# Вариант 10 Спортивная команда

Разработать приложение, позволяющее организовать сбор и обработку информации о результатах спортивного турнира и игроках спортивной команды. В составе спортивной команды игроки разных амплуа (вратарь, защитник, нападающий). Информация об игроке соответствующего амплуа хранится в специальном описателе.

Описатель *вратаря* содержит следующую информацию: фамилия и инициалы игрока, “время в игре”; “штрафное время” игрока за время игры; количество бросков по воротам, отражённых и нет.

Описатель *защитника* содержит следующую информацию: фамилия и инициалы игрока, “время в игре”; “штрафное время” игрока за время игры.

Описатель *нападающего* содержит следующую информацию: фамилия и инициалы игрока, “время в игре”; “штрафное время” игрока за время игры; количество бросков по воротам, результативных и нет.

Спортивная команда проводит турнирные игры, для каждой из которой собирается информация о составе игроков и их показателях в конкретной игре. Информация обо всех играх сведена в просматриваемую таблицу, каждый элемент которой содержит “дату игры”, название команды соперника, количество игроков, принимавших участие в игре и указатель на динамический массив описателей игроков.

Полная информация о команде собрана в описателе команды, который содержит: название команды, фамилию и инициалы тренера, таблицу игр.

Обеспечить выполнение следующих операций:

* Для описателя команды:
  + получить название команды; получить фамилию и инициалы тренера;
  + включить новый элемент – информацию об игре;
  + найти элемент по заданной “дате игры”;
  + удалить элемент, заданный “датой игры”;
  + показать содержимое таблицы.
* Для турнирной игры:
  + вывести информацию о дате игры, сопернике и составе команды;
  + модифицировать состав команды.
* Для любого игрока команды:
  + вывести информацию об игроке;
  + получить (вернуть в качестве результата) амплуа игрока;
  + получить время в игре; получить штрафное время.
* Для вратаря команды:
  + получить количество отражённых или пропущенных бросков по воротам.
* Для нападающего команды:
  + получить количество результативных или нерезультативных бросков по воротам.
* Для приложения:
  + редактирование описания игры - внесение, исключение и модификация описания игры;
  + вычислить общее “штрафное время” всей команды в конкретной игре
  + вывести краткую информацию об игроках команды с указанием их амплуа;
  + (\*) получить отчёт о турнирном положении команды, используя класс-итератор.

# Порядок выполнения работы

1. На основе описания задачи определить состав классов.
2. Разработать иерархию классов и схему их взаимодействия.
3. Для каждого класса определить его состояние и необходимые методы.
4. Разработать и отладить все классы. Приложение реализовать в виде простой диалоговой программы.
5. (\*) Разработать и отладить класс, реализующий работу приложения. Предусмотреть в классе возможность сохранения таблицы в файле и ее восстановления.
6. (\*) Разработать и отладить класс-итератор, с помощью которого реализовать дополнительные операции.
7. (\*) Повторить разработку контейнерного класса (вместе с необходимыми итераторами), используя стандартную библиотеку STL. Выбор шаблона классов согласовать с преподавателем.
8. (\*) Повторить разработку контейнерного класса (вместе с необходимыми итераторами), используя собственные шаблоны классов.
9. (\*) Реализовать приложение с использование средств оконного интерфейса.

Примечание: пункты задания, отмеченные (\*), являются дополнительными.