

Министерство образования Республики Беларусь
Учреждение образования
“Брестский государственный технический университет”
Кафедра ИИТ

**Отчёт
По лабораторной работе №3
По дисциплине ПрИС**

Выполнил
Студент группы ПО-3
3-го курса
Кулинкович И. Т.

Проверил
Лаврущук А. И.

Лабораторная работа №3

Гексагональная архитектура

ВАРИАНТ 17

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. Управление фитнес-центром.

Ход работы

Используем фреймворк Symphony, пакетный менеджер Composer, утилиту автогенерации symfony/maker-bundle. Преимуществом symphony является предоставление необходимой нам архитектуры из коробки.

Организуем проект с использованием symphony. Приложение разделено на следующие субдиректории:

```
└── config
    ├── packages
    ├── routes
    └── bundles.php
    └── preload.php
    └── routes.yaml
    └── services.yaml
    > migrations
    > public
    < src
        < Controller
            .gitignore
            BookingController.php
            TrainingController.php
        < Entity
            .gitignore
            Trainer.php
            TrainingSession.php
            Visitor.php
        < Form
            BookingType.php
            TrainingType.php
        < Repository
            .gitignore
            TrainerRepository.php
            TrainingSessionRepository.php
            VisitorRepository.php
        Kernel.php
    < templates
        < booking
            index.html.twig
        < training
            index.html.twig
            base.html.twig
        > tests
        > translations
```

Опишем важные субдиректории:

1. Controller — слой представления, организует маппинг от запроса на ответ к нему.
2. Migrations — упорядоченный набор миграций (модификаций структуры) БД.
3. Config — конфигурация проекта которую можно подставлять в зависимости от изменения среды.
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. Tests — автоматизированные тесты.
9. Translations — файлы для локализации приложения под различные языки.
10. Файл .env — переменные окружения определяющие ключевые аспекты работы приложения.
11. Файл composer.json — зависимости пакетного менеджера composer.

Опишем основные сущности, которые создадим:

1. BookingController — контроллер для flow записи клиента.
2. TrainingController — контроллер для проведения тренировки.
1. Trainer — модель тренера.
2. Visitor — модель посетителя фитнес-центра.
3. TrainingSession — модель определенной во времени тренировки.
1. BookingType — форма быстрой записи клиента.
2. TrainingType — форма быстрого создания тренировки.
1. TrainerRepository — репозиторий доступа к данным тренеров.
2. TrainingSessionRepository — репозиторий доступа к данным тренировок.
3. VisitorRepository — репозиторий доступа к данным посетителей.

Вывод

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