

**Акционерное общество «Всероссийский нефтегазовый научно-исследовательский институт имени академика А. П. Крылова»
(АО «ВНИИнефть»)**

**ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ТЕХНОЛОГИИ СИСТЕМНОГО
ВОЗДЕЙСТВИЯ НА ПЛАСТ (ПО «СВП»)**

информация, необходимая для установки
и эксплуатации программного обеспечения

Листов 15

Москва
2020 г.

СОДЕРЖАНИЕ

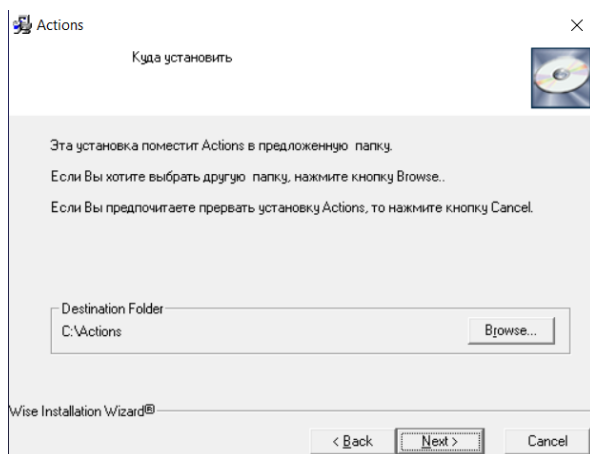
Введение.....	3
1. Информация, необходимая для установки и эксплуатации программного обеспечения. 4	
Листинг.....	5

Введение

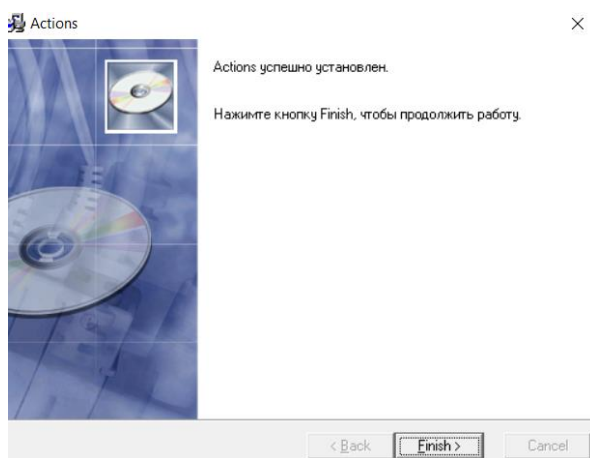
Документ содержит информацию, необходимую для установки и эксплуатации программного обеспечения. Также приведены минимальные требования к оборудованию и к программному окружению для функционирования ПО «СВП».

1. Информация, необходимая для установки и эксплуатации программного обеспечения

1. Загрузите на свой компьютер файл InstallActions.exe
2. Запустите файл InstallActions.exe.

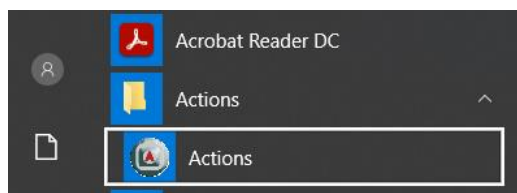


3. В этом диалоге можно поменять папку, куда будет помещена программа. Далее надо нажать на кнопку Next. Программа Actions будет установлена в указанную папку.



Кнопка Finish заканчивает работу инсталляции.

4. В стартовом меню Windows появится команда запуска Actions, которая запустит программное обеспечения «Системное воздействие на пласт» (ПО «СВП»).



Листинг

Листинг (частичный) исходного программного кода «(ПО «СВП»)» и основных методов обработки данных

```
#include "dlgtablet.h"
#include "QColor.h"
#include "GService.h"
#include "QService.h"
#include "OilObct.h"
#include "WellData.h"
#include "cdatetime\datecalc.h"

DlgTablet::DlgTablet(QWidget *parent)
    : QDialogMax(parent), m_pds(0)
{
    setupUi(this);

    _pPixmapGraphic = 0;

    setWindowTitle(QString::fromLocal8Bit("Планшет"));

    m_nMapWidth = 500;
    m_nMapHeight = 500;

    m_nHistoryCount = 12;

    _anTimeIndecis.SetGrowStep(100);

    tableWidget->setRowCount(2);
    tableWidget->setColumnCount(27);
    tableWidget->setEditTriggers(QAbstractItemView::NoEditTriggers);

    int i = 0;
    i = GQSetColumn(tableWidget, i, "Скважина-\nКандидат");
    i = GQSetColumn(tableWidget, i, "Объект");
    i = GQSetColumn(tableWidget, i, "Текущая\nприемистость,\nпкуб м/сут");
    i = GQSetColumn(tableWidget, i, "Реагирующие\nпскважины,\nпкоэф.\nпвзаимовлияния");
    i = GQSetColumn(tableWidget, i, "Прогнозируемая\nпДДН за\nп12 мес.,\nпт");
    i = GQSetColumn(tableWidget, i, "Прогнозируемое\nпСОПДВ за\nп12 мес.,\nпт");
    i = GQSetColumn(tableWidget, i, "Расчетный\nпобъем закачки\nпреагента,\nпкуб м");
    i = GQSetColumn(tableWidget, i, "Суммарный\nпранг");

    i = GQSetColumn(tableWidget, i, "ОИЗ из базы\nпдо ВПП,\nпт");
    i = GQSetColumn(tableWidget, i, "ОИЗ по ВНФ\nпдо ВПП,\nпт");
    i = GQSetColumn(tableWidget, i, "Прирост\nпНИЗ по ВНФ\nппосле ВПП,\nпт");

    i = GQSetColumn(tableWidget, i, "Стабильность\nпскв.-\nпкандидата\nп(3 мес.)");

    i = GQSetColumn(tableWidget, i, "Реагент\nпна 1 скв.-\nпопер.,\nптыс.руб.");
    i = GQSetColumn(tableWidget, i, "Закачка\nпзвеном\nпна 1 скв.-\nпопер.,\nптыс.руб.");
    i = GQSetColumn(tableWidget, i, "Научное\nпсопровождение работ\nпАО «ВНИИнефть»\nп(1 скв.-опер.),\nптыс.руб.");
    i = GQSetColumn(tableWidget, i, "Итого\nпзатраты на\nп1 скв.-опер.,\nптыс.руб.");
    i = GQSetColumn(tableWidget, i, "Экономия затрат\nпза счет сокращения\nпотборов попутно\nпдоб. воды,\nпмлн.руб.");
    i = GQSetColumn(tableWidget, i, "Чистый доход\nпнедропользователя\nпза счет дополнительной\nпдобычи нефти,\nпмлн.руб.");
    i = GQSetColumn(tableWidget, i, "Итоговый\nпчистый доход\nпнедропользователя,\nпмлн.руб.");
    i = GQSetColumn(tableWidget, i, "Расчетная\nпнаилучшая\nптехнология\nппо рейтингу");
```

```

i = GQSetColumn(tableWidget, i, "Дата\побработки,\пгод");
i = GQSetColumn(tableWidget, i, "Объем\пзакачки,\пкуб м");

i = GQSetColumn(tableWidget, i, "Вид\птехнологии");
i = GQSetColumn(tableWidget, i, "Длительность\пэффекта,\пмес");
i = GQSetColumn(tableWidget, i, "ДДН,\пт");
i = GQSetColumn(tableWidget, i, "СОПДВ,\пт");
i = GQSetColumn(tableWidget, i, "Прирост\пдебита\пнефти,\пт/сут");

tableWidget->horizontalHeader()->setSectionsMovable(true);
}

DlgTablet::~DlgTablet()
{
}

int DlgTablet::init()
{
    _adOil.Zero(m_nHistoryCount);
    _adFluid.Zero(m_nHistoryCount);
    GTDoubleArray adWaterCut;
    adWaterCut.Zero(m_nHistoryCount);

    _adOilBaseForecastle.Zero(13);
    _adFluidBaseForecastle.Zero(13);
    GTDoubleArray adWaterBaseForecastle;
    adWaterBaseForecastle.Zero(13);

    _adOilVPPForecastle.Zero(13);
    _adFluidVPPForecastle.Zero(13);
    GTDoubleArray adWaterVPPForecastle;
    adWaterVPPForecastle.Zero(13);

    _adXMonthForecast.SetGrowStep(100);
    _adXMonthHistory.SetGrowStep(100);

    int nSortColumn = 7;

    if (m_pds)
    {
        OilObject *pObject = m_pds->GetOilObject();
        if (pObject == 0) return -1;

        pObject->CalculateEconomicInit();

        int nYear, nMonth;
        m_pds->GetCurrentMapTime(nYear, nMonth);

        GLayers layers;
        GStatisticsItem statisticsItem;

        int nRow = m_pds->RegionCount();
        if (nRow > 0)
        {
            m_pds->RegionGetTimeIndices(m_nHistoryCount, &_anTimeIndecis);

            int nYear, nMonth;

            tableWidget->setRowCount(nRow + 1);

            GTDoubleArray adTotal;
            adTotal.Zero(100);

```

```

char szName[1024];
char szNameL[1024];
char sz[1024];
for (int i = 0; i < nRow; i++)
{
    WellData *pW = m_pds->RegionGetWell(i);
    if (pW == 0) continue;

    m_pds->RegionGetTime(_anTimeIndecis.Last(), nYear, nMonth);
    m_pds->GetInjectLayers(pW, nYear, nMonth, m_layers);

    GTIntegerArray aIDLayers;
    m_pds->GetActiveLayers(layers, aIDLayers);

    GTIntegerArray aIDGeoLayers;
    m_pds->GetActiveLayersAlias(layers, aIDGeoLayers);

    GStatistics::Fill(m_StatisticsItem);
    GStatistics *pStat = m_pds->GetStatistics() ;
    if (pStat)
    {
        pStat->Get(m_pds->GetReservoir(pW->m_sName), m_layers,
m_StatisticsItem);
    }

    GWellsRegion *pWR = m_pds->GetRegion(pW->m_sName);

    int nJ = 0;

    m_pds->RegionName(i, szName);
    adTotal.AddValue(nJ, 1);
    GQSetCell(tableWidget, i, nJ++, szName);
    int nInjectCount = 1;

    char szLayers[1024];

    m_pds->GetInjectWellLayers(pW->m_sName, szLayers);
    GQSetCell(tableWidget, i, nJ++, szLayers);

    // Текущая приемистость, м3/сут
    GQSetCell(tableWidget, i, nJ++, pW-
>m_dWaterInjectRate.ForceValue(), 0);

    szName[0] = 0;
    szNameL[0] = 0;
    int nProdCount = m_pds->RegionGetProdCount(i);
    for (int iw = 0; iw < nProdCount; iw++)
    {
        int j = (pW->m_aProds)[iw];
        m_pds->WellName(j, sz);
        strcat (szName, sz);
        strcat (szNameL, sz);
        WellData *pWell = pObject->FindWell(sz);
        if (pWell)
        {
            GLinkItem *pIL = pWell->FindInjectLink(pW-
>m_sName);

            if (pIL)
            {
                char buf[32];
                sprintf(buf, "(%.11f)", pIL-
>m_dvProcent.ForceValue() * 0.01);

                strcat (szNameL, buf);
            }
        }
    }
}

```

```

    }
    if (iw != nProdCount - 1)
    {
        strcat (szName, ",");
        strcat (szNameL, ",");
        if (((iw + 1) % 3) == 0)
        {
            strcat (szName, "\n");
            strcat (szNameL, "\n");
        }
    }
}
GQSetCell(tableWidget, i, nJ++, szNameL);

double dOilIncrease = 0;
double dWaterDecrease = 0;
if (pWR && pWR->m_dvDNNOilIncrease.IsEnabled())
{
    dOilIncrease = pWR->m_dvDNNOilIncrease.ForceValue();
    adTotal.AddValue(nJ, dOilIncrease);
    GQSetCell(tableWidget, i, nJ++, dOilIncrease, 0);
    dWaterDecrease = dOilIncrease *
m_StatisticsItem.m_dFractionWaterDecreaseTotal;
    adTotal.AddValue(nJ, dWaterDecrease);
    GQSetCell(tableWidget, i, nJ++, dWaterDecrease, 0);
}
else
{
    GQSetCell(tableWidget, i, nJ++, "");
    GQSetCell(tableWidget, i, nJ++, "");
}

// Расчетный объем закачки реагента, м3

double dWaterInj = 0;
if (pStat)
{
    double dwt = 0;
    m_pds->GetNeighborWaterInjectRate(pW, dwt);
    dWaterInj = GDivide(dwt,
m_StatisticsItem.m_dCoeffWaterInject);
}
adTotal.AddValue(nJ, dWaterInj);
GQSetCell(tableWidget, i, nJ++, dWaterInj, 0);

double dFluidRate, dOilRate;
m_pds->RegionGetProdRate(i, dFluidRate, dOilRate);

nInjectCount = 1;

double dWaterRate;
m_pds->RegionGetInjectRate(i, dWaterRate);

double dNIZ, dOIZ;
double dFluidTotal;
m_pds->RegionGetNIZOIZ(i, dNIZ, dOIZ, dFluidTotal); // Внутри
границы

double dOilTotalR = dNIZ - dOIZ;

double dRange = m_pds->CalculateTotalRange(pW, aIDGeoLayers,
nInjectCount, nProdCount, dFluidRate, dOilRate, dWaterRate, dNIZ, dOilTotalR);
GQSetCell(tableWidget, i, nJ++, dRange);

```



```

        GQSetCell(tableWidget, i, nJ++, dOIZ);

        if (pWR->m_dvNIZbyWaterOilFactor.IsEnabled() && pWR-
>m_dvNIZbyWaterOilFactorVPP.IsEnabled())
        {
            double dOIZWaterOilFactor = 0;
            if (pWR->m_dvOIZbyWaterOilFactor.IsEnabled())
            {
                dOIZWaterOilFactor = pWR-
>m_dvOIZbyWaterOilFactor.ForceValue();
            }
            GQSetCell(tableWidget, i, nJ++, dOIZWaterOilFactor);
            GQSetCell(tableWidget, i, nJ++, pWR-
>m_dvNIZbyWaterOilFactorVPP.ForceValue() - pWR->m_dvNIZbyWaterOilFactor.ForceValue());
        }
        else
        {
            nJ += 2;
        }

    if (pWR->m_nStability == 2)
    {
        GQSetCell(tableWidget, i, nJ++, "Стабильный");
    }
    else
    {
        if (pWR->m_nStability == 1)
        {
            GQSetCell(tableWidget, i, nJ++, "Нестабильный");
        }
        else
        {
            GQSetCell(tableWidget, i, nJ++, "");
        }
    }

    if (pWR && pWR->m_dvDNNOilIncrease.IsEnabled())
    {
        double d;
        d = pObject->CalculateEconomicReagentPrice(dWaterInj) *
0.001;

        adTotal.AddValue(nJ, d);
        GQSetCell(tableWidget, i, nJ++, d);

        d = pObject->m_dPriceInjectGroup;
        adTotal.AddValue(nJ, d);
        GQSetCell(tableWidget, i, nJ++, d);

        d = pObject->m_dPriceVNII;
        adTotal.AddValue(nJ, d);
        GQSetCell(tableWidget, i, nJ++, d);

        double dPrice = pObject-
>CalculateEconomicRegionCost(dWaterInj);
        d = dPrice * 0.001;
        adTotal.AddValue(nJ, d);
        GQSetCell(tableWidget, i, nJ++, d);

        d = pObject-
>CalculateEconomicWaterCostSavings(dWaterDecrease) * 0.000001;
        adTotal.AddValue(nJ, d);
        GQSetCell(tableWidget, i, nJ++, d);
    }
}

```

```

dPrice) * 0.000001;

d = pObject->CalculateEconomicPureProfit(dOilIncrease,
adTotal.AddValue(nJ, d);
GQSetCell(tableWidget, i, nJ++, d);

d = pObject-
>CalculateEconomicTotalPureProfit(dOilIncrease, dWaterDecrease, dPrice) * 0.000001;
adTotal.AddValue(nJ, d);
GQSetCell(tableWidget, i, nJ++, d);
}
else
{
nJ += 7;
}

if (pStat)
{
GToArray<GEffectItem> aEffects;
GToArray<GEffectItem> aSortedEffects;
pStat->GetEffects(m_pds->GetReservoir(pW->m_sName), pW-
>m_sName, m_layers, aEffects);

GStatistics::SortTime(aEffects);
GStatistics::Sort(aEffects, aSortedEffects);

char szTTitle1[1024];
char szTTitle2[1024];
char szTTitle3[1024];
char szTTitle4[1024];
char szTTitle5[1024];
char szTTitle6[1024];
char szTTitle7[1024];
char szTTitle8[1024];
char buf[1024];

szTTitle1[0] = 0;
szTTitle2[0] = 0;
szTTitle3[0] = 0;
szTTitle4[0] = 0;
szTTitle5[0] = 0;
szTTitle6[0] = 0;
szTTitle7[0] = 0;
szTTitle8[0] = 0;

if (aSortedEffects.GetCount() > 0)
{
for (int ip = 0; ip < aSortedEffects.GetCount();
ip++)
{
GEffectItem *pIt = aSortedEffects[ip];
if (ip != 0)
{
strcat(szTTitle1, "\n");
}
strcat(szTTitle1, pIt->m_szTechnology);
}
}
GQSetCell(tableWidget, i, nJ++, szTTitle1);

for (int ip = 0; ip < aEffects.GetCount(); ip++)
{
GEffectItem *pIt = aEffects[ip];
if (ip != 0)
{

```

```

        strcat(szTTitle2, "\n");
        strcat(szTTitle3, "\n");
        strcat(szTTitle4, "\n");
        strcat(szTTitle5, "\n");
        strcat(szTTitle6, "\n");
        strcat(szTTitle7, "\n");
        strcat(szTTitle8, "\n");
    }

    sprintf(buf, "%ld", (int) pIt-
>m_dvYear.ForceValue());
        strcat(szTTitle2, buf);

    sprintf(buf, "%.0lf", pIt-
>m_dvInjectWater.ForceValue());
        strcat(szTTitle3, buf);

    sprintf(buf, "%.0lf", pIt-
>m_dvTimeLong.ForceValue());
        strcat(szTTitle4, buf);

    sprintf(buf, "%.0lf", pIt-
>m_dvOilExtra.ForceValue());
        strcat(szTTitle5, buf);

    sprintf(buf, "%.0lf", pIt-
>m_dvWaterDecrease.ForceValue());
        strcat(szTTitle6, buf);

    double d = GDivide(pIt-
>m_dvOilExtra.ForceValue(), pIt->m_dvTimeLong.ForceValue() * 30.4);
    sprintf(buf, "%.1lf", d);
    strcat(szTTitle7, buf);

    strcat(szTTitle8, pIt->m_szTechnology);
}
GQSetCell(tableWidget, i, nJ++, szTTitle2);
GQSetCell(tableWidget, i, nJ++, szTTitle3);
GQSetCell(tableWidget, i, nJ++, szTTitle8);
GQSetCell(tableWidget, i, nJ++, szTTitle4);
GQSetCell(tableWidget, i, nJ++, szTTitle5);
GQSetCell(tableWidget, i, nJ++, szTTitle6);
GQSetCell(tableWidget, i, nJ++, szTTitle7);
}

if (pWR && pWR->m_dvDNNOilIncrease.IsEnabled() && pWR-
>m_aForecast.GetCount() > 0)
{
    double dWater = 0;
    for (int it = 0; it < _anTimeIndecis.GetCount(); it++)
    {
        m_pds->RegionGetTime(_anTimeIndecis[it], nYear,
nMonth);

        double dFluid, dOil, dFluidTotal, dOilTotal;
        double dWork;
        switch (pWR->m_nMethod)
        {
            case REG_METHOD_WATER_OIL_FACTOR:
                m_pds->RegionNeighborProduct(pW, nYear,
nMonth, dFluid, dOil, dFluidTotal, dOilTotal, dWork, 0);
                break;
            default:
                m_pds->RegionNeighborProduct(pW, nYear,
nMonth, dFluid, dOil, dFluidTotal, dOilTotal, dWork, 1);
                break;
        }
    }
}

```

```

    }

    _adOil.AddValue(it, dOil);
    _adFluid.AddValue(it, dFluid);

    dWater = dFluid - dOil;
}

double dDays = CDateCalc::DaysInMonth(nMonth,
CDateCalc::IsLeapYear(nYear));
double dWaterRate = GDivide(dWater, dDays);

int nNextYear = nYear;
int nNextMonth = nMonth;
for (int it = 0; it < pWR->m_aForecast.GetCount(); it++)
{
    GGetNextDate(nNextYear, nNextMonth);
    double dD = CDateCalc::DaysInMonth(nNextMonth,
CDateCalc::IsLeapYear(nNextYear));

    double dW = dWaterRate * dD;
    adWaterBaseForecastle.AddValue(it + 1, dW);
    _adOilBaseForecastle.AddValue(it + 1, pWR-
    _adFluidBaseForecastle.AddValue(it + 1, pWR-

    double dDNN = pWR-
    >m_dvDNNOilIncrease.ForceValue() * m_StatisticsItem.m_adFractionOilForecast[it];
    _adOilVPPForecastle.AddValue(it + 1, pWR-
    >m_aForecast[it] + dDNN);
    _adFluidVPPForecastle.AddValue(it + 1, pWR-
    >m_aForecast[it] + dDNN + dW);
}
}
}

tableWidget->sortByColumn(nSortColumn, Qt::DescendingOrder);

QTableWidgetItem *pTWI;
int nJ = 0;
sprintf(sz, "Итого %ld шт", (int) adTotal[0]);
pTWI = GQSetCell(tableWidget, nRow, nJ, sz);
pTWI->setTextColor(Qt::red);
nJ = 4;
pTWI = GQSetCell(tableWidget, nRow, nJ, adTotal[nJ], 0);
pTWI->setTextColor(Qt::red);
nJ++;
pTWI = GQSetCell(tableWidget, nRow, nJ, adTotal[nJ], 0);
pTWI->setTextColor(Qt::red);
nJ++;
pTWI = GQSetCell(tableWidget, nRow, nJ, adTotal[nJ], 0);
pTWI->setTextColor(Qt::red);
nJ = 12;
pTWI = GQSetCell(tableWidget, nRow, nJ, adTotal[nJ]);
pTWI->setTextColor(Qt::red);
nJ++;
pTWI = GQSetCell(tableWidget, nRow, nJ, adTotal[nJ]);
pTWI->setTextColor(Qt::red);
nJ++;
pTWI = GQSetCell(tableWidget, nRow, nJ, adTotal[nJ]);
pTWI->setTextColor(Qt::red);
nJ++;
pTWI = GQSetCell(tableWidget, nRow, nJ, adTotal[nJ]);
pTWI->setTextColor(Qt::red);

```

```

nJ++;
pTWI = GQSetCell(tableWidget, nRow, nJ, adTotal[nJ]);
pTWI->setTextColor(Qt::red);
nJ++;
pTWI = GQSetCell(tableWidget, nRow, nJ, adTotal[nJ]);
pTWI->setTextColor(Qt::red);
nJ++;
pTWI = GQSetCell(tableWidget, nRow, nJ, adTotal[nJ]);
pTWI->setTextColor(Qt::red);

_adOilBaseForecastle.Change(0, _adOil.Last());
_adFluidBaseForecastle.Change(0, _adFluid.Last());
_adOilVPPForecastle.Change(0, _adOil.Last());
_adFluidVPPForecastle.Change(0, _adFluid.Last());

_adWaterCutHistory.SetGrowStep(_anTimeIndecis.GetCount());
for (int it = 0; it < _anTimeIndecis.GetCount(); it++)
{
    m_pds->RegionGetTime(_anTimeIndecis[it], nYear, nMonth);
    double dX = GetMounthCount(nYear, nMonth);
    _adXMonthHistory.Add(dX);

    _adWaterCutHistory.Add(GDivide(_adFluid[it] - _adOil[it],
_adFluid[it]) * 100.0);
}

m_pds->RegionGetTime(_anTimeIndecis.Last(), nYear, nMonth);
int nNextYear = nYear;
int nNextMonth = nMonth;

_adWaterCutVPPForecastle.SetGrowStep(_adOilVPPForecastle.GetCount());
_adWaterCutBaseForecastle.SetGrowStep(_adOilVPPForecastle.GetCount());

for (int it = 0; it < _adOilVPPForecastle.GetCount(); it++)
{
    double dX = GetMounthCount(nNextYear, nNextMonth);
    _adXMonthForecast.Add(dX);
    GGetNextDate(nNextYear, nNextMonth);

    _adWaterCutBaseForecastle.Add(GDivide(_adFluidBaseForecastle[it] -
_adOilBaseForecastle[it], _adFluidBaseForecastle[it]) * 100.0);
    _adWaterCutVPPForecastle.Add(GDivide(_adFluidVPPForecastle[it]
- _adOilVPPForecastle[it], _adFluidVPPForecastle[it]) * 100.0);
}

createGraphic();

GQStretch(tableWidget);
GQSetColors(tableWidget);
}
}

tableWidget->setStyleSheet("QHeaderView::section { color:black; background-
color:#98FB98; }");
tableWidget->resizeRowsToContents();

QPixmap pix = m_pixmap.scaled(m_nMapWidth, m_nMapHeight,
Qt::KeepAspectRatioByExpanding, Qt::SmoothTransformation);
labelMap->setPixmap(pix);

```

```

        return 0;
    }

int DlgTablet::createGraphic()
{
    if (_pPixmapGraphic)
    {
        delete _pPixmapGraphic;
        _pPixmapGraphic = 0;
    }

    QQGraphic gqg;
    QQGraphic *pgqg = &gqg;
    GTDoubleArray adXVPP;
    GTDoubleArray adYVPP;
    adXVPP.SetGrowStep(2);
    adYVPP.SetGrowStep(2);

    adXVPP.Add(_adXMonthHistory.Last());
    adXVPP.Add(_adXMonthHistory.Last());

    adYVPP.Add(80);
    adYVPP.Add(100);

    pgqg->AddDataSeries(_adXMonthForecast.GetCount(),
_adXMonthForecast.m_pArray, _adOilBaseForecastle.m_pArray, "Базовый прогноз нефти, т",
SERIES_TYPE_DASH_LINE);
    pgqg->setSeriesColor(pgqg->m_apDataSeies.GetCount() - 1, QColor(200,
100, 0));

    pgqg->AddDataSeries(_adXMonthForecast.GetCount(),
_adXMonthForecast.m_pArray, _adOilVPPForecastle.m_pArray, "ВПП прогноз нефти, т",
SERIES_TYPE_POINT_AND_LINE);
    pgqg->setSeriesColor(pgqg->m_apDataSeies.GetCount() - 1, QColor(200,
100, 0));

    pgqg->AddDataSeries(_adXMonthHistory.GetCount(),
_adXMonthHistory.m_pArray, _adOil.m_pArray, "Добыча нефти, т",
SERIES_TYPE_POINT_AND_LINE);
    pgqg->setSeriesColor(pgqg->m_apDataSeies.GetCount() - 1, QColor(100,
50, 0));

    pgqg->AddDataSeries(_adXMonthForecast.GetCount(),
_adXMonthForecast.m_pArray, _adFluidBaseForecastle.m_pArray, "Базовый прогноз жидкости,
т", SERIES_TYPE_DASH_LINE);
    pgqg->setSeriesColor(pgqg->m_apDataSeies.GetCount() - 1,
QColor(Qt::green));
    pgqg->m_apDataSeies.Last()->SetY(1);

    pgqg->AddDataSeries(_adXMonthForecast.GetCount(),
_adXMonthForecast.m_pArray, _adFluidVPPForecastle.m_pArray, "ВПП прогноз жидкости, т",
SERIES_TYPE_POINT_AND_LINE);
    pgqg->setSeriesColor(pgqg->m_apDataSeies.GetCount() - 1,
QColor(Qt::green));
    pgqg->m_apDataSeies.Last()->SetY(1);
}

```

```

        pgqg->AddDataSeries(_adXMonthHistory.GetCount(),
_adXMonthHistory.m_pArray, _adFluid.m_pArray, "Добыча жидкости, т",
SERIES_TYPE_POINT_AND_LINE);
        pgqg->setSeriesColor(pgqg->m_apDataSeies.GetCount() - 1,
QColor(Qt::darkGreen));
        pgqg->m_apDataSeies.Last()->SetY(1);

        pgqg->AddDataSeries(_adXMonthForecast.GetCount(),
_adXMonthForecast.m_pArray, _adWaterCutBaseForecastle.m_pArray, "Базовый прогноз обводн,
%", SERIES_TYPE_DASH_LINE);
        pgqg->setSeriesColor(pgqg->m_apDataSeies.GetCount() - 1,
QColor(Qt::cyan));
        pgqg->m_apDataSeies.Last()->SetY(2);

        pgqg->AddDataSeries(_adXMonthForecast.GetCount(),
_adXMonthForecast.m_pArray, _adWaterCutVPPForecastle.m_pArray, "ВПП прогноз обводн, %",
SERIES_TYPE_POINT_AND_LINE);
        pgqg->setSeriesColor(pgqg->m_apDataSeies.GetCount() - 1,
QColor(Qt::cyan));
        pgqg->m_apDataSeies.Last()->SetY(2);

        pgqg->AddDataSeries(_adXMonthHistory.GetCount(),
_adXMonthHistory.m_pArray, _adWaterCutHistory.m_pArray, "Обводненность, %",
SERIES_TYPE_POINT_AND_LINE);
        pgqg->setSeriesColor(pgqg->m_apDataSeies.GetCount() - 1,
QColor(Qt::blue));
        pgqg->m_apDataSeies.Last()->SetY(2);

        pgqg->AddDataSeries(adXVPP.GetCount(), adXVPP.m_pArray,
adYVPP.m_pArray, "ВПП");
        pgqg->setSeriesColor(pgqg->m_apDataSeies.GetCount() - 1,
QColor(Qt::red));
        pgqg->m_apDataSeies.Last()->SetY(2);

        gqg.SetYRightAxeMinValue(0.0);
        gqg.SetY2RightAxeMinValue(80);

        gqg.SetYMaxMarginFactor(0.4);
        gqg.SetY1MaxMarginFactor(0.1);
        // gqg.SetY2MaxMarginFactor(0.2);
        gqg.SetNumberFormat("%.0lf");

        pgqg->SetXaxeTitle("Месяцы");
        pgqg->SetYaxeTitle("Нефть, т");
        pgqg->SetRightY1axeTitle("Жидкость, т");
        pgqg->SetRightY2axeTitle("Обводненность, %");
        pgqg->SetYaxeMinValue(0.0);
        pgqg->SetAxeTypes(GQGraphic::SCALE_MONTH, GQGraphic::SCALE_STANDARD);

        _pPixmapGraphic = pgqg->QCreateCurveIcon(labelGraphic->size());
        if (_pPixmapGraphic)
        {
            labelGraphic->setPixmap(*_pPixmapGraphic);
        }

        return 0;
}
void DlgTablet::resizeEvent(QResizeEvent *event)
{
    createGraphic();
    QWidget::resizeEvent(event);
}

```