ДОДАТОК А

КОД ПРОГРАМИ

ГЮІК.502810.0103 – 01 12 01

(позначення документу)

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

"ЗАТВЕРДЖУЮ"
керівник атестаційної роботи
_____ проф. Мінухін С. В.
(підпис)

РОЗРОБКА КОМПОНЕНТУ СИСТЕМИ ОБРОБКИ ДАНИХ З ВИКОРИСТАННЯМ САМООРГАНІЗОВНИХ КАРТ Т. КОХОНЕНА

Текст програми
ЛИСТ ЗАТВЕРДЖЕННЯ
ГЮІК.502810.0103 – 01 12 01

РОЗРОБИВ:
ст. гр. СПРм-19-1
Дмитрієв О. В.

2020

РОЗРОБКА СИСТЕМИ ЗБЕРІГАННЯ ТА АНАЛІЗУ ЯКОСТІ ЗВІТНОСТІ ПРО ВИКОНАННЯ ЗАВДАНЬ ВИРОБНИЧИМ ПЕРСОНАЛОМ(НА ПРИКЛАДІ ГІРНИЧОДОБУВНОГО ПІДПРИЄМСТВА)

Текст програми

ГЮІК.502810.0103 – 01 12 01

Аркушів 14

2020

Лістинг А.1 – Функції аналізу якості звітності про виконання завдань виробничим персоналом

```
private void GetAllErrorData(out IEnumerable<ReportDataRowDescriptor>
                                periodicityResult
    , out IEnumerable<ReportDataRowDescriptor> qualityResult
    , ShaftMonitoringDBContext context = null
    , Position position = null
    , SpatialModel_Type modelType = null
    , DateTime? from = null
    , DateTime? to = null
    , MineElement selectedElement = null)
{
    List<ReportDataRowDescriptor> resultPeriodicityData =
                                        new List<ReportDataRowDescriptor>();
    List<ReportDataRowDescriptor> resultQualityData =
                                        new List<ReportDataRowDescriptor>();

    context = context == null ? this.DBManager.GetNewDBContext() : context;
    modelType = modelType == null ? this.SelectedModelType : modelType;
    selectedElement = selectedElement == null ?
                this.SelectedItem.Item as MineElement : selectedElement;

    if(modelType != this.defaultModelType)
    {
        modelType = context.SpatialModel_Types.FirstOrDefault(x => x.ID ==
modelType.ID);
    }
    selectedElement = context.MineElements
                            .FirstOrDefault(x => x.ID ==
                        selectedElement.ID);
    DateTime dateForm = from == null ? this.DateFrom.Value : from.Value;
    DateTime dateTo = to == null ? this.DateTo.Value : to.Value;

    Dictionary<string, object> sqlParams = new Dictionary<string,
object>(){
        {"MineElementID", selectedElement.ID} };
    StringBuilder builder = new StringBuilder(defaultMineElementQuery);
    if (modelType != null && modelType != this.defaultModelType)
    {
         builder.Append(" WHERE c.MineCategoryId = @MineCategoryID ");
         sqlParams["MineCategoryID"] = modelType.MineElementCateryID;
    }

    List<MineElement> mineElements = new List<MineElement>();
    using (RawSqlHelper sqlHelper = new RawSqlHelper(context.Connection,
                                            builder.ToString(), sqlParams))
    {
    IEnumerable<int> mineIDs = sqlHelper.Select(x => (int)x["ID"]);
    mineIDs = mineIDs.Distinct().ToArray();
    mineElements.AddRange(context.MineElements.Where(x =>
```

```
                                        mineIDs.Contains(x.ID)).ToArray());
    }

    foreach (MineElement element in mineElements)
    {
        if (modelType != this.defaultModelType && modelType != null)
        {
            if (modelType != this.defaultModelType && modelType != null)
            {
                typesToProcess = new SpatialModel_Type[] { modelType };
            }
            else
            {
                typesToProcess = element.MineElementCategory
                                        .SpatialModel_Types.ToArray();
            }
            foreach (SpatialModel_Type type in typesToProcess)
            {
                IEnumerable<ReportDataRowDescriptor> periodicityData =
null;
                IEnumerable<ReportDataRowDescriptor> qualityData = null;
                try
                {
                    this.GetModelTypeReportData(type, element,
                                                selectedElement, dateForm,
                                                dateTo, context,
                                                out periodicityData,
                                                out qualityData);
                    resultPeriodicityData.AddRange(periodicityData);
                    resultQualityData.AddRange(qualityData);
                }
                catch (Exception exc)
                {}
            }
        }
    }

    periodicityResult = resultPeriodicityData;
    qualityResult = resultQualityData;
}

private void GetModelTypeReportData(SpatialModel_Type type
                                    , MineElement currentElement
                                    , MineElement rootElement
                                    , DateTime dateFrom
                                    , DateTime dateTo
                                    , ShaftMonitoringDBContext context
                  , out IEnumerable<ReportDataRowDescriptor>
periodicityData
                  , out IEnumerable<ReportDataRowDescriptor> qualityData)
{
```

```
    List<ReportDataRowDescriptor> periodicityResult = new
List<ReportDataRowDescriptor>();
    List<ReportDataRowDescriptor> qualityResult = new
List<ReportDataRowDescriptor>();

    if ((type.HasActivePeriodsEverForMineElement(currentElement.ID) &&
currentElement.HasActivePeriodsEver()) == false)
    {
        periodicityData = periodicityResult;
        qualityData = qualityResult;
        return;
    }

    IEnumerable<SpatialModel> models = null;

    IModelTypePeriodisity periodisityManager =
TypePeriodicityManagerFabric.GetPeriodisityManager(type.Periodisity);
    if (periodisityManager is NonePeriodicityManager)
    {
        long mainBrush =
currentElement.GetActualStateForDate(dateTo).Color;
        mainBrush = mainBrush == SpatialModel_TypeState.FinishedStateColor
? System.Windows.Media.Colors.White.ToUint() : mainBrush;
        //getting all models for current model type and current mine
element that was added while model type was active
        models = type.GetModelsForActivePeriods(currentElement.ID);
        //getting all models for current model type and current mine
element that was added while mine element was active
        IEnumerable<int> modelIDs =
currentElement.GetModelsForActivePeriods(x => x.MineModelTypeID ==
type.ID).Select(x => x.ID);
        //by intersecting this values we have models that was added while
bouth mine element and model type were active
        models = models.Where(x=>modelIDs.Contains(x.ID));

        ReportDataRowDescriptor periodDesc = new ReportDataRowDescriptor()
            {
                MineElement =
currentElement.GetStringPathToElement(rootElement),
                ModelType = type.Name,
                Period =
ResourcesManager.Current.Local.GetString("MSRW_allTimePeriodName",
AssemblyResourceInfo.Path),
                ErrorType =
ResourcesManager.Current.Local.GetString("MSRW_noModelsErrorGroup",
AssemblyResourceInfo.Path),
                MainColor = mainBrush,
                Periodisity = periodisityManager.ToString(),
                MineElementID = currentElement.ID
            };
        if(!models.Any())
```

```
        {
            periodicityResult.Add(periodDesc);
        }
        else if(!(models.Where(x=>x.Date>=dateFrom &&
x.Date<=dateTo).Any()))
        {
            periodDesc.ErrorColor = MinorErrorColor;
            periodicityResult.Add(periodDesc);
        }
        models = models.Where(x => x.Date >= dateFrom && x.Date <=
dateTo).ToArray();
        foreach(SpatialModel model in models)
        {
            ReportDataRowDescriptor error = new ReportDataRowDescriptor()
            {
                MineElement =
currentElement.GetStringPathToElement(rootElement),
                ModelType = type.Name,
                Period =
ResourcesManager.Current.Local.GetString("MSRW_allTimePeriodName",
AssemblyResourceInfo.Path),
                Model = model.FullModelName,
                MainColor = mainBrush,
                Periodisity = periodisityManager.ToString(),
                MineElementID = currentElement.ID
            };
            if(!model.SpatialModel_OriginalItems.Any() &&
!model.SpatialModel_ConvertedItems.Any())
            {
                error.ErrorType =
ResourcesManager.Current.Local.GetString("MSRW_modelsWithoutFilesErrorGroup
", AssemblyResourceInfo.Path);
                qualityResult.Add(error);
                continue;
            }

            if(model.AreAllRequiredFilesPresent() == false)
            {
                error.ErrorType =
ResourcesManager.Current.Local.GetString("MSRW_modelsWithoutRequiredFilesEr
rorGroup", AssemblyResourceInfo.Path);
                qualityResult.Add(error);
                continue;
            }

            if(type.AreConvertedFilesRequired &&
!model.SpatialModel_ConvertedItems.Any())
            {
                error.ErrorType =
ResourcesManager.Current.Local.GetString("MSRW_modelsWithoutConvertedErrorG
roup", AssemblyResourceInfo.Path);
```

```
                qualityResult.Add(error);
                continue;
            }
        }
        qualityResult.ForAll(x => { x.DateTo = dateTo; x.DateFrom =
dateFrom; });
        qualityData = qualityResult;
        periodicityResult.ForAll(x => { x.DateTo = dateTo; x.DateFrom =
dateFrom; });
        periodicityData = periodicityResult;
        return;
    }

    DateTime periodStartDate =
periodisityManager.GetPeriodStartDate(dateFrom);
    DateTime nextperiodStartDate =
periodisityManager.GetNextPeriodStartDate(dateFrom);

    //to get boundary dates we get report for
    DateTime modelTypeStartDate = periodStartDate;
    DateTime modelTypeEndDate =
periodisityManager.GetPeriodStartDate(dateTo).AddDays(-1);

    if (nextperiodStartDate >= dateTo)//this means that no one period ends
between selected dates so we don`t check any data
    {
        qualityData = Enumerable.Empty<ReportDataRowDescriptor>();
        periodicityData = Enumerable.Empty<ReportDataRowDescriptor>();
        return;
    }

    models = type.SpatialModels.Where(x=>x.MineElementID ==
currentElement.ID);

    while (nextperiodStartDate <= dateTo)
    {
        SpatialModel_TypeState currentTypeState =
type.GetActualStateForMineElementForDate(currentElement.ID,
nextperiodStartDate.AddDays(-1));
        MineElementState currentElementState =
currentElement.GetActualStateForDate(nextperiodStartDate.AddDays(-1));
        if (currentElementState.IsActive == false ||
currentTypeState.IsActive == false)
        {
            periodStartDate = nextperiodStartDate;
            nextperiodStartDate =
periodisityManager.GetNextPeriodStartDate(periodStartDate);
            continue;
        }

        long mainBrush = currentElementState.Color;
```

```
        IEnumerable<SpatialModel> periodModels = models.Where(m => m.Date
>= periodStartDate && m.Date < nextperiodStartDate).ToArray();
        if(!periodModels.Any())
        {
            periodicityResult.Add( new ReportDataRowDescriptor()
            {
                MineElement =
currentElement.GetStringPathToElement(rootElement),
                ModelType = type.Name,
                Period = periodisityManager.GetPeriodName(periodStartDate),
                ErrorType =
ResourcesManager.Current.Local.GetString("MSRW_noModelsErrorGroup",
AssemblyResourceInfo.Path),
                MainColor = mainBrush,
                Periodisity = periodisityManager.ToString(),
                MineElementID = currentElement.ID
            });
        }
        else
        {
            foreach (SpatialModel model in periodModels)
            {
                ReportDataRowDescriptor error = new
ReportDataRowDescriptor()
                    {
                        MineElement =
currentElement.GetStringPathToElement(rootElement),
                        ModelType = type.Name,
                        Period =
periodisityManager.GetPeriodName(periodStartDate),
                        ErrorType =
ResourcesManager.Current.Local.GetString("MSRW_modelsWithoutFilesErrorGroup
", AssemblyResourceInfo.Path),
                        Model = model.FullModelName,
                        MainColor = mainBrush,
                        Periodisity = periodisityManager.ToString(),
                        MineElementID = currentElement.ID
                    };
                if (!model.SpatialModel_OriginalItems.Any() &&
!model.SpatialModel_ConvertedItems.Any())
                {
                    error.ErrorType =
ResourcesManager.Current.Local.GetString("MSRW_modelsWithoutFilesErrorGroup
", AssemblyResourceInfo.Path);
                    qualityResult.Add(error);
                    continue;
                }

                if (model.AreAllRequiredFilesPresent() == false)
                {
```

```
                    error.ErrorType =
ResourcesManager.Current.Local.GetString("MSRW_modelsWithoutRequiredFilesEr
rorGroup", AssemblyResourceInfo.Path);
                    qualityResult.Add(error);
                    continue;
                }

                if (type.AreConvertedFilesRequired &&
!model.SpatialModel_ConvertedItems.Any())
                {
                    error.ErrorType =
ResourcesManager.Current.Local.GetString("MSRW_modelsWithoutConvertedErrorG
roup", AssemblyResourceInfo.Path);
                    qualityResult.Add(error);
                    continue;
                }
            }
        }
        periodStartDate = nextperiodStartDate;
        nextperiodStartDate =
periodisityManager.GetNextPeriodStartDate(periodStartDate);
    }

    qualityResult.ForAll(x => { x.DateTo = modelTypeEndDate; x.DateFrom =
modelTypeStartDate; });
    qualityData = qualityResult;
    periodicityResult.ForAll(x => { x.DateTo = modelTypeEndDate; x.DateFrom
= modelTypeStartDate; });
    periodicityData = periodicityResult;
}
```

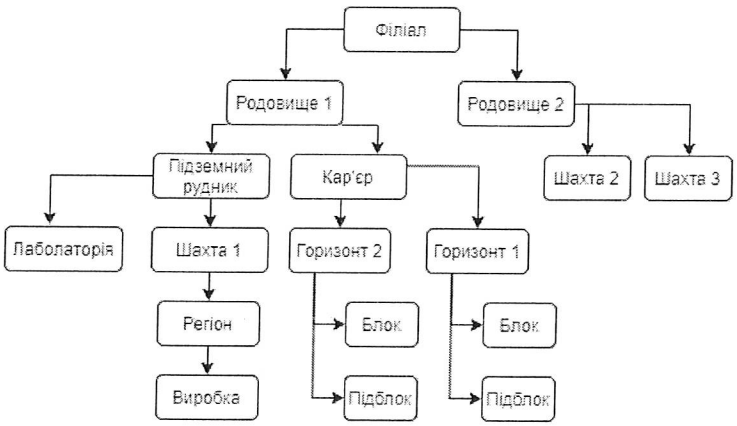
ДОДАТОК Б

ГРАФІЧНІ МАТЕРІАЛИ АТЕСТАЦІЙНОЇ РОБОТИ

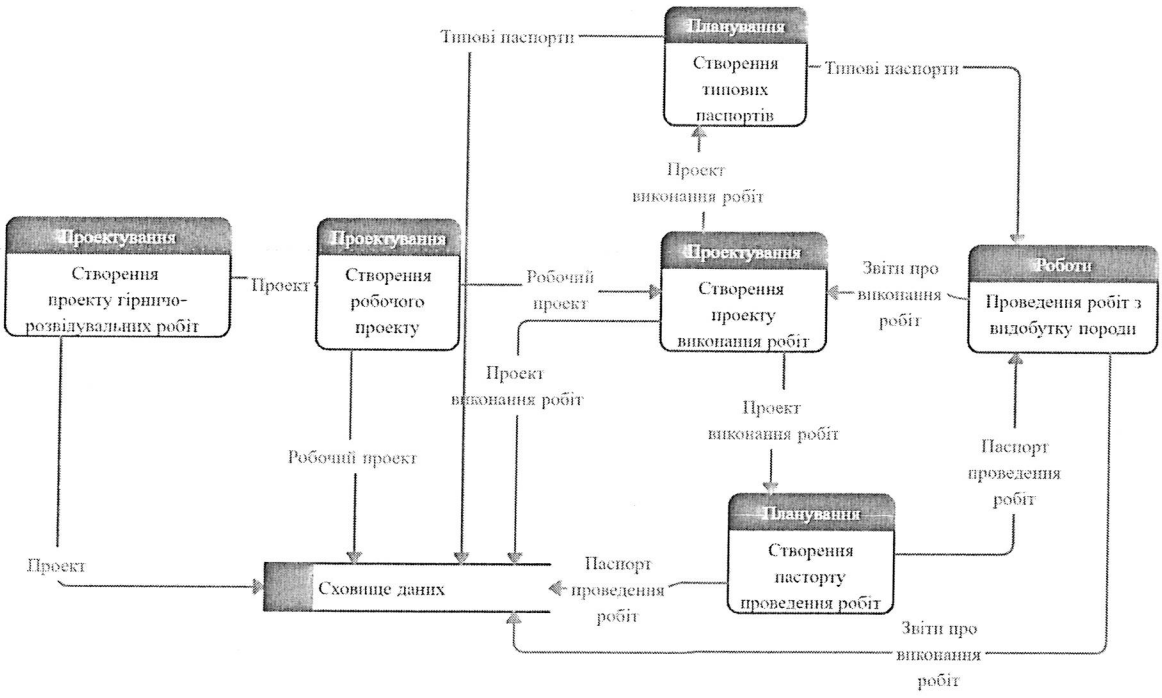
ГЮІК. 502810.0103 С2

(позначення документу)

УЗАГАЛЬНЕНА ІЄРАРХІЧНА СХЕМА ВИРОБНИЧИХ ПІДРОЗДІЛІВ ГІРНИЧОДОБУВНОГО ПІДПРИЄМСТВА



УЗАГАЛЬНЕНА СХЕМА ДОКУМЕНТООБІГУ ГІРНИЧОДОБУВНОГО ПІДПРИЄМСТВА



Розробив	Дмитрієв О.В.		14.12.20	Розробка системи зберігання та аналізу якості звітності про виконання завдань виробничим персоналом	
Перевірив	Мінухін С.В.		14.12.20		
Н. Контр.	Мінухін С.В.		14.12.20		
				СПРм-19-1	Аркуш 1
Затвердив	Гребеннік І.В.			СТ	Аркушів 1

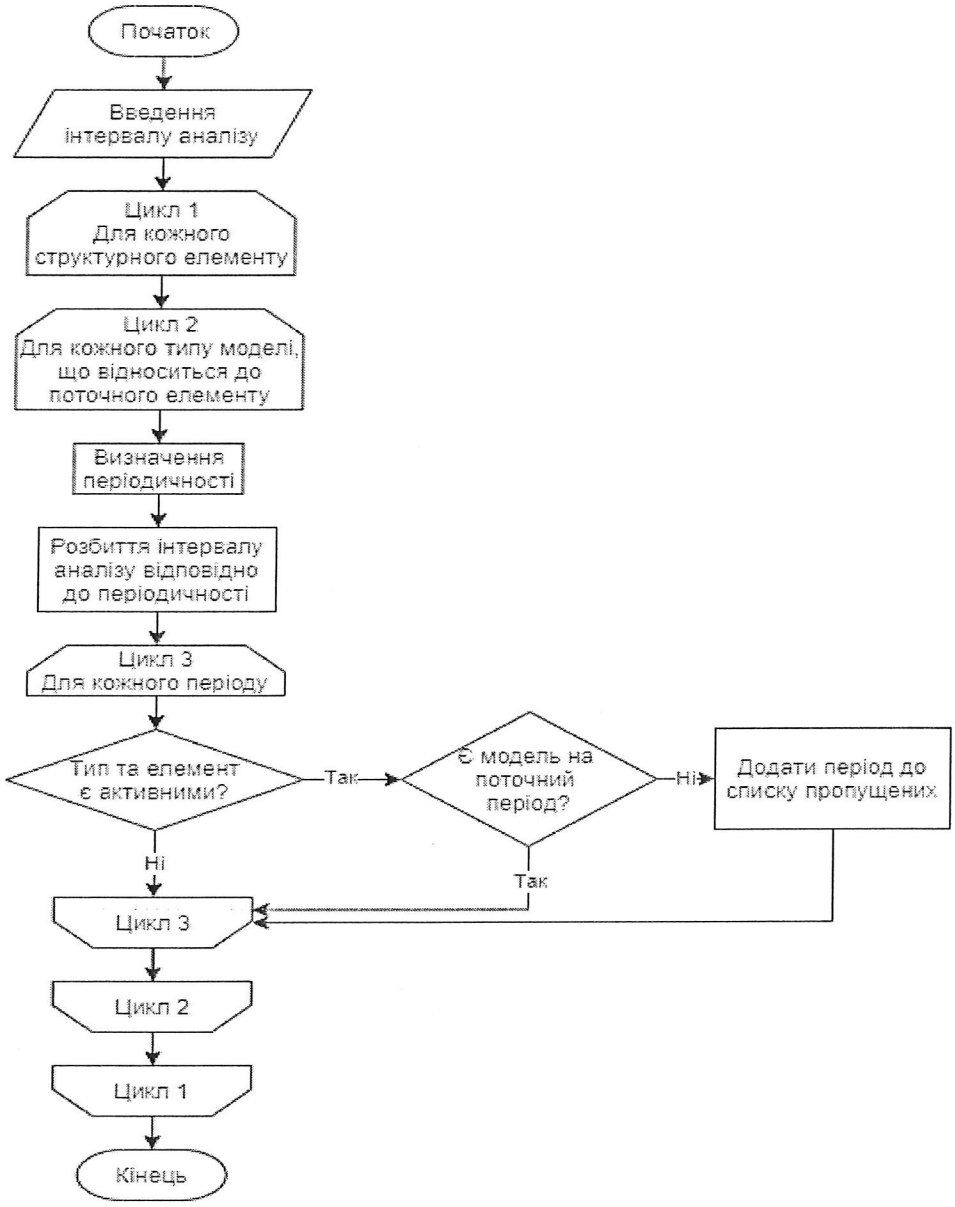
ГЮІК. 502810.0103 С2

СХЕМА АЛГОРИТМУ ПЕРЕВІРКИ ЯКОСТІ ЗВІТІВ ЗА КРИТЕРІЄМ НАЯВНОСТІ ФАЙЛІВ

					ГЮІК. 502810.0103 С2	
					Схема алгоритму перевірки якості звітів за критерієм наявності файлів	Літ. / Маса. / Масштаб
Зм.	Літ.	№ докум.	Підпис	Дата		
Розробив		Дмитрієв О.В.		14.12.20		Аркуш 1 / Аркушів 1
Перевірив		Мінухін С.В.		14.12.20		
Т. Контр.						ХНУРЕ Кафедра СТ
Рецензія		Чугай А. М.				
Н.Контр.		Мінухін С.В.		14.12.20		
Затвердив		Гребеннік І. В.				

ГЮІК. 502810.0103 С2

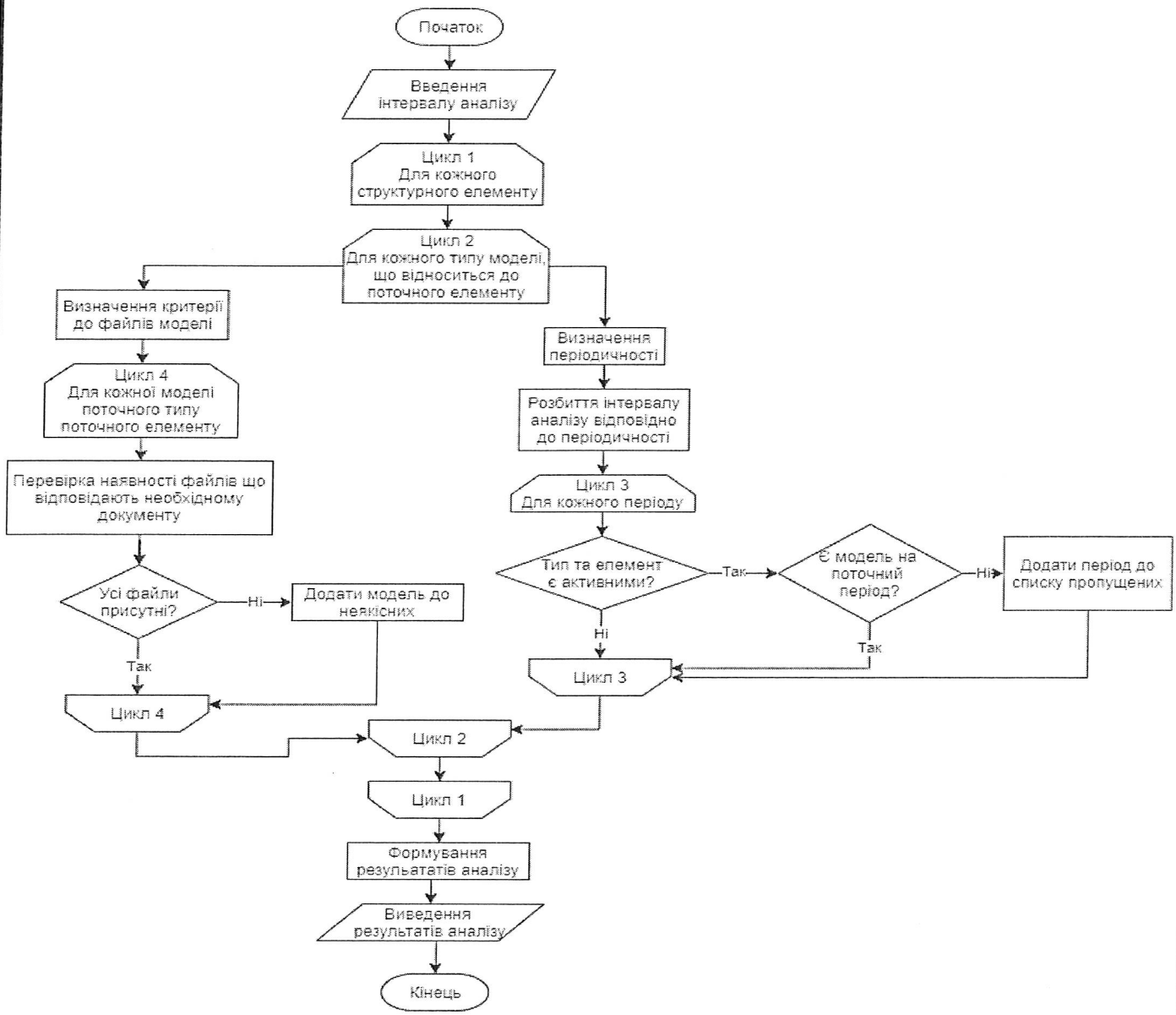
СХЕМА АЛГОРИТМУ ПЕРЕВІРКИ ЯКОСТІ ЗВІТІВ ЗА КРИТЕРІЄМ СВОЄЧАСНОСТІ



					ГЮІК. 502810.0103 С2			
					Схема алгоритму перевірки якості звітів за критерієм своєчасності	Літ.	Маса.	Масштаб
Зм.	Літ.	№ докум.	Підпис	Дата				
Розробив		Дмитрієв О.В.		14.12.20				
Перевірив		Мінухін С.В.		14.12.20		Аркуш 1	Аркушів 1	
Т. Контр.								
Рецензія		Чугай А. М.				ХНУРЕ Кафедра СТ		
Н.Контр.		Мінухін С.В.		14.12.20				
Затвердив		Гребеннік І. В.						

ГЮІК. 502810.0103 С2

СХЕМА ЗАГАЛЬНОГО АЛГОРИТМУ ПЕРЕВІРКИ ЯКОСТІ ЗВІТІВ



Зм.	Літ.	№ докум.	Підпис	Дата
Розробив		Дмитрієв О.В.		14.12.20
Перевірив		Мінухін С.В.		14.12.20
Т. Контр.				
Рецензія		Чугай А. М.		
Н.Контр.		Мінухін С.В.		
Затвердив		Гребеннік І. В.		

ГЮІК. 502810.0103 С2

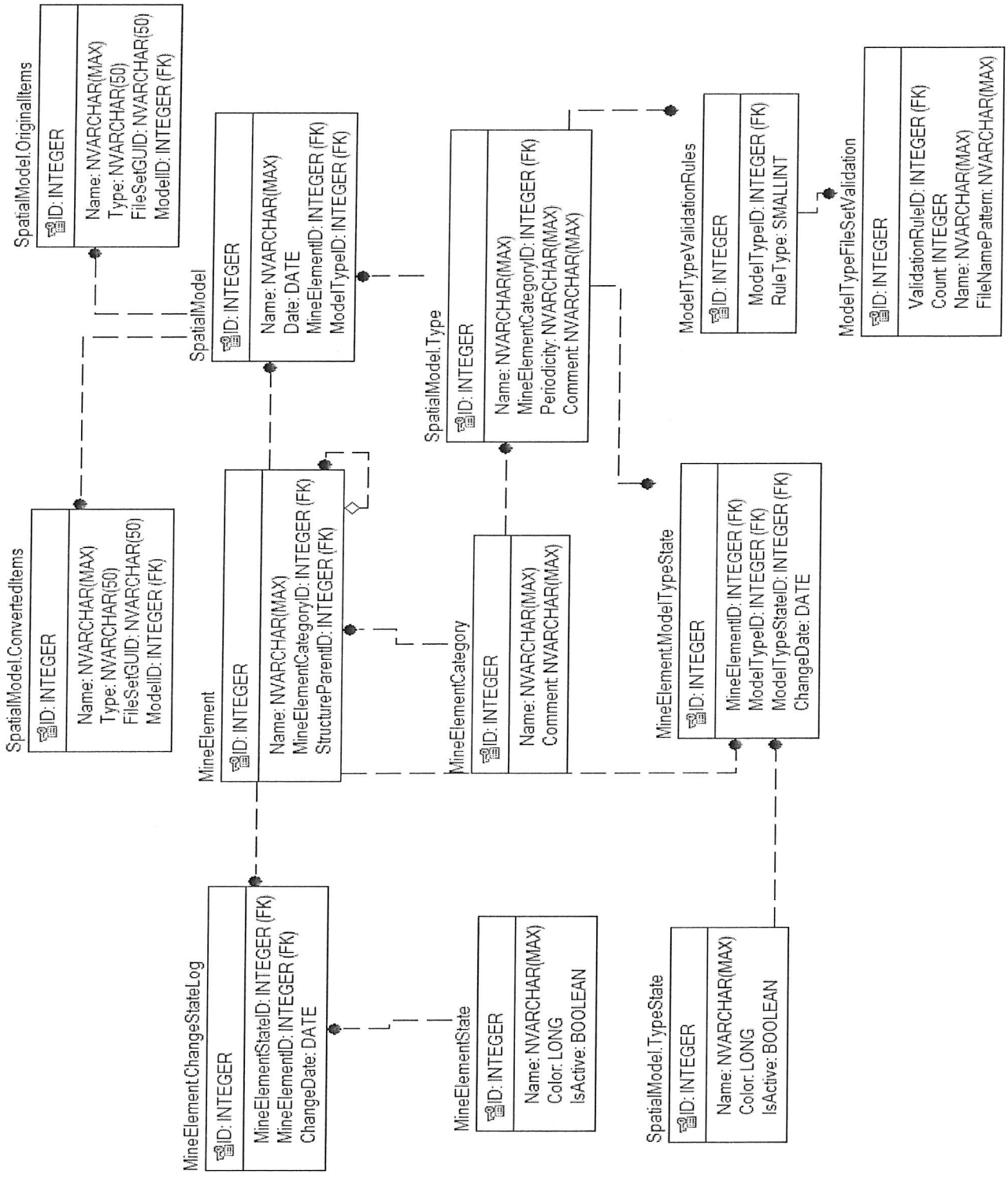
Схема загального алгоритму перевірки якості звітів

Літ.	Маса.	Масштаб

Аркуш 1	Аркушів 1

ХНУРЕ
Кафедра СТ

ЛОГІЧНА МОДЕЛЬ БАЗИ ДАНИХ



Розробив	*Дмитрієв О.В.*		*14.12.20*	*Розробка системи зберігання та аналізу якості звітності про виконання завдань виробничим персоналом*	
Перевірив	*Мінухін С.В.*		*14.12.20*		
Н. Контр.	*Мінухін С.В.*		*14.12.20*		
				СПРм-19-1	*Аркуш 1*
Затвердив	*Гребеннік І.В.*			*СТ*	*Аркушів 1*

ДОДАТОК В

ВІДОМІСТЬ АТЕСТАЦІЙНОЇ РОБОТИ

ГЮІК. 502810.0103 ДЗ

(позначення документа)

№	Позначення	Назва	Дод. відом.
		Текстові документи	
1.	ГЮІК. 502810.0103 ПЗ	Пояснювальна записка	104 арк.
2.	ГЮІК. 502810.0103 01-12-01	Текст програми	9 арк.
		Графічні матеріали	
3.		Узагальнена ієрархічна схема	Включено в ПЗ
		виробничих підрозділів	
		гірничодобувного підприємства	
4.		Узагальнена схема	Включено в ПЗ
		документообігу гірничодобувного	
		підприємства	
5.	ГЮІК.502810.0103 С2	Схема алгоритму перевірки	Включено в ПЗ
		якості звітів за критерієм	
		наявності файлів	
6.	ГЮІК.502810.0103 С2	Схема алгоритму перевірки	Включено в ПЗ
		якості звітів за критерієм	
		своєчасності	
7.	ГЮІК.502810.0103 С2	Схема загального алгоритму	Включено в ПЗ
		перевірки якості звітів	
8.		Логічна модель бази даних	Включено в ПЗ

Изм.	Лист	№ докум	Підпис	Дата	ГЮІК.502810.0103 ДЗ		
Розроб.		Дмитрієв О.В.		14.12.20	Розробка системи зберігання та аналізу якості звітності про виконання завдань виробничим персоналом(на прикладі гірничодобувного підприємства)	Лист	Листів
Перевір.		Мінухін С. В.		14.12.20		1	2
Н. Контр.		Мінухін С.В.		14.12.20	ХНУРЕ Кафедра СТ		
Утверд.		Гребеннік І. В.					

№	Позначення	Назва	Дод. відом.

					Розробка системи зберігання та аналізу якості звітності про виконання завдань виробничим персоналом(на прикладі гірничодобувного підприємства)	*Аркуш*
						2
Зм.	*Лист*	*№ докум.*	*Підпис*	*Дата*		