Владелец вашего любимого кафе обратился к вам с просьбой разработать программу для учета заказов. У него есть подозрения, что некоторые из нечистых на руку официантов часть заказов проводят мимо кассы.

В конце каждого дня он готов присылать вам следующую информацию в виде отдельных текстовых файлов:

1. в файле с кассы учитываются заказы с указанием блюд и количества заказанных порций каждым клиентом за день

Пример orders-20160201.txt:

Заказ №1

шарлотка = 1

кофе американо = 2

итого: 310,00

================

Заказ №2

мороженое = 3

лимон = 1

кофе латте = 2

итого: 460,00

================

1. В файле от доверенного менеджера с кухни указываются количество блюд, приготовленных за сутки, в том порядке, в котором передавали заказы официанты.

Пример kitchen-20160201.txt:

шарлотка = 1

кофе американо = 3

вайли с джемом = 2

мороженое = 5

кофе латте = 1

лимон = 1

кофе латте = 1

1. В отдельном файле задано меню кафе с указанием цен за одну порцию.

Пример menu-20160201.txt:

шарлотка = 150,00

вафли с джемом = 120,00

кофе американо = 80,00

кофе латте = 140,00

мороженое = 50,00

лимон = 30,00

Владелец кафе просит, чтобы ваша программа выдавала ему отчет о кражах за сутки со следующей информацией:

* перечень блюд и количество порций, которые были проведены мимо кассы;
* итоговая стоимость украденных блюд.

**Задания:**

**Этап I. Реализация на структурах и функциях**

1. Определить структуры данных.
2. Разделить реализацию на ввод/вывод и логику (алгоритмы обработки). Согласовать интерфейсы функций (названия, входные/выходные параметры) и их взаимодействие.
3. Каждый из членов бригады (из 2-х человек) реализует свою часть функций.
4. Интегрировать решения и протестировать на различных входных данных.

Необходимо:

* Установить и соблюдать соглашения по именованию и форматированию кода: корректное именование функций, переменных, структур данных.
* Грамотно декомпозировать алгоритм на функции: одна функция решает только одну задачу.
* Грамотная работа с динамической памятью new/delete (выделение/освобождение, отсутствие утечек)
* Обрабатывать некорректные ситуации при работе программы (не найден файл, некорректное содержание входных файлов - ошибки форматирования (пропущено количество или название, символ = и т.д.), ошибки в данных (опечатки в названиях блюд), отсутствие краж и т.д.)

Затем, разбить на модули: один или несколько заголовочных файлов, плюс реализация в одном или нескольких cpp-файлах.

**На защите:**

1. 60 баллов - сдача выполненного основного задания и ответы на вопросы
2. 40 баллов - выполнение дополнительного задания на защите

**Этап II. Реализация на классах. Инкапсуляция**

Расширение функциональности приложения:

* Блюда в меню разбиваются по категориям. Категории могут быть вложенными (подкатегории). Вложенность обозначается отступами (пробелами) перед названием категории в файле меню  
  Пример файла:  
   **Выпечка**

шарлотка = 150,00

вафли с джемом = 120,00

**Напитки**

**Кофе**

кофе американо = 80,00

кофе латте = 140,00

**Чай**

ассам (400 мл) = 100,00

эрл грей (400 мл) = 100,00

**Топинги**

мороженое = 50,00

лимон = 30,00

* Файлы заказчика с входными данными за несколько дней размещены в одном каталоге. Каждый файл содержит информацию за определенный день работы кафе (дата указывается в названии файла).
* Заказчик просит предоставить следующие виды отчетов и размещать их в каталоге reports:

1. Отчет об украденных блюдах (содержание отчета аналогично 1-му этапу) *за период по выбору пользователя:*
   1. За последний день (текущая дата)
   2. За определенные сутки (дата задается пользователем)
   3. За последнюю неделю (текущая дата и 6 предшествующих дней) с детализацией по дням
   4. За заданный период (задаются даты начала и окончания) с детализацией по дням
2. Отчет по статистике заказов, который содержит следующую информацию: о сумме выручки, о количестве заказов, о величине среднего чека (средняя сумма заказа за указанный период), о количестве заказанных блюд - по категориям и подкатегориям (с учетом вложенности) с детализацией до каждого наименования блюда  
   *за период по выбору пользователя:*
   1. За последний день (текущая дата)
   2. За определенные сутки (дата задается пользователем)
   3. За последнюю неделю (текущая дата и 6 предшествующих дней) с детализацией по дням
   4. За заданный период (задаются даты начала и окончания) с детализацией по дням
3. Отчет по динамике цен в меню - изменение цены в абсолютной величине и в процентах по каждому блюду в меню в сравнении *между двумя датами (заданными пользователем)* (указать в конце отчета, если появилось новое блюдо или некоторое блюдо изъято из меню)

Необходимо реализовать удобный текстовый (=консольный) интерфейс пользователя, позволяющий выбрать вид отчета из меню и задать параметры.

Входные данные в файлах, именование файлов в каталоге и входные параметры от пользователя в консоли необходимо продолжать проверять на корректность и сообщать пользователю об ошибках.

Нарисовать диаграмму классов для этапа, указав наиболее важные поля и методы, их видимость и структурные отношения между классами (в бумажном или электронном виде).

***Требования к этапу:***

1. Грамотная объектная декомпозиция и абстрагирование - разделение на классы, выделение полей и методов классов.
2. Отсутствие глобальных функций в коде (кроме функции main).
3. Использование конструкторов/деструкторов для создания/уничтожения объектов и грамотная работа с динамической памятью (отсутствие утечек).
4. Реализация требований инкапсуляции.
5. Реализация агрегации/композиции в классах.
6. Соблюдение соглашений по именованию и форматированию кода: корректное именование модулей с кодом и заголовочных файлов, классов, методов и полей классов, переменных.

**Сроки выполнения этапа:**

**с 14 по 25 марта:** установочные занятия, ответы на вопросы (Кирюшкин М.А.)

**с 4 апреля по 8 апреля: дедлайн** - защита этапа 2 (Ботов Д.С., Кондратенко И.Р.)

**Этап III. Наследование. Полиморфизм**

Расширение функциональности приложения:

Директор кафе вводит гибкую систему скидок

Скидки могут быть следующих видов:

* **Блюдо дня** - (процент скидки для каждого из блюд дня указывается после исходной цены блюда в файле меню на соответствующий день)
* **Вечерняя скидка** 10% - скидка при заказе после 21 часа (после номера каждого заказа в файле заказов указывается время заказа)

Например:   
 Заказ №12 21:15

* **Скидка имениннику** - 20% на определенные категории блюд в меню - признак того, что заказа сделал именинник указывается в строке названия заказа

Например:   
 Заказ №12 21:15 именинник

В меню после названия определенных категорий блюд, на которые распространяется скидка для именинников, указывается тег #именинник

* **Скидка на крупный заказ** -   
   5% скидка при заказе на сумму более 1500 рублей  
   6% скидка при заказе на сумму более 2500 рублей  
   8% скидка при заказе на сумму более 5000 рублей

Существует важное правило при расчете стоимости заказа:   
**скидки разного типа не суммируются друг с другом**

*Пояснение:* если к заказу (или его отдельным позициям) могут быть применены несколько разных видов скидок - то выбирается тот вид, который наиболее выгоден клиенту - он делает сумму заказа минимальной

**Отчет по контролю за расчетом скидок**

Заказчик не уверен, что кассиры корректно научились рассчитывать скидки и просит вас сделать новый отчет с проверкой расчета итогов по каждому заказу из файлов checks.txt за соответствующий период.

В отчете указываются заказы и сумма по заказу, которая была рассчитана неверно кассиром (из соответствующего файла checks), а также правильная сумма и расчетный размер и тип скидки:

<Дата и время заказа> <номер заказа> <неверная сумма> <верная сумма> <размер и тип скидки>

22.01.2016 11:00 Заказ №15 4100,30 4050,10 5% крупный заказ

24.01.2016 21:00 Заказ №15 5105,35 4850,00 10% вечерняя скидка

*Отчет по контролю составляется за период по выбору пользователя:*

* 1. За последний день (текущая дата)
  2. За определенные сутки (дата задается пользователем)
  3. За последнюю неделю (текущая дата и 6 предшествующих дней)
  4. За заданный период (задаются даты начала и окончания)

**Интерфейс настройки системы скидок**

Заказчик планирует настраивать систему скидок - менять процент скидок, условия скидки (например, время для вечерней скидки, сумму для скидки за крупный заказ и т.д.). Он просит разработать **удобный текстовый интерфейс пользователя** для настройки параметров и размеров для различных типов скидок.

Если пользователь не меняет настройки параметров скидок, то считать значения параметров по умолчанию из условия выше (см. перечень скидок).

***Требования к этапу:***

1. Грамотное применение принципов наследования, агрегации/композиции.
2. Реализация принципа полиморфизма для возможности гибкого изменения поведения системы при расчете скидок.
3. Грамотная объектная декомпозиция и абстрагирование - разделение на классы, выделение полей и методов классов. Разделение ответственностей классов (логика и ввод/вывод)

**Сроки выполнения этапа:**

**с 18 по 29 апреля:** консультации, ответы на вопросы (Ботов Д.С., Кондратенко И.Р.)

**с 2 мая по 6 мая: дедлайн** - защита этапа 3 (Ботов Д.С., Кондратенко И.Р.)

**с 10 мая по 13 мая: дедлайн** - итоговая сдача всего задания Кафе (Ботов Д.С., Кондратенко И.Р.)

**Чеклист итоговой сдачи**

* Полностью реализована **вся функциональность** всех этапов задания, программная система работоспособна
* **Не происходит сбоев и падений** системы во время работы, при некорректных входных данных программная система информирует об этом пользователя
* Выполнена **объектная декомпозиция** - каждый класс решает свою задачу, **разделены ответственности**: хранение данных, логика, ввод/вывод (интерфейс пользователя), взаимодействие классов реализуется через передачу объектов в методы, методы классов компактны (пара экранов кода - максимум 20-30 строк)
* Выполнена **модульная декомпозиция** - решение разделено на модули и заголовочные файлы, сильно взаимодействующие классы размещены в одном модуле
* Спроектирована **UML-диаграмма классов**, отражающая ключевые классы решения, их основные поля и методы, взаимодействие между классами
* Соблюдение **соглашений по именованию и форматированию** кода: именование модулей, классов, полей, методов, параметров и локальных переменных - названия четко и однозначно отражают суть элемента
* Отсутствуют **элементы процедурного программирования** - глобальные функции и переменные, злоупотребление статическими полями и методами классов, большие и сложные условные ветвления в методах
* Работа с **динамической памятью** без утечек, грамотное использование **конструкторов и деструкторов** для создания и уничтожения объектов
* Реализована **инкапсуляция** - внутренние особенности реализации классов скрыты, обозначен понятный интерфейс для взаимодействия с классом
* Отсутствует **дублирование кода методов**, структуры полей класса за счет применения **принципов агрегации/композиции и наследования**
* За счет применения **принципа полиморфизма** обеспечивается **гибкость системы**, которая выражается в расширении путем добавления новых классов и **минимальном изменении** существующего кода.   
  Моделируются следующие ситуации:
  + Добавление нового вида скидки
  + Добавление нового вида отчета
  + Добавление нового типа периода в параметры существующих отчетов