

Министерство образования Республики Беларусь
Учреждение образования
«Брестский государственный технический
университет» Кафедра ИИТ

**Отчет по лабораторной работе З
Дисциплина “ПрИС”**

Выполнил:

Студент группы ПО-3

Кабачук Д. С.

Проверил:

Лаврущик А. И.

Брест 2021

Гексагональная архитектура

Вариант 11

Цель работы: познакомиться с гексагональной архитектурой и проектированием систем на её основе.

Задание для выполнения: определите структуру вашего приложения (ЛР 1–2) в соответствии с архитектурным стилем «гексагональная архитектура».

Разработайте файловую структуру для всех уровней, примерно представьте, какие сущности, репозитории, сервисы вам понадобятся, где необходима инверсия зависимостей и т.д.

Конкретную реализацию на каждом уровне вы будете делать на последующих работах.

Предметная область: продажа игрушек.

Используем фреймворк Symphony, пакетный менеджер Composer.

Структура проекта:

```
▼ └── lab1 ~/Рабочий стол/Политех/ПИС/lab1
    ├── bin
    ├── config
    ├── public
    └── src
        ├── Controller
        │   ├── .gitignore
        │   ├── PassengersController.php
        │   ├── ShipsController.php
        │   └── TravelsController.php
        ├── Entity
        │   ├── .gitignore
        │   ├── Passenger.php
        │   ├── Ship.php
        │   └── Travel.php
        ├── Form
        ├── Migrations
        └── Repository
            ├── .gitignore
            └── PassengerRepository.php
```

```
    ⓒ ShipRepository.php
    ⓒ TravelRepository.php
    ⓒ Kernel.php
▼ └── templates
    └── passenger
        └── passenger_all.html.twig
        └── passenger_form.html.twig
    └── ship
        └── ship_all.html.twig
        └── ship_form.html.twig
    └── travel
        └── base.html.twig
    └── var
    └── vendor
        └── .env
        └── .gitignore
        └── composer.json
        └── composer.lock
        └── symfony.lock
    └── External Libraries
```

Опишем важные субдиректории:

1. Config — конфигурация проекта которую можно подставлять в зависимости от изменения среды.
2. Migrations — упорядоченный набор миграций (модификаций структуры) БД.
3. Controller — слой представления, организует маппинг от запроса на ответ к нему.
4. Entity — модели данных, соответствующие хранимым в БД.
5. Repository — слой для работы с внешними данными, такими как содержимое локальной БД, либо внешний сервис.
6. Form — веб-формы для entity, позволяющие быстро создавать сущности и

заполнять их предопределенными данными.

7. Templates — шаблоны для формирования графического интерфейса для пользователя.
8. Файл .env — переменные окружения определяющие ключевые аспекты
9. Файл composer.json — зависимости пакетного менеджера composer.

Опишем основные сущности:

Контроллеры:

1. PassengersController.php – контроллер для работы с запросами контроля пассажиров.
2. ShipsController.php – контроллер для работы с запросами контроля кораблей.
3. TravelsController.php – контроллер для работы с запросами контроля путешествий.

Модели:

1. Passenger – модель пассажира.
2. Ship – модель корабля.
3. Travel – модель путешествия.

Репозитории:

1. PassengerRepository – репозиторий доступа к данным пассажира.
2. ShipRepository – репозиторий доступа к данным корабля.
3. TravelRepository – репозиторий доступа к данным путешествия.

4. Выводы

В данной лабораторной работе я познакомился с гексагональной архитектурой и проектированием систем на её основе.