Содержание

Примеры LINQ-запросов

Сумма по справочникам выбранных объектов

```
var deals = dataContext.Kontragent_9e5de8_List
   .Where(z => z.Name.Contains("Фрагмент"))
   .Select(z => new {z.Name, Trudochasy =
z.GetChildrenHierarchy<Uchet_vremeni_3156dc>().Sum(r => r.Chasi), Dengi =
z.GetChildrenHierarchy<Oplata_45b34c>().Sum(r => r.Summa)})
   .OrderBy(z=> z.Name);

return deals;
```

Просроченные задачи пользователей

```
var dToday = DateTime.Today;

var tasks = dataContext.Zadacha_c88ec2_List
    .Where(z => z.SystemEndDate < dToday && (z.Status ==
ProjectStatus.InWork || z.Status == ProjectStatus.NotStarted) )
    .OrderBy(z => z.Responsible.FirstName).ThenBy(z =>
z.Responsible.LastName).ThenBy(z => z.Name)
    .Select(z => new { z.Name,
                       user_id = z.Responsible.Id,
                       Ispolnitel = String.Concat(z.Responsible.FirstName, "
", z.Responsible.LastName),
                       z.Responsible.Email,
                       z.SystemStartDate, z.SystemEndDate
                     }
           );

return tasks;
```

Задачи за 3 дня до начала (Задать дату начала)

```
var dStart = new DateTime(2019, 5, 1);
var dEnd = dStart.AddDays(4);

var deals = dataContext.SMART__zadacha_45a3a0_List
    .Where(z => z.PlannedStartDate >= dStart && z.PlannedStartDate < dEnd )
    .OrderBy(z => z.Name)
    .Select(z => new { z.Name,
                       boss = String.Concat(z.Owner.LastName, " ",
z.Owner.FirstName),
```

```
                        responsible = String.Concat(z.Responsible.LastName, "
", z.Responsible.FirstName),
                        Creator = String.Concat(z.Creator.LastName, " ",
z.Creator.FirstName),
                        Status = (z.Status == ProjectStatus.InWork ? "В
работе" :
                                (z.Status == ProjectStatus.Complete ?
"Выполнен" :
                                  (z.Status == ProjectStatus.Cancelled ?
"Отменен" :
                                     (z.Status == ProjectStatus.Freeze ?
"Отложен" :
                                        (z.Status == ProjectStatus.NotStarted
? "Не начат" :
                                           "Готов к проверке"
                                        )
                                     )
                                  )
                                )
                        ),
                        z.PlannedStartDate, z.SystemStartDate,
z.ActualEndDate, dStart, dEnd});
return deals;
```

Задача с датой окончания и статус отчёта по задаче

```
var tasks = dataContext.Zadacha_c88ec2_List
    .Where(z =>
z.GetChildrenHierarchy<Otchet_o_statuse_zadachi_3c7214>().Any())
    .OrderByDescending(z => z.SystemEndDate)
    .Select(z => new
    {
      z.Name,
      z.PlannedEndDate, //Плановая ДО (ограничение)
      z.SystemEndDate,  //Расчетная ДО (из планировщика)
      Otchet_o_statuse =
z.GetChildrenHierarchy<Otchet_o_statuse_zadachi_3c7214>()
                          .OrderByDescending(r => r.Date)
                          .FirstOrDefault()
                          .Status_056648.Name
    });

return tasks;
```

Контрагенты, у которых есть не закончившиеся

инсталляции (через связи)

```
var dStart = DateTime.Now;

var Kontragents = dataContext.Kontragent_9e5de8_List
    .Where(z => z.Installyaciya_180ce0.Where(p => p.Data_okonchaniya_licenzii
>= dStart).Any() )
   .OrderBy(z => z.Name)
   .Select(z => new {
     z.Name,
     z.Id,
     RespobsibleId = z.Responsible.Id,
     z.Installyaciya_180ce0.FirstOrDefault().Data_okonchaniya_licenzii,
     Nazvanie_inst = z.Installyaciya_180ce0.FirstOrDefault().Name,
     Chislo = z.Installyaciya_180ce0.Count()
   });

return Kontragents;
```

Прошлогодние задачи конкретного пользователя

```
var personId = parameters.GetValueOrDefault<Guid>("PersonId", new
Guid("50330e78-01c7-4280-9c74-2be072991628"));
var previousYear = DateTime.Now.AddYears(-1).Year;

var tasks = dataContext.Zadacha_razrabotki_6ad838_List
    .Where(z => z.Celj_Opisanie != null &&
           z.PlannedStartDate.Value.Year == previousYear &&
           z.Responsible.Id == personId)
    .OrderByDescending(z => z.PlannedStartDate)
    .Select(z => new { z.Name, z.SystemStartDate, z.SystemEndDate,
z.Celj_Opisanie, z.Trudoyomkostj });

return tasks;
```

Сделки созданные за апрель 2019

```
var deals = dataContext.Sdelka_3f169d_List
    .Where(z => z.CreationDate >= new DateTime(2019, 4, 1) && z.CreationDate
< new DateTime(2019, 5, 1) )
    .OrderBy(z => z.Name)
    .Select(z => new { Sdelka = z.Name,
                       KontragentName =
z.GetParentHierarchy<Kontragent_9e5de8>().FirstOrDefault().Name,
                       boss = String.Concat(z.Owner.LastName, " ",
z.Owner.FirstName),
```

```
                        responsible = String.Concat(z.Responsible.LastName, "
", z.Responsible.FirstName),
                        Creator = String.Concat(z.Creator.LastName, " ",
z.Creator.FirstName),
                        Status = (z.Status == ProjectStatus.InWork ? "В
работе" :
                                (z.Status == ProjectStatus.Complete ?
"Выполнен" :
                                  (z.Status == ProjectStatus.Cancelled ?
"Отменен" :
                                     (z.Status == ProjectStatus.Freeze ?
"Отложен" :
                                        (z.Status == ProjectStatus.NotStarted
? "Не начат" :
                                           "Готов к проверке"
                                        )
                                     )
                                  )
                               )
                            ),
                        Prichina_otkaza = z.Prichini_otkaza_5ad9a4.Name,
                        z.SystemEndDate, z.SystemStartDate, z.ActualEndDate,
                        delegirovano_menedjeru = ((DateTime?)
z.Data_delegirovaniya_sotrudniku).HasValue ? 1 : 0 ,
                        Sdelka_length =
DbFunctions.DiffDays(z.SystemStartDate, z.SystemEndDate) });
return deals;
```

Получение ID родителя

у объекта есть метод `public TCustomType GetParent<TCustomType>()` - получение родителя, если родитель не TCustomType, то вернется `null` - возвращает прямого (ровно на 1 уровень вверх) родителя

если надо выше, то - `public ICollection<TCustomType> GetParentHierarchy<TCustomType>(bool includeRoot)` получение родителей с учетом иерархии

Поиск родителя

```
var nodeGUID =
parameters.GetValueOrDefault<Guid?>(QueryParameters.SpreadsheetReport.Projec
tId, new Guid("3e637a10-51b8-4a02-89fe-35b645969ca5"));

var projects = dataContext.Projects
  .Where(p => p.GetParentHierarchy<Proekt_vnedreniya_15f54a>(true).Any(i =>
i.Id == nodeGUID))
  .OrderBy(p => p.Parent.Id)
```

```
  .OrderBy(p => p.CreationDate)
  .Select(p => new{
    p.Name,
    object_id = p.Id,
    ParentId = p.Parent.Id
      });
return projects;
```

Запрос по проекту (с выводом количества разного рода объектов) для анализа состояния проекта

Запрос по проекту, позволяющий посчитать количество объектов для анализа состояния проекта:

1. Все объекты.
2. Все конечные объекты.
3. Количество работ без зависимостей.
4. Количество работ с фиксированными сроками.
5. Всего объектов просрочено.
6. Количество конечных работ с несвоевременным началом.
7. Количество приоритетных задач.
8. Конечных работе без отчета о ходе работ и т.п.

```
var id =
parameters.GetValueOrDefault<Guid?>(QueryParameters.SpreadsheetReport.Projec
tId,  null /* new Guid("273bf10a-f403-4e62-a95d-464ad94616af")*/ );
var idProject = id;
var сегодня = DateTime.Now;
var сегодня_7дн = сегодня.AddDays (-7);
var сегодня_14дн = сегодня.AddDays (-14);
var сегодня_плюс_12 = сегодня.AddMonths(12);
var сегодня_плюс_3 = сегодня.AddMonths(3);
var сегодня_мин_3_мес = сегодня.AddMonths(-3);
// var currentYear = new DateTime(сегодня.Year.Month,1);
var defaultCalendar = workCalendars.GetDefaultCalendar();

var dToday = DateTime.Now;
var child = dataContext.Projects
    .Where(p => p.GetParentHierarchy<Project>(false).Any(a => a.Id ==
idProject))
    .Where(a => a.GetChildren<Project>().Any() == false )
    .Where(a => a.BaselinePlanEndDate != null && (a.Status ==
ProjectStatus.NotStarted  || a.Status == ProjectStatus.InWork))
    .Select(p => new {
        p.Id,
        ParentId = p.Parent.Id,
        Calendar = p.CalendarId,
        Start = p.SystemStartDate,
        End = p.SystemEndDate,
```

```
        UDN = p.BaselinePlanStartDate,
        UDO = p.BaselinePlanEndDate,
        Status1 = p.Status,
//        HasChild = p.GetChildren<Project>().Any() == false
})
  .ToList()
  .Where(a =>
(Math.Round(workCalendars.GetWorkCalendar(a.Calendar).GetWorktimeDays(a.Star
t.Value.Date, a.End.Value.Date)) <  (a.UDO != null ?
Math.Round(workCalendars.GetWorkCalendar(a.Calendar).GetWorktimeDays(a.UDN.V
alue.Date, a.UDO.Value.Date)) : 0)))
  .Select(p => new
    {   p.Id,
        p.ParentId,
        p.Calendar,
        p.Start,
        p.End,
        p.UDN,
        p.UDO,
        p.Status1,
    План_длительность =
Math.Round(workCalendars.GetWorkCalendar(p.Calendar).GetWorktimeDays(p.Start
.Value.Date, p.End.Value.Date)),
    Утв_длительность = p.UDO != null ?
Math.Round(workCalendars.GetWorkCalendar(p.Calendar).GetWorktimeDays(p.UDN.V
alue.Date, p.UDO.Value.Date)) : 0,
 });

var projects = dataContext.Projects
      .Where(p => p.GetParentHierarchy<Project>(true).Any(r => r.Id ==
idProject))
      .Where(p => p is Proekt_tipovoj_b9f9ae || p is Proekt_tipovoj_2_82b061
|| p is Proekt_Lean_a65539)
      .Where(p => p.Name != null)

  .Select(p => new {
    ID_проекта = p.Id,
  Название_проекта = p.Name,
  Руководитель = String.Concat(p.Owner.LastName, " ", p.Owner.FirstName),
  План_завершения = DbFunctions.TruncateTime(p.SystemEndDate),
    SystemStartDate = DbFunctions.TruncateTime(p.SystemStartDate),
    SystemEndDate = DbFunctions.TruncateTime(p.SystemEndDate),
  Утвержд_завершение = DbFunctions.TruncateTime(p.BaselinePlanEndDate),
  Утвержд_начало = DbFunctions.TruncateTime(p.BaselinePlanStartDate),
  Процент_завершения = p.PercentComplete,
  Всего_объектов = p.GetChildrenHierarchy<Project>(false).Select(a => a.Name
!= null).Count(),
  Всего_конечных_объектов = p.GetChildrenHierarchy<Project>(false).Where(a =>
a.GetChildren<Project>().Any() == false).Count(),
  Конечных_год_вперед_1 = p.GetChildrenHierarchy<Project>(false).Where(a =>
a.GetChildren<Project>().Any() == false && a.SystemEndDate >= сегодня &&
```

```
a.SystemEndDate <= сегодня_плюс_12).Count(),
  Конечных_более1мес_1 = p.GetChildrenHierarchy<Project>(false).Where(a =>
a.GetChildren<Project>().Any() == false && a.SystemEndDate >= сегодня &&
a.SystemEndDate <= сегодня_плюс_12 &&
DbFunctions.DiffDays(a.SystemStartDate,a.SystemEndDate)> 31).Count(),
  Колво_работ_без_связей_2 = p.GetChildrenHierarchy<Project>(false).Where(a =>
a.GetChildren<Project>().Any() == false && a.Dependencies.In.Count ==
0).Count(),
  Колво_работ_с_фиксом_3 = p.GetChildrenHierarchy<Project>(false).Where(a =>
a.GetChildren<Project>().Any() == false && a.PlannedStartDate.HasValue ||
a.PlannedEndDate.HasValue).Count(),
  Всего_объектов_просрочено_5 = p.GetChildrenHierarchy<Project>(false).Where(a
=> a.Status == ProjectStatus.NotStarted  || a.Status == ProjectStatus.InWork
|| a.Status == ProjectStatus.Ready).Where(s =>
DbFunctions.TruncateTime(s.SystemEndDate) <
DbFunctions.TruncateTime(сегодня)).Count(),
  Всего_объектов_критич_6 = p.GetChildrenHierarchy<Project>(false).Where(a =>
a.Priority == 1).Count(),

    /*В ПОСТОБРАБОТКЕ СЧИТАЕТ РАБ.ДНИ*/
  Конечных_длит_7 = p.GetChildrenHierarchy<Project>(false).Where(a =>
a.GetChildren<Project>().Any() == false && a.BaselinePlanEndDate != null &&
(a.Status == ProjectStatus.NotStarted  || a.Status == ProjectStatus.InWork)
&& DbFunctions.DiffDays(a.SystemStartDate,a.SystemEndDate) <
DbFunctions.DiffDays(a.BaselinePlanStartDate,a.BaselinePlanEndDate)).Count()
,

  Длительность_проекта = DbFunctions.DiffDays(p.SystemStartDate,
p.SystemEndDate),
  Длительность_проекта_9 = (100 -
((DbFunctions.DiffDays(p.BaselinePlanEndDate,
p.BaselinePlanStartDate))/(DbFunctions.DiffDays(p.SystemEndDate,
p.SystemStartDate))) * 100),
  Конечных_несвоевр_начало_12 = p.GetChildrenHierarchy<Project>(false).Where(a
=> a.GetChildren<Project>().Any() == false && a.Priority != 1).Where (s =>
s.ActualStartDate != null ?
DbFunctions.DiffDays(s.SystemStartDate,s.ActualStartDate) > 7  :
s.SystemStartDate < сегодня_7дн).Count(),
    /*ПРОВЕРИТЬ*/
  Конечных_несвоевр_начало_критич_12 =
p.GetChildrenHierarchy<Project>(false).Where(a =>
a.GetChildren<Project>().Any() == false && a.Priority == 1).Where (s =>
s.ActualStartDate != null ?
DbFunctions.DiffDays(s.SystemStartDate,s.ActualStartDate) > 7  :
s.SystemStartDate < сегодня_7дн).Count(),

  Конечных_без_отчета_13 = p.GetChildrenHierarchy<Project>(false).Where(a =>
a.GetChildren<Project>().Any() == false && a.Priority != 1).Where(a =>
(a.Status == ProjectStatus.InWork && a.SystemStartDate < сегодня_14дн) || (
a.Status == ProjectStatus.NotStarted &&
DbFunctions.TruncateTime(a.SystemStartDate) <
```

```
DbFunctions.TruncateTime(сегодня) )).Where(a =>
a.GetChildren<Otchet_o_hode_rabot_868b12>().OrderByDescending(r =>
r.Date).FirstOrDefault().Date < сегодня_14дн ||
a.GetChildren<Otchet_o_hode_rabot_868b12>().OrderByDescending(r =>
r.Date).FirstOrDefault().Date == null ).Count(),

  Конечных_крит_без_отчета_13 = p.GetChildrenHierarchy<Project>(false).Where(a
=> a.GetChildren<Project>().Any() == false &&  a.Priority == 1).Where(a =>
(a.Status == ProjectStatus.InWork && a.SystemStartDate < сегодня_14дн) || (
a.Status == ProjectStatus.NotStarted &&
DbFunctions.TruncateTime(a.SystemStartDate) <
DbFunctions.TruncateTime(сегодня) )).Where(a =>
a.GetChildren<Otchet_o_hode_rabot_868b12>().OrderByDescending(r =>
r.Date).FirstOrDefault().Date < сегодня_14дн ||
a.GetChildren<Otchet_o_hode_rabot_868b12>().OrderByDescending(r =>
r.Date).FirstOrDefault().Date == null ).Count(),

  Конечных_просроченных_без_отчета_14 =
p.GetChildrenHierarchy<Project>(false).Where(a =>
a.GetChildren<Project>().Any() == false && a.Priority != 1)
                                        .Where(a => a.Status ==
ProjectStatus.InWork || a.Status == ProjectStatus.NotStarted || a.Status ==
ProjectStatus.Freeze)
                                        .Where(s => s.BaselinePlanEndDate
!= null && DbFunctions.TruncateTime(сегодня) >
DbFunctions.TruncateTime(s.BaselinePlanEndDate)).Count(),
  Конечных_просроченных_14_со_статусом =
p.GetChildrenHierarchy<Project>(false).Where(a =>
a.GetChildren<Project>().Any() == false && a.Priority != 1)
                                        .Where(a => a.Status ==
ProjectStatus.InWork || a.Status == ProjectStatus.NotStarted || a.Status ==
ProjectStatus.Freeze)
                                        .Where(s => s.BaselinePlanEndDate
!= null && DbFunctions.TruncateTime(сегодня) >
DbFunctions.TruncateTime(s.BaselinePlanEndDate))
                                        .Where(h =>
h.GetChildren<Otchet_o_hode_rabot_868b12>().Where(r => r.Date >=
сегодня_мин_3_мес).Any(s => s.Indikator_problem_b7386a.Id ==
Classifier_Indikator_problem_6c3d8e.Krizis_96cbc3ec_Id ||
s.Indikator_problem_b7386a.Id ==
Classifier_Indikator_problem_6c3d8e.Problemi_47ab098b_Id)).Count(),

  Конечных_критич_без_отчета_14 =
p.GetChildrenHierarchy<Project>(false).Where(a =>
a.GetChildren<Project>().Any() == false && a.Priority == 1)
                                        .Where(a => a.Status ==
ProjectStatus.InWork || a.Status == ProjectStatus.NotStarted || a.Status ==
ProjectStatus.Freeze)
                                        .Where(s => s.BaselinePlanEndDate
!= null && DbFunctions.TruncateTime(сегодня) >
DbFunctions.TruncateTime(s.BaselinePlanEndDate)).Count(),
```

```
  Конечных_критич_14_со_статусом =
p.GetChildrenHierarchy<Project>(false).Where(a =>
a.GetChildren<Project>().Any() == false && a.Priority == 1)
                                          .Where(a => a.Status ==
ProjectStatus.InWork || a.Status == ProjectStatus.NotStarted || a.Status ==
ProjectStatus.Freeze)
                                          .Where(s => s.BaselinePlanEndDate
!= null && DbFunctions.TruncateTime(сегодня) >
DbFunctions.TruncateTime(s.BaselinePlanEndDate))
                                          .Where(h =>
h.GetChildren<Otchet_o_hode_rabot_868b12>().Where(r => r.Date >=
сегодня_мин_3_мес).Any(s => s.Indikator_problem_b7386a.Id ==
Classifier_Indikator_problem_6c3d8e.Krizis_96cbc3ec_Id ||
s.Indikator_problem_b7386a.Id ==
Classifier_Indikator_problem_6c3d8e.Problemi_47ab098b_Id)).Count(),

    new_колво_мес_КТ_4 = p.GetChildrenHierarchy<Project>(false).Where(k => k
is KT_1_f0f97e || k is KT_2_6393db || k is KT_3_f1cf19).Where(a =>
a.SystemEndDate >= сегодня && a.SystemEndDate <= сегодня_плюс_12).GroupBy(g =>
g.SystemEndDate.Value.Month).Count(),
    new_колво_мес_проект_4 = DbFunctions.DiffMonths(сегодня, p.SystemEndDate <=
сегодня_плюс_12 ? p.SystemEndDate : сегодня_плюс_12),
  Сегодня = DbFunctions.TruncateTime(сегодня),
  Плюс_12_мес_ = сегодня_плюс_12,
  Тек_мес_14дн = сегодня_14дн,
 })
 .ToList()

  .Select(p => new{
    p.ID_проекта,
    p.Название_проекта,
    p.Руководитель,
    p.SystemStartDate,
    p.SystemEndDate,
    p.Утвержд_завершение,
    p.Утвержд_начало,
    p.Процент_завершения,
    p.Всего_объектов,
    p.Всего_конечных_объектов,
    p.Конечных_год_вперед_1,
    p.Конечных_более1мес_1,
  Колво_работ_без_связей_2 = p.Колво_работ_без_связей_2 == 0 ? 0 :
p.Колво_работ_без_связей_2 - 1,
    p.Колво_работ_с_фиксом_3,
    p.Всего_объектов_просрочено_5,
    p.Всего_объектов_критич_6,
  Конечных_длит_7 = child.Count(),
    p.Длительность_проекта,
    p.Длительность_проекта_9,
    p.Конечных_несвоевр_начало_12,
    p.Конечных_несвоевр_начало_критич_12,
```

```
    p.Конечных_без_отчета_13,
    p.Конечных_крит_без_отчета_13,
  Конечных_просроченных_без_отчета_14 = p.Конечных_просроченных_без_отчета_14 -
p.Конечных_просроченных_14_со_статусом,
  Конечных_критич_без_отчета_14 = p.Конечных_критич_без_отчета_14 -
p.Конечных_критич_14_со_статусом,
    p.new_колво_мес_КТ_4,
    p.new_колво_мес_проект_4,
    p.Сегодня,

 })

  .ToList();
return projects;
```

Объединение плановых и фактических трудозатрат

```
var Portfel = parameters.GetValueOrDefault<string>("Portfel",
"14c8bd68-96c9-46e5-b870-5ef4e341e4f9");
var Текущие_проекты = new Guid("1d1ac40d-156b-4726-b2be-877aae71deaa");

var plan = dataContext.ProjectCosts
      .Where(p => p.Project.GetParentHierarchy<Katalog_743df4>(true).Any(r
=> r.Id == Текущие_проекты))
  .Select(c => new
          {
            ProjectUID =
(Guid?)c.Project.GetParentHierarchy<Proekt_a3e2fb>(true).FirstOrDefault().Id
,
            TaskUID = c.ProjectId,
         //   ProjectName = c.Project.Name,
         //   ResourceUID_plan = c.PersonId,
         //   ResourseUID_fakt = (String?) "факт",
            Date = (DateTime?)c.Date,
            Plan = (double?)c.Value,
            Fact = (double?)null
          })
  .ToList();

var fact = dataContext.Fakticheskie_trudozatrati_977ff0_List
      .Where(p => p.Project.GetParentHierarchy<Katalog_743df4>(true).Any(r
=> r.Id == Текущие_проекты))
  .Select(f => new
          {
            ProjectUID =
(Guid?)f.Project.GetParentHierarchy<Proekt_a3e2fb>(true).FirstOrDefault().Id
,
            TaskUID = f.Project.Id,
       //     ProjectName = f.Project.Name,
```

```
        //     ResourceUID_plan = (String?) "план",
        //     ResourseUID_fakt = f.Resurs_caaaaa,
              f.Date,
              Plan = (double?)null,
              Fact = (double?)f.Summa_06ff93,
            })
  .ToList();

var costs = plan.Union(fact)
  .GroupBy(c => (c.Date, c.TaskUID),
          (keys, values) => new
           {
             values.FirstOrDefault().ProjectUID,
             keys.TaskUID,
         //     values.FirstOrDefault().ProjectName,
             keys.Date,
             Plan = values.Sum(s => s.Plan),
             Fact = values.Sum(s => s.Fact)
           })
  .ToList();

return costs;
```

Запрос с параметром, для использования предфильтра в excel-отчете

Сам параметр отдельно:

```
var classifier1 =
parameters.GetValueOrDefault<ClassifierItem>("Proizvodstvo___peredel", new
ClassifierItem{}).Id;
```

Фильтр по параметру в основном запросе:

```
.Where(ps => classifier1 != Guid.Empty ?
(ps.Proizvodstvo___peredel_93e62e.Id == classifier1) : true)
```

В самом предфильтре значение параметра:

```
Proizvodstvo___peredel
```

```
var id =
parameters.GetValueOrDefault<Guid?>(QueryParameters.SpreadsheetReport.Projec
tId, null);
var startDatePTR =
parameters.GetValueOrDefault<DateTime?>(QueryParameters.SpreadsheetReport.St
artDate, DateTime.Now);
var idPTR = id; //
```

```
parameters.GetValueOrDefault<Guid?>(QueryParameters.SpreadsheetReport.ProjectId, id);
var endDatePTR = parameters.GetValueOrDefault<DateTime?>(QueryParameters.SpreadsheetReport.EndDate, DateTime.Now);
var endDate = endDatePTR.Value.AddDays (1);
var currentYear = new DateTime(endDatePTR.Value.Year, 1, 1);
var currentYearend = currentYear.AddMonths(12);
var k1 = currentYear.AddMonths(3);
var k2 = k1.AddMonths(3);
var k3 = k2.AddMonths(3);
var k4 = k3.AddMonths(3);
var y2 = currentYear.AddYears(1);
var y3 = currentYear.AddYears(2);
var y4 = currentYear.AddYears(3);
//*************************************************************************
*
//ФИЛЬТР_1
var classifier1 = parameters.GetValueOrDefault<ClassifierItem>("Proizvodstvo___peredel", new ClassifierItem{}).Id;
//*************************************************************************
*
var kt_3 = dataContext.KT_3_f1cf19_List
  .Where(kt3 => kt3.GetParentHierarchy<_PTR_predpriyatiya_Holdinga_41f8ea>(true).Any(r => r.Id == idPTR))
  .Where(kt3 => kt3.GetParentHierarchy<_Meropriyatie_POF_fada13>(true).Any())
  .Where(kt3 => kt3.Status != ProjectStatus.Cancelled )
  .OrderBy(kt3 => kt3.CreationDate)
  .Select(kt3 => new {
    kt3.Name,
    MeropriyatieId = kt3.GetParentHierarchy<_Meropriyatie_POF_fada13>(true).FirstOrDefault().Id,
    EtapPtrId = (Guid?)kt3.GetParentHierarchy<Etap_PTR_0eed50>(true).FirstOrDefault().Id,
    PsId = (Guid?)kt3.GetParentHierarchy<Poziciya_specifikacii_437efa>(true).FirstOrDefault().Id,

  Служебный_тип_для_оборудования = (Guid?)kt3.Sluzhebnij_Tip_KT_dlya_oborudovaniya_SMR_ab7e86.Id,
    kt3.BaselinePlanEndDate,
    kt3.ActualEndDate,
  })
  .ToList();

var report22 = dataContext.Poziciya_specifikacii_437efa_List
  .Where(ps => ps.Otnoshenie_k_chasti_PTR_c2c87f.Id == Classifier_Otnoshenie_k_chasti_PTR_0d5548.Osnovnaya_chastj_890632c4_Id ||
```

```
ps.Otnoshenie_k_chasti_PTR_c2c87f.Id == null)
  .Where(ps => ps.Data_vklyucheniya_v_PTR_73d436 >= startDatePTR.Value)
  .Where(ps =>
ps.GetParentHierarchy<_Meropriyatie_POF_fada13>(true).FirstOrDefault().Basel
inePlanStartDate < currentYearend)
  .Where(ps =>
ps.GetParentHierarchy<_PTR_predpriyatiya_Holdinga_41f8ea>(true).Any(r =>
r.Id == idPTR))
  .Where(ps => ps.Status == ProjectStatus.InWork ||
               ps.Status == ProjectStatus.NotStarted ||
               ps.Status == ProjectStatus.Ready ||
               ps.Status == ProjectStatus.Complete ||
               ps.Status == ProjectStatus.Freeze
          )
  .Where(ps => ps.Gruppa_oborudovaniya_fdcaf9.Id ==
Classifier_Gruppa_oborudovaniya_94eb92.Osnovnoe_0dfc21cd_Id ||
               ps.Gruppa_oborudovaniya_fdcaf9.Id ==
Classifier_Gruppa_oborudovaniya_94eb92.Komplektuyuschee_7517702d_Id)
  .Where(ps => ps.Kategoriya_rashodov_d87c7c.Id ==
Classifier_Kategoriya_rashodov_df2928.Kapitaliziruemie_bcc3e43c_Id)
  .Where(ps => ps.Razdel_PTR_730d6c.Id ==
Classifier_Razdel_PTR_27f85e._22_6200148b_Id)
  .Where(ps => classifier1 != Guid.Empty ?
(ps.Proizvodstvo___peredel_93e62e.Id == classifier1) : true)
  .OrderBy(ps => ps.CreationDate)
  .Select(ps => new {
    ID_проекта = ps.Id,
  Название_проекта = ps.Name,
  Производство_передел = ps.Proizvodstvo___peredel_93e62e.Name,
  Группа_оборудования = ps.Gruppa_oborudovaniya_fdcaf9.Name,
  Отношение_к_части_ПТР = ps.Otnoshenie_k_chasti_PTR_c2c87f.Name,
  Код_МПОФ =
ps.GetParentHierarchy<_Meropriyatie_POF_fada13>(true).FirstOrDefault().Kod_m
eropriyatiya_POF_b040c3,
    ID_МПОФ =
ps.GetParentHierarchy<_Meropriyatie_POF_fada13>(true).FirstOrDefault().Id,
  Марка_оборудования =
ps.GetChildren<Objekti_OF_5a0b23>().FirstOrDefault().Marka_oborudovaniya_2d9
569,
  Фирма_страна_производитель =
ps.GetChildren<Objekti_OF_5a0b23>().FirstOrDefault().Firma___Strana_proizvod
itelj_e5f645,
  Фирма_страна_поставщик =
ps.GetChildren<Objekti_OF_5a0b23>().FirstOrDefault().Firma___Strana_postavsc
hik_b87136,
    ID_узкого_места =
(Guid?)ps.Svyazan_s__uzkim_mestom_459c92_08ee48.FirstOrDefault().Id,
  Код_узкого_места =
ps.Svyazan_s__uzkim_mestom_459c92_08ee48.FirstOrDefault().Kod_uzkogo_mesta_6
f8714,
  Кол_во = ps.Kol__vo_6a6a43,
```

```
  Бюджет_ПОФ = ps.GetChildren<Byudzhet_POF_a981b9>().Sum(s =>
s.Byudzhet_POF_rub_07eb8e),
  Дата_включения_в_ПТР = ps.Data_vklyucheniya_v_PTR_73d436,
  ФАКТ_платежи = ps.GetChildren<FAKT_Platezhi_8c97d9>().Sum(s =>
s.Summa_platezha_1627e7),
  ФАКТ_платежи_с_предыдущих_лет =
ps.GetChildren<FAKT_Platezhi_8c97d9>().Where(f => f.Date <
currentYear).Sum(s => s.Summa_platezha_1627e7),
  Причины_отклонений =
ps.GetChildren<Otchet_o_hode_rabot_868b12>().OrderByDescending(r =>
r.Date).FirstOrDefault().Prichini_otklonenij_737f87,
  Предприятия_холдинга =
ps.GetParentHierarchy<_PTR_predpriyatiya_Holdinga_41f8ea>(true).FirstOrDefau
lt().Predpriyatie_Holdinga_d4b276.Name,
  Кол_во_Факт = ps.Kol__vo_FAKT_747cbf,

  План_сумма_1_кв =
ps.GetChildren<Byudzhet_POF_Grafik_platezhej_947f82>().Where(p => p.Date >=
currentYear && p.Date < k1).Sum(s => s.Byudzhet_POF_Platezhi_rub_411552),
  План_сумма_2_кв =
ps.GetChildren<Byudzhet_POF_Grafik_platezhej_947f82>().Where(p => p.Date >=
k1 && p.Date < k2).Sum(s => s.Byudzhet_POF_Platezhi_rub_411552),
  План_сумма_3_кв =
ps.GetChildren<Byudzhet_POF_Grafik_platezhej_947f82>().Where(p => p.Date >=
k2 && p.Date < k3).Sum(s => s.Byudzhet_POF_Platezhi_rub_411552),
  План_сумма_4_кв =
ps.GetChildren<Byudzhet_POF_Grafik_platezhej_947f82>().Where(p => p.Date >=
k3 && p.Date < k4).Sum(s => s.Byudzhet_POF_Platezhi_rub_411552),

  Факт_сумма_1_кв = ps.GetChildren<FAKT_Platezhi_8c97d9>().Where(p => p.Date
>= currentYear && p.Date < k1  && p.Date <= endDatePTR).Sum(s =>
s.Summa_platezha_1627e7),
  Факт_сумма_2_кв = ps.GetChildren<FAKT_Platezhi_8c97d9>().Where(p => p.Date
>= k1 && p.Date < k2  && p.Date <= endDatePTR).Sum(s =>
s.Summa_platezha_1627e7),
  Факт_сумма_3_кв = ps.GetChildren<FAKT_Platezhi_8c97d9>().Where(p => p.Date
>= k2 && p.Date < k3  && p.Date <= endDatePTR).Sum(s =>
s.Summa_platezha_1627e7),
  Факт_сумма_4_кв = ps.GetChildren<FAKT_Platezhi_8c97d9>().Where(p => p.Date
>= k3 && p.Date < k4  && p.Date <= endDatePTR).Sum(s =>
s.Summa_platezha_1627e7),

  Номер_договора =
ps.GetParentHierarchy<Dogovor_so_specifikaciej_a302b7>(true).FirstOrDefault(
).Nomer_dogovora_4c7e89,

    MeropriyatieId =
(Guid?)ps.GetParentHierarchy<_Meropriyatie_POF_fada13>(true).FirstOrDefault(
).Id,
    EtapPtrId =
(Guid?)ps.GetParentHierarchy<Etap_PTR_0eed50>(true).FirstOrDefault().Id,
```

```
  })
  .ToList()
  .Select(p => new{
    p.ID_проекта,
    p.Название_проекта,
    p.Производство_передел,
    p.Группа_оборудования,
    p.Отношение_к_части_ПТР,
    p.Код_МПОФ,
    p.ID_МПОФ,
    p.Марка_оборудования,
    p.Фирма_страна_производитель,
    p.Фирма_страна_поставщик,
    p.ID_узкого_места,
    p.Код_узкого_места,
    p.Кол_во,
    p.Бюджет_ПОФ,
    p.Дата_включения_в_ПТР,
    p.ФАКТ_платежи,
    p.ФАКТ_платежи_с_предыдущих_лет,
    p.Причины_отклонений,
    p.Предприятия_холдинга,
    p.Кол_во_Факт,

    p.План_сумма_1_кв,
    p.План_сумма_2_кв,
    p.План_сумма_3_кв,
    p.План_сумма_4_кв,

    p.Факт_сумма_1_кв,
    p.Факт_сумма_2_кв,
    p.Факт_сумма_3_кв,
    p.Факт_сумма_4_кв,

  КТ_ТЗ_план =
      kt_3
        .Where(k => k.Служебный_тип_для_оборудования ==
Classifier_Sluzhebnij_Tip_KT_dlya_oborudovaniya_SMR_611543.TZ_i_konkursnaya_
dokumentaciya_utverzhdeni_79c8892e_Id)
        .FirstOrDefault(k => k.EtapPtrId == p.EtapPtrId && k.EtapPtrId !=
null)?
        .BaselinePlanEndDate ??
      kt_3
        .Where(k => k.Служебный_тип_для_оборудования ==
Classifier_Sluzhebnij_Tip_KT_dlya_oborudovaniya_SMR_611543.TZ_i_konkursnaya_
dokumentaciya_utverzhdeni_79c8892e_Id)
        .FirstOrDefault(k => k.MeropriyatieId == p.MeropriyatieId &&
k.EtapPtrId == null)?
        .BaselinePlanEndDate,

  КТ_ТЗ_факт =
```

```
      kt_3
        .Where(k => k.Служебный_тип_для_оборудования ==
Classifier_Sluzhebnij_Tip_KT_dlya_oborudovaniya_SMR_611543.TZ_i_konkursnaya_
dokumentaciya_utverzhdeni_79c8892e_Id)
        .FirstOrDefault(k => k.EtapPtrId == p.EtapPtrId && k.EtapPtrId !=
null)?
        .ActualEndDate ??
      kt_3
        .Where(k => k.Служебный_тип_для_оборудования ==
Classifier_Sluzhebnij_Tip_KT_dlya_oborudovaniya_SMR_611543.TZ_i_konkursnaya_
dokumentaciya_utverzhdeni_79c8892e_Id)
        .FirstOrDefault(k => k.MeropriyatieId == p.MeropriyatieId &&
k.EtapPtrId == null)?
        .ActualEndDate,

  KT_Конкурс_план =
      kt_3
        .Where(k => k.Служебный_тип_для_оборудования ==
Classifier_Sluzhebnij_Tip_KT_dlya_oborudovaniya_SMR_611543.Konkurs_zavershen
_9b06d84d_Id)
        .FirstOrDefault(k => k.EtapPtrId == p.EtapPtrId && k.EtapPtrId !=
null)?
        .BaselinePlanEndDate ??
      kt_3
        .Where(k => k.Служебный_тип_для_оборудования ==
Classifier_Sluzhebnij_Tip_KT_dlya_oborudovaniya_SMR_611543.Konkurs_zavershen
_9b06d84d_Id)
        .FirstOrDefault(k => k.MeropriyatieId == p.MeropriyatieId &&
k.EtapPtrId == null)?
        .BaselinePlanEndDate,

  KT_Конкурс_факт =
      kt_3
        .Where(k => k.Служебный_тип_для_оборудования ==
Classifier_Sluzhebnij_Tip_KT_dlya_oborudovaniya_SMR_611543.Konkurs_zavershen
_9b06d84d_Id)
        .FirstOrDefault(k => k.EtapPtrId == p.EtapPtrId && k.EtapPtrId !=
null)?
        .ActualEndDate ??
      kt_3
        .Where(k => k.Служебный_тип_для_оборудования ==
Classifier_Sluzhebnij_Tip_KT_dlya_oborudovaniya_SMR_611543.Konkurs_zavershen
_9b06d84d_Id)
        .FirstOrDefault(k => k.MeropriyatieId == p.MeropriyatieId &&
k.EtapPtrId == null)?
        .ActualEndDate,

  KT_Договор_план =
      kt_3
        .Where(k => k.Служебный_тип_для_оборудования ==
Classifier_Sluzhebnij_Tip_KT_dlya_oborudovaniya_SMR_611543.Dogovor_zaklyuche
```

```
n_3a8ff8d9_Id)
        .FirstOrDefault(k => k.EtapPtrId == p.EtapPtrId && k.EtapPtrId !=
null)?
        .BaselinePlanEndDate ??
      kt_3
        .Where(k => k.Служебный_тип_для_оборудования ==
Classifier_Sluzhebnij_Tip_KT_dlya_oborudovaniya_SMR_611543.Dogovor_zaklyuche
n_3a8ff8d9_Id)
        .FirstOrDefault(k => k.MeropriyatieId == p.MeropriyatieId &&
k.EtapPtrId == null)?
        .BaselinePlanEndDate,

  КТ_Договор_факт =
      kt_3
        .Where(k => k.Служебный_тип_для_оборудования ==
Classifier_Sluzhebnij_Tip_KT_dlya_oborudovaniya_SMR_611543.Dogovor_zaklyuche
n_3a8ff8d9_Id)
        .FirstOrDefault(k => k.EtapPtrId == p.EtapPtrId && k.EtapPtrId !=
null)?
        .ActualEndDate ??
      kt_3
        .Where(k => k.Служебный_тип_для_оборудования ==
Classifier_Sluzhebnij_Tip_KT_dlya_oborudovaniya_SMR_611543.Dogovor_zaklyuche
n_3a8ff8d9_Id)
        .FirstOrDefault(k => k.MeropriyatieId == p.MeropriyatieId &&
k.EtapPtrId == null)?
        .ActualEndDate,

  КТ_Поставка_план =
      kt_3
        .Where(k => k.Служебный_тип_для_оборудования ==
Classifier_Sluzhebnij_Tip_KT_dlya_oborudovaniya_SMR_611543.Postavka_osusches
tvlena_d8967234_Id)
        .FirstOrDefault(k => k.PsId == p.ID_проекта && k.PsId != null)?
        .BaselinePlanEndDate,
  КТ_Поставка_факт =
      kt_3
        .Where(k => k.Служебный_тип_для_оборудования ==
Classifier_Sluzhebnij_Tip_KT_dlya_oborudovaniya_SMR_611543.Postavka_osusches
tvlena_d8967234_Id)
        .FirstOrDefault(k => k.PsId == p.ID_проекта && k.PsId != null)?
        .ActualEndDate,

  КТ_Монтаж_план =
      kt_3
        .Where(k => k.Служебный_тип_для_оборудования ==
Classifier_Sluzhebnij_Tip_KT_dlya_oborudovaniya_SMR_611543.Montazh_osuschest
vlen_73254f39_Id)
        .FirstOrDefault(k => k.PsId == p.ID_проекта && k.PsId != null)?
        .BaselinePlanEndDate,
  КТ_Монтаж_факт =
```

```
      kt_3
        .Where(k => k.Служебный_тип_для_оборудования ==
Classifier_Sluzhebnij_Tip_KT_dlya_oborudovaniya_SMR_611543.Montazh_osuschestvlen_73254f39_Id)
        .FirstOrDefault(k => k.PsId == p.ID_проекта && k.PsId != null)?
        .ActualEndDate,

  КТ_Ввод_в_Э_план =
      kt_3
        .Where(k => k.Служебный_тип_для_оборудования ==
Classifier_Sluzhebnij_Tip_KT_dlya_oborudovaniya_SMR_611543.Oborudovanie_vvedeno_v_ekspluataciyu_c2bc6e62_Id)
        .FirstOrDefault(k => k.PsId == p.ID_проекта && k.PsId != null)?
        .BaselinePlanEndDate,
  КТ_Ввод_в_Э_факт =
      kt_3
        .Where(k => k.Служебный_тип_для_оборудования ==
Classifier_Sluzhebnij_Tip_KT_dlya_oborudovaniya_SMR_611543.Oborudovanie_vvedeno_v_ekspluataciyu_c2bc6e62_Id)
        .FirstOrDefault(k => k.PsId == p.ID_проекта && k.PsId != null)?
        .ActualEndDate,

    p.Номер_договора,
  })
    .Select(p => new{
    p.ID_проекта,
    p.Название_проекта,
    p.Производство_передел,
    p.Группа_оборудования,
    p.Отношение_к_части_ПТР,
    p.Код_МПОФ,
    p.ID_МПОФ,
    p.Марка_оборудования,
    p.Фирма_страна_производитель,
    p.Фирма_страна_поставщик,
    p.ID_узкого_места,
    p.Код_узкого_места,
    p.Кол_во,
    p.Бюджет_ПОФ,
    p.Дата_включения_в_ПТР,
    p.ФАКТ_платежи,
    p.ФАКТ_платежи_с_предыдущих_лет,
    p.Причины_отклонений,
    p.Предприятия_холдинга,

    p.План_сумма_1_кв,
    p.План_сумма_2_кв,
    p.План_сумма_3_кв,
    p.План_сумма_4_кв,

    p.Факт_сумма_1_кв,
```

```
    p.Факт_сумма_2_кв,
    p.Факт_сумма_3_кв,
    p.Факт_сумма_4_кв,

    p.КТ_ТЗ_план,
  КТ_ТЗ_факт = p.КТ_ТЗ_факт < endDate ? p.КТ_ТЗ_факт : null,
    p.КТ_Конкурс_план,
КТ_Конкурс_факт = p.КТ_Конкурс_факт < endDate ? p.КТ_Конкурс_факт : null,
    p.КТ_Договор_план,
КТ_Договор_факт = p.КТ_Договор_факт < endDate ? p.КТ_Договор_факт : null,
    p.КТ_Поставка_план,
КТ_Поставка_факт = p.КТ_Поставка_факт < endDate ? p.КТ_Поставка_факт : null,
    p.КТ_Монтаж_план,
КТ_Монтаж_факт = p.КТ_Монтаж_факт < endDate ? p.КТ_Монтаж_факт : null,
    p.КТ_Ввод_в_Э_план,
КТ_Ввод_в_Э_факт = p.КТ_Ввод_в_Э_факт < endDate ? p.КТ_Ввод_в_Э_факт : null,
    p.Номер_договора,
    p.Кол_во_Факт,

  })
  .ToList()
  .OrderBy(o => o.Производство_передел)
  .ThenByDescending(o => o.Группа_оборудования)
  .ThenBy(o => o.Код_МПОФ);

return report22;
```

Объединение трудозатрат по ресурсам, проектам

```
var Today = DateTime.Today;
var ThisMonth = new DateTime (Today.Year, Today.Month, 1);
//var EndMonth = ThisMonth.AddYears(2).AddMonths(-1);
var IDprj = parameters.GetValueOrDefault<Guid>("RootProjectId", new
Guid("402b52cf-65d2-4c94-9e8b-f141e77aac55"));
var Person = dataContext.Persons.Select(a => new {a.Id,
a.LastName}).ToList();

var Costs = dataContext.ProjectCosts
            .Where( p => p.Project.Name != null && p.Value != 0 &&
p.Project.GetParentHierarchy<Project>(true).Any(x => x.Id == IDprj))
            .Select( p => new {
       Договор =
p.Project.GetParentHierarchy<Proekt_b31b01>(true).FirstOrDefault().Name,
       Ресурс = p.Person.LastName,
              ID_Ресурса = p.Person.Id,
       Трудозатраты = p.Value,
            }).ToList().GroupBy(u=>new {u.Договор, u.Ресурс}, (key, group) =>
new {
        Договор = key.Договор,
```

```
        Ресурс = key.Ресурс,
                 ID_Ресурса = group.FirstOrDefault().ID_Ресурса.ToString(),
        Трудозатраты = Math.Round(group.Sum(a => a.Трудозатраты))}).OrderBy(x
=> x.Договор).ToList();

var FactCost = dataContext.Uchet_vremeni_977ff0_List
            .Where (p => p.Project.Name != null && p.Summa_06ff93 != null &&
p.Project.GetParentHierarchy<Project>(true).Any(x => x.Id == IDprj))
            .Select(p => new{
       Договор =
p.Project.GetParentHierarchy<Proekt_b31b01>(true).FirstOrDefault().Name,
              ID_Ресурса = p.Resurs_caaaaa,
       Факт = p.Summa_06ff93,
            })
            .ToList()
            .Select(p => new {
              p.Договор,
       Ресурс = Person.Where(a => a.Id.ToString() ==
p.ID_Ресурса).FirstOrDefault().LastName,
              p.ID_Ресурса,
              p.Факт
            })
            .ToList()
            .GroupBy(u=>new {u.Договор, u.ID_Ресурса}, (key, group) => new {
        Договор = key.Договор,
        Ресурс = group.FirstOrDefault().Ресурс,
                 ID_Ресурса = key.ID_Ресурса.ToString(),
        Факт = Math.Round((double)group.Sum(a => a.Факт))}).OrderBy(x =>
x.Договор).ToList();

var CorrCost = dataContext.Korrektirovka_videlennih_chasov_421cbf_List
.Where (p => p.Project.Name != null && p.Summa_06ff93 != null &&
p.Project.GetParentHierarchy<Project>(true).Any(x => x.Id == IDprj))
            .Select(p => new{
       Договор =
p.Project.GetParentHierarchy<Proekt_b31b01>(true).FirstOrDefault().Name,
              ID_Ресурса = p.Resurs_caaaaa,
       Корректировка = p.Summa_06ff93,
       Дата = p.Date
            })
            .ToList()
            .Select(p => new {
              p.Договор,
       Ресурс = Person.Where(a => a.Id.ToString() ==
p.ID_Ресурса).FirstOrDefault().LastName,
              p.ID_Ресурса,
              p.Корректировка,
       Месяц_Корректировки = new DateTime (p.Дата.Value.Year,
p.Дата.Value.Month, 1)
            })
            .ToList()
```

```
            .GroupBy(u=>new {u.Договор, u.ID_Ресурса}, (key, group) => new {
        Договор = key.Договор,
        Ресурс = group.FirstOrDefault().Ресурс,
                ID_Ресурса = key.ID_Ресурса.ToString(),
        Корректировка = Math.Round((double)group.Sum(a => a.Корректировка)),
              }).OrderBy(x => x.Договор).ToList();

var JoinCost1 = FactCost.GroupJoin(Costs, x => new {x.ID_Ресурса, x.Договор},
u => new {u.ID_Ресурса, u.Договор}, (x,u) => new{
          Договор = x.Договор,
          Ресурс = x.Ресурс,
                    ID_Ресурса = x.ID_Ресурса,
          Трудозатраты = u.Sum(a =>a.Трудозатраты),
          Факт = x.Факт
                    }).ToList();
var JoinCost2 = Costs.GroupJoin(FactCost, x => new {x.ID_Ресурса, x.Договор},
u => new {u.ID_Ресурса, u.Договор}, (x,u) => new{
          Договор = x.Договор,
          Ресурс = x.Ресурс,
                    ID_Ресурса = x.ID_Ресурса,
          Трудозатраты = x.Трудозатраты,
          Факт = u.Sum(a=>a.Факт)
                    }).ToList();

var UnionCost1 = JoinCost1.Union(JoinCost2).ToList();

var JoinCost3 = UnionCost1.GroupJoin(CorrCost, x => new {x.ID_Ресурса,
x.Договор}, u => new {u.ID_Ресурса, u.Договор}, (x,u) => new{
          Договор = x.Договор,
          Ресурс = x.Ресурс,
                    ID_Ресурса = x.ID_Ресурса,
          Трудозатраты = x.Трудозатраты,
          Факт = x.Факт,
          Корректировка = u.Sum(a=>a.Корректировка),
                    }).ToList();
var JoinCost4 = CorrCost.GroupJoin(UnionCost1, x => new {x.ID_Ресурса,
x.Договор}, u => new {u.ID_Ресурса, u.Договор}, (x,u) => new{
          Договор = x.Договор,
          Ресурс = x.Ресурс,
                    ID_Ресурса = x.ID_Ресурса,
          Трудозатраты = u.Sum(a =>a.Трудозатраты),
          Факт = u.Sum(a => a.Факт),
          Корректировка = x.Корректировка,
                    }).ToList();

var UnionCost2 = JoinCost3.Union(JoinCost4).ToList();

var CostSum = UnionCost2

            .Select( p => new {
       Договор = p.Договор,
```

```
        Ресурс = "",
                ID_Ресурса = "",
        Трудозатраты = p.Трудозатраты,
        Факт = p.Факт,
        Корректировка = p.Корректировка,
             }).ToList().GroupBy(u=>new {u.Договор}, (key, group) => new {
         Договор = key.Договор,
         Ресурс = key.Договор,
                 ID_Ресурса = "",
         Трудозатраты = Math.Round(group.Sum(a => a.Трудозатраты)),
         Факт = Math.Round(group.Sum(a => a.Факт)),
                 }).OrderBy(x => x.Договор).ToList();

var Result = UnionCost2.Concat(CostSum).OrderBy(x => x.Ресурс).OrderBy(x =>
x.Договор).ToList();

return Result;
```