

Перечень вопросов
по оценке сформированности компетенций образовательной программы
09.03.02 «Информационные системы и технологии»
по дисциплине «Технологии программирования»

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

Раздел 1. Методы и приемы программирования на C++

1. Этапы разработки программ на C/C++. Средства разработки (компиляторы, инструментальные среды разработки).
2. Основные понятия языка программирования: состав языка.
3. Типы данных C++. Концепция типа данных. Основные типы данных.
4. Операторы, переменные и выражения. Структура программы.
5. Организация ввода/вывода. Функции консольного ввода/вывода.
6. Условные операторы.
7. Оператор-переключатель.
8. Циклические операторы. Операторы передачи управления.
9. Одномерные и двумерные числовые массивы.
10. Массивы и указатели.
11. Символьные массивы.
12. Обработка символьных массивов.
13. Понятия модульного программирования: функции, директивы, файловая структура.
14. Объявление и определение функций. Области действия идентификаторов.
15. Файлы. Организация файлового (потокowego) ввода/вывода.
16. Строки и тексты как массивы символов. Функции обработки строк.

Раздел 2. Основные базовые понятия объектно-ориентированного программирования

17. Типы данных, определенные пользователем: переименование, перечисление. Структурированные типы данных.
18. Тип данных структура в языке C++. Объявление типа, переменной, доступ к полям структуры.
19. Массив структур в языке C++. Передача в функцию массива структур.
20. Динамические структуры данных. Линейные списки.
21. Парадигма объектно-ориентированного программирования.
22. Понятие класса и метода. Инкапсуляция.
23. Объявление классов и создание объектов.
24. Доступ к членам классов.
25. Конструкторы. Назначение и технология использования.
26. Деструкторы. Назначение и использование.
27. Передача объектов функциям.
28. Дружественные функции.
29. Иерархия классов.
30. Наследование. Технология доступа к членам классов.
31. Защищенные члены классов.
32. Правила наследования защищенных членов классов.
33. Технология использования конструкторов в наследовании.
34. Технология использования деструкторов в наследовании.
35. Понятие полиморфизма в C++.
36. Технология перегрузки функций.

37. Перегрузка операторов и операций.
38. Наследование. Ключи доступа.
39. Простое наследование.
40. Виртуальные методы.
41. Множественное наследование.

Раздел 3. Технология конструирования программ

42. Технология программирования и основные этапы её развития.
43. Блочный-иерархический подход к созданию сложных систем.
44. Жизненный цикл и этапы разработки программного обеспечения.
45. Модели жизненного цикла программного обеспечения.
46. Ускорение разработки программного обеспечения.
47. Оценка качества процессов создания программного обеспечения.
48. Спецификации программ. Тестирование программ.
49. Особенности процесса синтеза программных систем.
50. Особенности архитектурного этапа проектирования.
51. Структурирование программной системы.
52. Архитектура с хранилищем данных.
53. Клиент-серверная архитектура.
54. Многоуровневая архитектура

ОПК-6. Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий

Типовые обобщенные практические задания

1. Вычислить и вывести на экран в виде таблицы значения функции $f(x)$ на интервале $[a,b]$ с шагом h .
2. Вычислить и вывести на экран сумму (разность, произведение, среднее значение) заданных элементов массива.
3. Вычислить и вывести на экран отсортированные элементы массива в заданном порядке.
4. Описать структуру с заданным именем и выполнить указанные действия над ее элементами: ввод, поиск, сортировка, вывод в указанном виде.
5. Написать функцию, которая записывает в бинарный файл (считывает из файла) указанные данные.
6. Написать программу обрабатывающую текстовые данные в виде текста, хранящиеся в бинарном текстовом файле: вывод на экран, поиск и действия над элементами текста.
7. Написать программу, в которой описание структуры размещается в заголовочном файле, а определения функций и главной функции в отдельных файлах
8. Составить программу, которая обеспечивает указанные операции над линейным списком: ввод/вывод данных, контроль ошибок при вводе, диалоговое меню.
9. Спроектировать класс, содержащий две закрытые переменные и функции доступа к ним, конструктор по умолчанию, конструктор с параметрами и конструктор копирования, дружественную функцию, выводящую на экран сумму значений закрытых переменных. Создать объекты и продемонстрировать работу функций
10. Спроектировать два класса, один из которых наследует другой по типу `public (private)`. Закрытой переменной в первом классе является указатель на символьную строку, а во втором – целочисленная переменная. Функциями класса являются конструкторы без параметров и с параметрами, деструкторы. Разработать виртуальную функцию, выводящую на экран значения переменных. Создать по объекту в каждом классе и продемонстрировать работу функций.
11. Спроектировать класс для работы с символьными строками. Состав класса:
 - две закрытые переменные - указатель на символьную строку (`char *s`) и размер строки (`int n`);
 - три конструктора: первый – без параметров, второй – с параметрами, третий – для создания копии объекта;

- функции доступа к закрытым членам класса: возвращение размера строки **length()**, извлечение символа с указанным индексом **at()**, установка символа строки **setchar()**;
- функции работы со строками: определение длины строки **strlen()**, удаления фрагмента строки **erase()**, вставка фрагмента в строку **insert()**, добавления фрагмента в конец строки **append()**;
- оператор-функции для перегрузки: операции присваивания (=) – для копирования строк, операции «+» - для «склеивания» объектов типа String;
- оператор-функции перегрузки операторов консольного вывода и ввода объектов класса.

Создать необходимое количество объектов и продемонстрировать работу спроектированных средств при использовании объектов.