Вопросы к экзамену.

Контрольные вопросы к промежуточной аттестации:

1. Динамическое выделение и перераспределение памяти

2. Операторы управления динамической памятью (в С++)

3. Структуры данных. Линейные списки.

4. Краткий обзор С++. Объектно-ориентированное программирование (ООП). Понятие объект.

5. Фундаментальные понятия ООП. Класс

6. Полиморфизм

7. Наследование

8. Консольный ввод-вывод в С++

9. Перегрузка функций

10. Введение в классы. Конструкторы и деструкторы. Конструкторы с параметрами

11. Указатели на объект

12. Присваивание объектов

13. Дружественные функции

14. Передача объектов функции

15. Объекты в качестве возвращаемого значения функции

16. Ссылки

17. Преимущества параметров-ссылок по сравнению с параметрами-указателями

18. Ссылка в качестве возвращаемого значения

19. Перегрузка конструкторов

20. Создание и использование конструктора копий

21. Перегрузка операторов

22. Перегрузка бинарных операторов

23. Перегрузка операторов отношения и логических операторов

24. Перегрузка унарных операторов

25. Дружественные операторы функции

26. Особенности использования оператора присваивания

27. Перегрузка оператора индекса массива

28. Наследование. Управление доступом к базовому классу.

29. Защищенные члены класса

30. Конструкторы, деструкторы и наследование. Синтаксис передачи аргументов из производного в базовый класс

31. Множественное наследование

32. Виртуальные базовые классы

33. Указатели на производные типы

34. Виртуальные функции. Отличие переопределения виртуальных функций в производных классах

35. Вызов виртуальной функции ссылкой на объект базового класса. Атрибут virtual наследуется

36. Потоки и файлы

37. Текстовые потоки

38. Бинарные потоки

39. Файлы

40. Указатель файла

41. Открытие файла

42. Закрытие файла

43. Запись символа

44. Чтение символа

45. Примеры использования функций: fopen(), fgetc(), fputc(), fclose()

46. Библиотека STL. Основные понятия и классы. Методы работы в STL

47. История создания кроссплатформенных систем программирования.

48. Концепции кроссплатформенного программного обеспечения

49. Характеристики существующих платформ разработки и области их применения

50. Современное состояние и принципиальные возможности языков программирования Java, C#, C++, python;

51. Объектно-ориентированное программирование как доминирующая парадигма программирования.

52. Введение в язык Java. История. Основные направления Java.

53. Синтаксис языка Java. Создание простой программы. Основы GUI. События в Java.

54. JDK и JRE

55. Особенности архитектуры .NET. Среды разработки для Net.

56. Методы и средства разработки приложений с использованием Visual Studio.NET.

57. Синтаксис языка C#.

58. Типы данных в Java, C#, C++, Python.

59. Абстракции и объекты.

60. Абстрактные классы.

61. Интерфейсы.

62. Управление доступом.

63. Инкапуляция.

64. Наследование и полиморфизм.

65. Коллекции объектов.

66. Обработка ошибок и исключения.

67. Внутренние и анонимные (безымянные) внутренние классы.

68. Система ввода-вывода.

69. События и их обработка

70. Концепция Model-View-Controller.

71. Портируемость Python.

72. Ссылочные и атомарные объекты.

73. Объектно-ориентированное и функциональное программирование в Python.

74. Модули в языке Python. Модули, написанные на «чистом» Python, и модули расширения (extension modules), написанные на других языках программирования.

75. Стандартная библиотека Python.

76. Взаимодействие с API

77. Платформа Microsoft .NET Framework 4.0. CLR, FCL, JIT compiler. Компиляция и выполнение программы в среде CLR.

78. Встроенные типы языка C#. Создание пользовательских типов. Значимые и ссылочные типы.

79. Состав языка C#. Правила именования идентификаторов рекомендации. Объектно-ориентированное программирование в C#.

80. Создание перечислимых типов. Сравнение ссылочных и значимых типов. Передача ссылочных и значимых типов в подпрограммы. Особенности.

81. Виды массивов в С#. Инициализация массивов. Выделение памяти. Класс Array.

82. Способы представления строк в C#. Преимущества и недостатки каждого способа.

83. Массивы как коллекции. Использование цикла foreach для работы с массивами

84. Перегрузка методов в языке C#. Рекурсивные методы. Методы с переменным числом аргументов.39. Индексаторы. Определение индексаторов для своего класса. Перегрузка других (унарных и бинарных) операций. Примеры перегрузки.

85. Концепция деструкторов в C#. Перегрузка деструкторов. Конструкторы и перегрузка конструкторов. Статические и динамические конструкторы.

Контрольные вопросы к текущему контролю:

Контрольная точка № 1:

1. Динамическое выделение и перераспределение памяти

2. Операторы управления динамической памятью (в С++)

3. Структуры данных. Линейные списки.

4. Краткий обзор С++. Объектно-ориентированное программирование (ООП). Понятие объект.

5. Фундаментальные понятия ООП. Класс

6. Полиморфизм

7. Наследование

8. Консольный ввод-вывод в С++

9. Перегрузка функций

10. Введение в классы. Конструкторы и деструкторы. Конструкторы с параметрами

11. Указатели на объект

12. Присваивание объектов

13. Дружественные функции

14. Передача объектов функции

15. Объекты в качестве возвращаемого значения функции

16. Ссылки

17. Преимущества параметров-ссылок по сравнению с параметрами-указателями

18. Ссылка в качестве возвращаемого значения

19. Перегрузка конструкторов

20. Создание и использование конструктора копий

21. Перегрузка операторов

22. Перегрузка бинарных операторов

23. Перегрузка операторов отношения и логических операторов

24. Перегрузка унарных операторов

25. Дружественные операторы функции

26. Особенности использования оператора присваивания

27. Перегрузка оператора индекса массива

28. Наследование. Управление доступом к базовому классу.

29. Защищенные члены класса

30. Конструкторы, деструкторы и наследование. Синтаксис передачи аргументов из производного в базовый класс

31. Множественное наследование

32. Виртуальные базовые классы

33. Указатели на производные типы

34. Виртуальные функции. Отличие переопределения виртуальных функций в производных классах

35. Вызов виртуальной функции ссылкой на объект базового класса. Атрибут virtual наследуется

36. Потоки и файлы

37. Текстовые потоки

38. Бинарные потоки

39. Файлы

40. Указатель файла

41. Открытие файла

42. Закрытие файла

43. Запись символа

Контрольная точка № 2:

44. Чтение символа

45. Примеры использования функций: fopen(), fgetc(), fputc(), fclose()

46. Библиотека STL. Основные понятия и классы. Методы работы в STL

47. История создания кроссплатформенных систем программирования.

48. Концепции кроссплатформенного программного обеспечения

49. Характеристики существующих платформ разработки и области их применения

50. Современное состояние и принципиальные возможности языков программирования Java, C#, C++, python;

51. Объектно-ориентированное программирование как доминирующая парадигма программирования.

52. Введение в язык Java. История. Основные направления Java.

53. Синтаксис языка Java. Создание простой программы. Основы GUI. События в Java.

54. JDK и JRE

55. Особенности архитектуры .NET. Среды разработки для Net.

56. Методы и средства разработки приложений с использованием Visual Studio.NET.

57. Синтаксис языка C#.

58. Типы данных в Java, C#, C++, Python.

59. Абстракции и объекты.

60. Абстрактные классы.

61. Интерфейсы.

62. Управление доступом.

63. Инкапуляция.

64. Наследование и полиморфизм.

65. Коллекции объектов.

66. Обработка ошибок и исключения.

67. Внутренние и анонимные (безымянные) внутренние классы.

68. Система ввода-вывода.

69. События и их обработка

70. Концепция Model-View-Controller.

71. Портируемость Python.

72. Ссылочные и атомарные объекты.

73. Объектно-ориентированное и функциональное программирование в Python.

74. Модули в языке Python. Модули, написанные на «чистом» Python, и модули расширения (extension modules), написанные на других языках программирования.

75. Стандартная библиотека Python.

76. Взаимодействие с API

77. Платформа Microsoft .NET Framework 4.0. CLR, FCL, JIT compiler. Компиляция и выполнение программы в среде CLR.

78. Встроенные типы языка C#. Создание пользовательских типов. Значимые и ссылочные типы.

79. Состав языка C#. Правила именования идентификаторов рекомендации. Объектно-ориентированное программирование в C#.

80. Создание перечислимых типов. Сравнение ссылочных и значимых типов. Передача ссылочных и значимых типов в подпрограммы. Особенности.

81. Виды массивов в С#. Инициализация массивов. Выделение памяти. Класс Array.

82. Способы представления строк в C#. Преимущества и недостатки каждого способа.

83. Массивы как коллекции. Использование цикла foreach для работы с массивами

84. Перегрузка методов в языке C#. Рекурсивные методы. Методы с переменным числом аргументов.39. Индексаторы. Определение индексаторов для своего класса. Перегрузка других (унарных и бинарных) операций. Примеры перегрузки.

85. Концепция деструкторов в C#. Перегрузка деструкторов. Конструкторы и перегрузка конструкторов. Статические и динамические конструкторы.