***Наш код условно можно разделить на 4 части с соответствующей иерархией:***

1. Workspace

Workspace – наша рабочая среда, через которую будет производится запуск программы, некая точка входа (Она же main)

1. Storage

Storage – отвечает за хранение всех наших данных. С помощью сериализации и десериализации мы сохраняем и обновляем наши данные для соответствующих файлов, все файлы хранятся в пределах одного пакета, Files.

1. 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)

1. Course

Course – Описывается состояние текущих предметов, каждый предмет рассматривается как отдельный объект с соответствующими полями

***Наша иерархия программы:***

1. Workspace
   1. Users - folder
      1. User
         1. Admin
         2. Student
         3. Employee
            1. Manager

Email ↓

Email↓

* + - * 1. Teacher

Order↑↓

Order↓↑

* + - * 1. TechSupport
  1. Storage – folder
     1. Storage
  2. Course – folder
     1. Course
     2. CourseFiles
  3. Enums – folder
     1. 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 соответсвующие 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

*ChangeFunctions:* Вспомогательный класс, необходимый для оптимизации кода, повышает удобочитаемость. Здесь находятся некоторые функции, необходимые для корректной работы класса Admin.

*Order:* Благодаря запросам, происходит коммуникация между двумя классами, TechSupport и Teacher. Может работать в обе стороны.

*Email:* Электронные письма служат для связи между классами Manager и Teacher. Работает лишь в одну сторону, учитель никак не может отвечать на письма Managerа.

*Driver:* Играет роль прикладного уровня, осуществляя связь между нашей программой и пользователем.

За более подробной информацией обращаться к UML-диаграмме.