Организация экзамена по курсу программирования: 1-й семестр 2011/12 уч. года (вечернее обучение)

Экзамен проводится в форме компьютерного теста. Студенту предлагается одна задача. Требуется разработать программный проект в соответствии с требованиями задания. На выполнение задания студенту дается 90 минут (2 ак. часа). Оценка выставляется в соответствии со следующими правилами:

Состояние проекта	**Оценка**
Программный проект представляет собой работающую программу, полностью реализующую заявленную функциональность. Проект реализован в соответствии с принципами модульности, функционально-иерархической декомпозиции. Определены необходимые пользовательские типы (структуры или классы). Правильно использованы средства стандартной библиотеки C/C++. Обеспечена обработка возможных ошибок, возникающих в ходе работы программы. Проект характеризует хороший стиль программного кода. Студент в ходе собеседования может обосновать принятые проектные решения.	отлично
Программный проект представляет собой работающую программу, реализующую большую часть заявленной функциональности. Проект реализован в соответствии с принципами модульности, функционально-иерархической декомпозиции. Частично обеспечена обработка ошибок, возникающих в ходе работы программы. Студент в ходе собеседования может обосновать принятые проектные решения.	хорошо
Программный проект представляет собой работающую программу, реализующую большую часть заявленной функциональности. Имеются проблемы с применением концепций модульности и функционально-иерархической декомпозиции. Имеются замечания к стилю программного кода. Студент в ходе собеседования может обосновать основные проектные решения.	удовлетворительно
Программный проект не удается исполнить, однако рассмотрение проекта позволяет заключить, что разработчик следовал принципам модульности и функционально-иерархической декомпозиции. К большей части программного кода нет серьезных замечаний. Студент в ходе собеседования может обосновать основные проектные решения.	удовлетворительно
Программный проект не реализует поставленную задачу. Студент затрудняется в объяснении проектных решений.	неудовлетворительно

Далее приводятся основные темы, изученные в ходе семестра, и примеры задач.

Зав. кафедрой _____

Основные темы, рассмотренные в семестре 1

1. Представление данных в памяти компьютера. Системы счисления. Преобразования представлений.
2. Система типов языка высокого уровня. Операции над объектами данных встроенных типов. Выражения. Выбор подходящих типов для решаемой задачи.
3. Определение действий над объектами данных с использованием основных управляющих конструкций языка высокого уровня (последовательные вычисления, ветвления, циклы).
4. Работа с внешними файлами: ввод и вывод данных средствами библиотек C/C++.
5. Аргументы командной строки.
6. Функционально-иерархическая декомпозиция. Объявление и определение функций. Связывание аргументов и параметров. Представление вычислительного процесса как набора взаимодействующих функций.
7. Модульная структура проекта. Области видимости и классы памяти программных объектов. Раздельная компиляция модулей проекта.
8. Работа с массивами объектов. Создание массива, обращение к элементам массива, размещение массива в динамической памяти.
9. Определение и использование структурных типов. Разработка функций, обеспечивающих обработку структурных объектов. Принятие решений об организации модульной структуры проекта.
10. Умение выбрать правильные имена программных объектов. Разработка программы, которую может понять другой разработчик.
11. Объектно-ориентированный проект. Понятие об абстрагировании и инкапсуляции. Проектирование простых классов.
12. Определение класса. Реализация методов класса. Взаимодействие объектов класса.

Зав. кафедрой _____

Примеры экзаменационных задач

Задача 1

В файле “timetable.txt” содержится расписание движения поезда в виде последовательности строк в следующем формате:

<название-станции><пробел><время-прибытия><пробел><время-отправления>

Для начальной станции указывается только время отправления, для конечной станции – только время прибытия. Для простоты считаем, что поезд двигается в пределах одних суток, а названия станций состоят из одного слова. Время прибытия и время отправления задается в виде пары целых значений, разделенных двоеточием.

Необходимо обеспечить контроль корректности значений времени.

Программа для задаваемых пользователем двух станций вычисляет время в пути, если такое путешествие возможно на данном поезде. В противном случае, программа выдает сообщение, указывающее на причину невозможности вычисления пути. Результат вычислений выводится на консоль и дублируется в выходном файле “found.txt”.

Требования к интерфейсу программы. Программа выводит на консоль перенумерованный список возможных станций. Пользователь вводит номера станций, для которых производятся соответствующие вычисления.

Пример входного файла “timetable.txt”:

```
Петербург 8:22
Окуловка 9:25 9:30
Бологое 10:58 11:01
Тверь 11:31 11:40
Москва 12:45
```

Примерный сценарий взаимодействия с программой выглядит следующим образом:

```
Вычисление времени в пути. Список станций:

1. Петербург
2. Окуловка
3. Бологое
4. Тверь
5. Москва

Введите номера станций: 2 4

Время в пути от ст. Окуловка до ст. Тверь составляет 2:01
```

Зав. кафедрой _____

Задача 2

В файле “students.txt” содержится список студентов и оценок, полученных на экзаменах, в виде последовательности строк в следующем формате:

<фамилия ><пробел><число-экзаменов>[<пробел><оценка >]…

Число оценок должно соответствовать числу экзаменов. Оценка является целым числом в интервале от 2 до 5.

Программа в соответствии с пожеланиями пользователя формирует один их трех списков:

- Список студентов, сдавших экзамены (то есть не имеющих неудовлетворительных оценок), упорядоченный по убыванию среднего балла (средний балл печатается с точностью два знака после десятичной точки);
- Список студентов, сдавших экзамены на «хорошо» и «отлично» (в произвольном порядке);
- Список студентов, не сдавших экзамены (то есть имеющих хотя бы одну «двойку»).

Требования к интерфейсу программы. Программа выводит перенумерованный перечень возможных операций. Пользователь вводит номер операции. Программа выполняет соответствующую обработку и выводит требуемый список в указанный пользователем файл.

Пример входного файла “students.txt”:

```
Иванов 3 4 4 2
Петров 5 5 5 3 4 5
Александров 3 4 4 4
Сидоров 4 2 5 5 5
Кузнецов 3 5 5 4
```

Примерный сценарий взаимодействия с программой выглядит следующим образом:

```
Список операций:

1. Вывод списка успевающих студентов
2. Вывод списка студентов, имеющих только хорошие и отличные оценки
3. Вывод списка неуспевающих студентов

Введите номер операции: 1

Введите имя файла: output.txt
```

Содержимое файла “output.txt” после обработки:

```
Кузнецов 4.66
Петров 4.40
Александров 4.00
```

Зав. кафедрой _____