Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новосибирский государственный технический университет»
Кафедра автоматизированных систем управления
Кафедра автоматики
Кафедра вычислительной техники

Паспорт экзамена

по дисциплине «Программирование», 2 семестр

1. Методика оценки

Экзамен проводится в устной форме по билетам. Билет формируется по следующему правилу:

1. Теоретическая часть – в виде 10 тестовых вопросов из разных тем (темы – из списка экзаменационные вопросов, тестовые вопросы выбираются случайным образом из закрытого списка. На каждый тестовый вопрос дается короткий письменный ответ 1-2 абзаца. Тест оценивается 0..2 балла, суммарно 0..20.
2. Задача – выбирается из списка. Оценка 0..20 баллов. Задача реализуется в виде компьютерной программы, исполняемой на тестовых данных.

Таким образом, проверяются результаты обучения по дисциплине, соотнесенные с установленными в программе индикаторами достижения компетенций.

На экзамене преподаватель вправе задавать студенту уточняющие и дополнительные вопросы из общего перечня (п. 4).

Форма экзаменационного билета

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
Факультет АВТФ

Билет № _____
к экзамену по дисциплине «Программирование»

Тестовый вопрос 1 ----------
Тема: Вопрос 20. Двоичный файл. Модель двоичного файла. Последовательный и произвольный доступ

8. Cprog. Вопросы без ответов. 9.5. (Закладка ПРОГРАММЫ, 95-17). Тестовый вопрос 5. В каком формате читаются (пишутся) данные в двоичный файл? Опишите структуру данных в двоичном файле (4 балла).

Тестовый вопрос 2 ----------
Тема: Вопрос 1. Тип данных и переменная. Базовые и производные типы данных. Иерархия определений типов данных и вложенности компонент переменных. Контекстный способ определения типа данных в Си. Абстрактный тип данных. Спецификация typedef.

13. Выберите синтаксически допустимые выражения для определяемых типов данных и переменных
int f(char *)[20];

_____

int FF(char *s){ return 0; } f[0]=FF;
for(int i=0;i<20;i++) f[i]("aaa");
недопустимый ТД

Тестовый вопрос 3 ----------
Тема: Вопрос 5. Адресная арифметика. Указатели и массивы. Способы работы через указатель с массивом.

14. Cprog. Вопросы без ответов. 5.2. (Закладка ПРОГРАММЫ, 52-09). Тестовый вопрос 11. Результат функции, комментарии, вызов на статических данных, назначение указателя и смысл операций над ним в Тестовый вопросе (3 балла).

Тестовый вопрос 4 ----------
Тема: Вопрос 20. Двоичный файл. Модель двоичного файла. Последовательный и произвольный доступ

7. Определите термин «указатель в двоичном файле». Какой тип данных используется для его представления?

Тестовый вопрос 5 ----------
Тема: Вопрос 5. Адресная арифметика. Указатели и массивы. Способы работы через указатель с массивом.

4. Что произойдет при выполнении фрагмента (ошибка компиляции, исполнения, непредсказуемое поведение программы)? Какой результат можно ожидать, исходя из правил адресной арифметики?
int A[10]; A [-1]=0; A[12]=0;

Тестовый вопрос 6 ----------
Тема: Вопрос 14. Деревья. Способы представления деревьев. Полный рекурсивный обход дерева (для всех способов представления). Алгоритмы, основанные на полном рекурсивном обходе дерева.

23. Cprog. Вопросы без ответов. 8.5. Тестовый вопрос 23. Для примера вызова и инициализатора (закладка ПРОГРАММЫ 85-10): способ представления дерева, содержательный результат рекурсивной функции. Изобразить дерево, соответствующее инициализатору (предок-потомки), прокомментировать текст функции (3 балла)

Тестовый вопрос 7 ----------
Тема: Вопрос 9. Двусвязные (циклические) списки. Представление очереди на циклическом списке.

9. Cprog. Вопросы без ответов. 6.3. (Закладка ПРОГРАММЫ, 63-11). Тестовый вопрос 6. Результат функции, комментарии, смысл указателей, вызов на статических данных (4 балла).

Тестовый вопрос 8 ----------
Тема: Вопрос 20. Двоичный файл. Модель двоичного файла. Последовательный и произвольный доступ

3. Какие параметры используют функции fread/fwrite и как они соотносятся с моделью двоичного файла, с указателями и адресной арифметикой в Си?

Тестовый вопрос 9 ----------
Тема: Вопрос 19. Работа с памятью на физическом уровне. Преобразование типов указателя. Упаковка данных произвольного формата.

1. Что происходит с адресом, содержащемся в указателе, при смене его типа?

Тестовый вопрос 10 ----------
Тема: Вопрос 14. Деревья. Способы представления деревьев. Полный рекурсивный обход дерева (для всех способов представления). Алгоритмы, основанные на полном рекурсивном обходе дерева.

8. Cprog. Вопросы без ответов. 8.5. Тестовый вопрос 8. Для примера вызова и инициализатора (закладка ПРОГРАММЫ 85-10): способ представления дерева, содержательный результат рекурсивной функции. Изобразить дерево, соответствующее инициализатору (предок-потомки), прокомментировать текст функции (3 балла)

Утверждаю: зав. кафедрой ____ _____ должность, ФИО
(подпись)
(дата)

2. Критерии оценки результатов обучения, соотнесенных с уровнями освоения индикаторов достижения компетенций

Ответ на экзаменационный билет (тест) засчитывается на **продвинутом** уровне, если студент проводит сравнительный комплексный анализ материала, выявляет проблемы, предлагает механизмы их решения, представляет количественные характеристики определенных процессов, приводит конкретные примеры, не допускает ошибок и способен обосновать выбор метода решения задачи Совокупность результатов обучения по дисциплине (модулю) и соотнесенных с ними индикаторов достижения компетенций не содержит пробелов. Установленные в программе компетенции сформированы в полном объеме. Оценка составляет *от 35 до 40 баллов.*

Ответ на экзаменационный билет (тест) засчитывается на **базовом** уровне, если студент при ответе на вопросы формулирует основные понятия, дает характеристику процессов, явлений, проводит анализ причин, условий, способен представить количественные и качественные характеристики процессов, не допускает существенных ошибок при решении задачи Совокупность результатов обучения по дисциплине (модулю) и соотнесенных с ними индикаторов достижения компетенций содержит несущественные пробелы. Установленные в программе компетенции сформированы на базовом уровне. Оценка составляет *от 34 до 27 баллов.*

Ответ на экзаменационный билет (тест) засчитывается на **пороговом** уровне, если студент при ответе на вопросы дает определение основных понятий, может показать причинно-следственные связи явлений, при решении задачи допускает непринципиальные ошибки, например, вычислительные. Совокупность результатов обучения по дисциплине (модулю) и соотнесенных с ними индикаторов достижения компетенций содержит пробелы. Установленные в программе компетенции сформированы на пороговом уровне. Оценка составляет *от 26 до 20 баллов.*

Ответ на экзаменационный билет (тест) считается **неудовлетворительным** (ниже порогового уровня), если студент при ответе на вопросы не дает определений основных

понятий, не способен показать причинно-следственные связи явлений, при решении задачи допускает принципиальные ошибки. Совокупность результатов обучения по дисциплине (модулю) и соотнесенных с ними индикаторов достижения компетенций содержит существенные пробелы. Установленные в программе компетенции не сформированы. Оценка составляет *менее 20 баллов.*

3. Шкала оценки

Экзамен считается сданным, если сумма баллов по всем заданиям билета составляет от 20 до 40 баллов включительно. Сумма менее 20 баллов признается неудовлетворительным результатом промежуточной аттестации по дисциплине

В общей оценке по дисциплине экзаменационные баллы учитываются в соответствии с правилами балльно-рейтинговой системы, установленными в НГТУ.

4. **Вопросы к** экзамену **по дисциплине «**Программирование**»**

Экзаменационные вопросы (20 баллов)

1. Тип данных и переменная. Базовые и производные типы данных. Иерархия определений типов данных и вложенности компонент переменных. Контекстный способ определения типа данных в Си. Абстрактный тип данных. Спецификация **typedef**.
2. Модульная организация программы. Время жизни и область действия переменных. Классификация. Определение и объявление переменных. Внешние, автоматические и статические переменные. Область действия функций.
3. Модульное программирование. Статическое связывание. Библиотеки. Заголовочные файлы, их назначение и содержание. Файл проекта в классическом программировании.
4. Указатели. Указатель как элемент архитектуры компьютера. Синтаксис указателя в Си. Указатель и ссылка. Передача формальных параметров и результата по значению и по ссылке.
5. Адресная арифметика. Указатели и массивы. Способы работы через указатель с массивом.
6. Динамическая память. Динамические переменные и массивы. Операторы и функции управления динамической памятью.
7. Массивы указателей. Способы формирования массивов указателей - статические, динамические, смешанные. Работа с массивами указателей.
8. Списки. Определение элемента списка. Способы формирования списков. Односвязные списки. Представление стека и очереди на односвязном списке.
9. Двусвязные (циклические) списки. Представление очереди на циклическом списке.
10. Рекурсия. Рекурсивная структура данных и функция. Реализация рекурсивных функций, роль стека. Инвариант рекурсивной функции. Особенности разработки рекурсивных алгоритмов. Смысл локальных и глобальных переменных, формальных и фактических параметров в рекурсивной функции. Способы накопления результата.
11. Рекурсивные алгоритмы сортировки, трудоемкость.
12. Рекурсивные алгоритмы комбинаторного перебора. Генерация подмножеств заданного множества. Генерация последовательностей (цепочек) заданного множества

13. Рекурсивные поисковые алгоритмы. Сокращение пространства перебора. Жадные алгоритмы.
14. Деревья. Способы представления деревьев. Полный рекурсивный обход дерева (для всех способов представления). Алгоритмы, основанные на полном рекурсивном обходе дерева.
15. Двоичное дерево. Основные характеристики и алгоритмы. Балансировка дерева.
16. Дерево, упорядоченное по вертикали. Пирамидальная сортировка.
17. Указатель на функцию. Указатель на функцию - формальный параметр. Динамическое связывание.
18. Машинно-ориентированные (поразрядные) операции. Их интерпретация.
19. Работа с памятью на физическом уровне. Преобразование типов указателя. Упаковка данных произвольного формата.
20. Двоичный файл. Модель двоичного файла. Последовательный и произвольный доступ

Экзаменационные задачи (20 баллов)

1. Задачу реализовать в виде функции, получающей все данные через параметры.
2. В main создать две статические структуры данных, вызвать функцию 2 раза и вывести 2 результата. Никакие входные данные не вводятся, программа запускается «от одного клика». Структуру данных можно создавать динамически с помощью функции добавления элемента и ее многократного вызова.
3. Если функция создает структуру данных, то необходима функция трассировки – вывод значений, хранимых в структуре данных, и необходимых данных о текущей конфигурации (уровень вершины дерева, номер элемента списка и т.п.)

1. Функция получает массив размерности N и возвращает массив указателей на 10 односвязных списков. В i-ый список помещаются значения из массива с младшей цифрой, равной i;

2.Функция получает массив размерности N и возвращает односвязный список. Каждый элемент списка содержит уникальное значение от 0 до 9 и заголовок односвязного списка со значениями из массива, оканчивающимися на эту цифру.

3. Функция получает односвязный список и разделяет на два списка с четными и нечетными значениями.

4. Функция получает односвязный список, вычисляет среднее арифметическое значений и разделяет на три списка с меньшими, большими и равными значениями.

5. Функция получает двусвязный список и разделяет на два списка с четными и нечетными значениями и возвращает их.

6. Функция получает двусвязный список, вычисляет среднее между минимальным и максимальным и разделяет на три списка с меньшими, большими и равными значениями.

7. Элемент односвязного списка содержит множитель и количество его повторений. Функция получает число, раскладывает на множители и возвращает список со значениями <множитель,кол-во повторений>

8. Функция получает односвязный список, разрезает его на N списков и возвращает массив указателей на них.

9. Функция получает двусвязный список, разрезает его на N списков и возвращает массив указателей на них.

10. Функция получает циклический список, движется по кольцу и вырезает каждый N-ый

элемент, помещая в новый список, пока список не станет пустым (протестировать при N=0).

11. Сортировка односвязного списка выбором. Выбирается максимальный элемент, вырезается из списка и помещается в начало выходного.

12. Сортировка двусвязного списка выбором. Выбирается максимальный элемент, вырезается из списка и помещается в начало выходного.

13. Элемент односвязного списка содержит массив целых, их текущее количество и сумму. Создается список из N пустых элементов. Очередное значение из массива добавляется в элемент списка с минимальной суммой.

14. Функция получает односвязный список, находит максимальное значение V, удаляет элемент и вставляет в список элементы со значениями V/2 и v-V/2. Повторяет процесс N раз.

15. Функция получает двусвязный список, находит максимальное значение V, удаляет элемент и вставляет в список элементы со значениями V/2 и v-V/2. Повторяет процесс N раз.

16. Функция получает односвязный список, находит максимальное и минимальное значение, удаляет элементы и вставляет в список элемент со средним значением минимума и максимума. Повторяет процесс N раз.

17. Функция получает двусвязный список, находит максимальное и минимальное значение, удаляет элементы и вставляет в список элемент со средним значением минимума и максимума. Повторяет процесс N раз.

18. Элемент односвязного списка содержит массив целых фиксированной размерности. Функция создает структуру данных, читает из входного массива значения, пока массив не кончится. В каждом элементе списка запоминается текущее количество заполненных элементов массива.

19. Элемент двусвязного списка содержит массив целых фиксированной размерности. Функция создает структуру данных, читает из входного массива значения, пока массив не кончится. В каждом элементе списка запоминается текущее количество заполненных элементов массива

20. Элемент двусвязного циклического списка содержит массив целых фиксированной размерности. Функция создает структуру данных, читает из входного массива значения, пока массив не кончится. В каждом элементе списка запоминается текущее количество заполненных элементов массива

21. Элемент односвязного списка содержит заголовок односвязного списка (двухуровневый список). Размерность списка нижнего уровня ограничена.Функция создает структуру данных, читает из входного массива значения, пока массив не кончится.

22. Массив указателей содержит заголовки односвязных списков. Размерность списка ограничена. Функция создает структуру данных, читает из входного массива значения, пока входной массив не кончится. Размерность списка задается параметром