**НЕГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ ЧАСТНОЕ** **УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ** **«МОСКОВСКИЙ ФИНАНСОВО-ПРОМЫШЛЕННЫЙ УНИВЕРСИТЕТ** **“СИНЕРГИЯ”»**

|  |  |  |
| --- | --- | --- |
| **Факультет/Институт** |  | Кафедра ИМиИКТ им. В.В. Дика |
|  |  | (наименование факультета/ Института) |
| **Направление/специальность** |  | Информационные технологии и программирование |
| **подготовки:** |  | (код и наименование направления /специальности подготовки) |
| **Форма обучения:** |  | Очная |
|  |  | (очная, очно-заочная, заочная) |
|  |  |  |

**Отчет по практической работе №2**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **на тему** |  | **Разработка модульной структуры проекта** | | |
|  |  | (наименование темы) | | |
|  |  |  | | |
| **по дисциплине** | | |  | **Инструментальные средства разработки программного обеспечения** |
|  | | |  | (наименование дисциплины) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Обучающийся** |  | Ткаченко Ричард Николаевич |  |  |
|  |  | (ФИО) |  | (подпись) |
| **Группа** |  | Группа – ДКИП 111 |  |  |
|  |  |  |  |  |

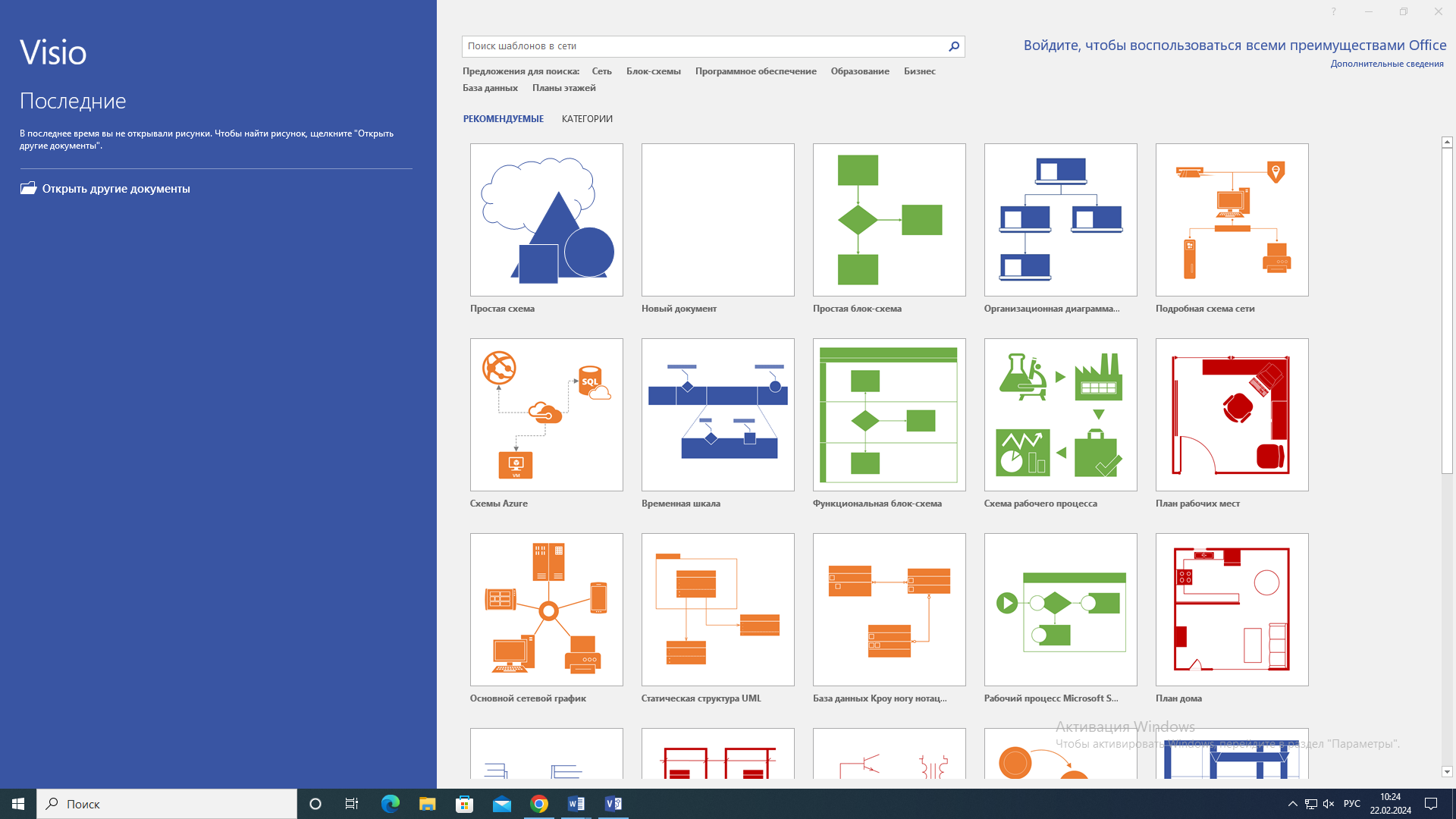
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Преподаватель** |  | **Кондратенко Николай Александрович** |  |  |
|  |  | (ФИО) |  | (подпись) |

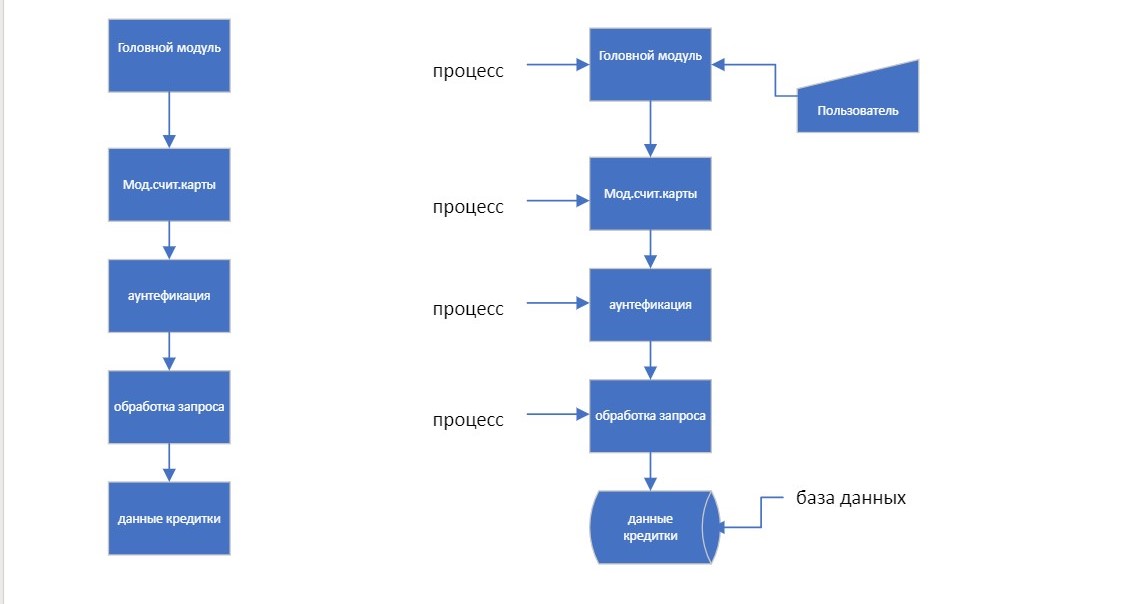
**Москва 2024 г.**

**Цель занятия**: изучение процесса разработки модульной структуры программного обеспечения, осуществляемого с помощью структурных карт Константайна.

Задание

Запустил пакет MS Visio





**Контрольные вопросы**

1. Цель разработки модульной структуры - разбиение программы на отдельные части (модули), которые могут быть разработаны и протестированы независимо друг от друга, а затем объединены в единое целое. Это позволяет повысить эффективность и качество разработки программного обеспечения.
2. Программный модуль - это часть программы, которая выполняет определенную функцию и может быть вызвана из других частей программы. Передача управления между модулями осуществляется с помощью вызова процедур или функций. Организация связи по управлению и по данным между модулями может осуществляться с помощью различных механизмов, таких как передача параметров, использование глобальных переменных или обмен сообщениями.
3. Виды связности модулей:

* Функциональная связность - модули связаны между собой по выполняемым ими функциям.
* Информационная связность - модули связаны между собой по используемым ими данным.
* Коммуникационная связность - модули связаны между собой по обмениваемым ими сообщениям.

1. Виды целостности модулей:

* Логическая целостность - модуль выполняет одну четко определенную функцию.
* Структурная целостность - модуль имеет четко определенную структуру, которая облегчает его понимание и модификацию.
* Функциональная целостность - модуль выполняет свою функцию без ошибок.

1. Типовые модульные структуры:

* Иерархическая структура - модули организованы в виде дерева, где каждый модуль может вызывать только модули, расположенные ниже него в дереве.
* Сетевая структура - модули организованы в виде сети, где каждый модуль может вызывать любой другой модуль.
* Звездная структура - все модули связаны с одним центральным модулем, который выполняет функции управления и координации.

1. Проектирование модульной структуры с помощью структурных карт. Структурные карты - это графические представления модульной структуры программы. Они позволяют визуализировать связи между модулями и оценить их сложность.
2. Физическая модель системы - это модель, которая описывает физическую структуру системы, включая ее компоненты, связи между ними и их свойства.
3. Компонента на диаграмме UML отображается в виде прямоугольника с тремя отсеками:

* Имя компонента
* Список интерфейсов, которые предоставляет компонент
* Список интерфейсов, которые использует компонент

1. Интерфейс - это контракт между компонентами, который определяет набор операций, которые компонент может выполнять.
2. Связи между интерфейсами и компонентами:

* Реализация - компонент реализует интерфейс, если он предоставляет все операции, определенные в интерфейсе.
* Использование - компонент использует интерфейс, если он вызывает операции, определенные в интерфейсе.