

Наш код условно можно поделить на 4 части с соответствующей иерархией:

1. Workspace
Workspace – наша рабочая среда, через которую будет производиться запуск программы, некая точка входа (Она же main)
2. Storage
Storage – отвечает за хранение всех наших данных. С помощью сериализации и десериализации мы сохраняем и обновляем наши данные для соответствующих файлов
3. Users
Users – Именно здесь осуществляется взаимодействие всех пользователей нашего Интранета, каждый класс имеет собственные функции, которые влияют на состояние наших данных (За более подробными данными обращаться к Case Diagram)
4. Course
Course – Описывается состояние текущих предметов, каждый предмет рассматривается как отдельный объект с соответствующими полями

Наша иерархия программы:

- 1) Workspace
 - a) Users - folder
 - i) User
 - (1) Admin
 - (2) Student
 - (3) Employee
 - (a) Teacher
 - (b) Manager
 - (c) TechSupport
 - b) Storage – folder
 - i) Storage
 - c) Course – folder
 - i) Course
 - ii) CourseFiles
 - d) Enums – folder
 - i) AccessRights

Далее изложены цели работы каждого класса

Workspace: Точка входа

Storage: Служит базой данных, реализованной с помощью сериализации/десериализации основанной на файлах.

User: Обобщает всех пользователей нашего Интранета

Employee: Обобщает работников Университета, наследуется от User

Teacher, Manager, TechSupport : Выполняют соответствующие обязанности по работе, наследуются от Employee

Admin: Имеет доступ ко всему функционалу Интранета, наследуется от User.

Student: Роль студента, наследуется от User.

Course: Предмет.

CourseFile: Представляет собой файл с данными по предмету.

AccessRights: Осуществляет режим в котором будет работать User

За более подробной информацией обращаться к UML-диаграмме.