление на 2 или битовый сдвиг вправо?

23. Арифметические операции выполняется быстрее с целыми или вещественными числами?

24. Какие существуют правила именования идентификаторов?

25. Что называют стандартом кодирования?

26. Как меняет ли рефакторинг кода поведение программы?

27. Какие действия могут произойти при выполнении рефакторинга кода программы?

28. К какому стилю относится именование переменной ClientName?

29. К какому стилю относится именование переменной client_name?

30. К какому стилю относится именование переменной NumberOrder?

Составитель: Мохнач О.А.