

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

Отчет по лабораторной работе 5
Дисциплина “ПриС”

Выполнил:

Студент группы ПО-3

Кабачук Д. С.

Проверил:

Лаврущик А. И.

Брест 2021

Доменная модель

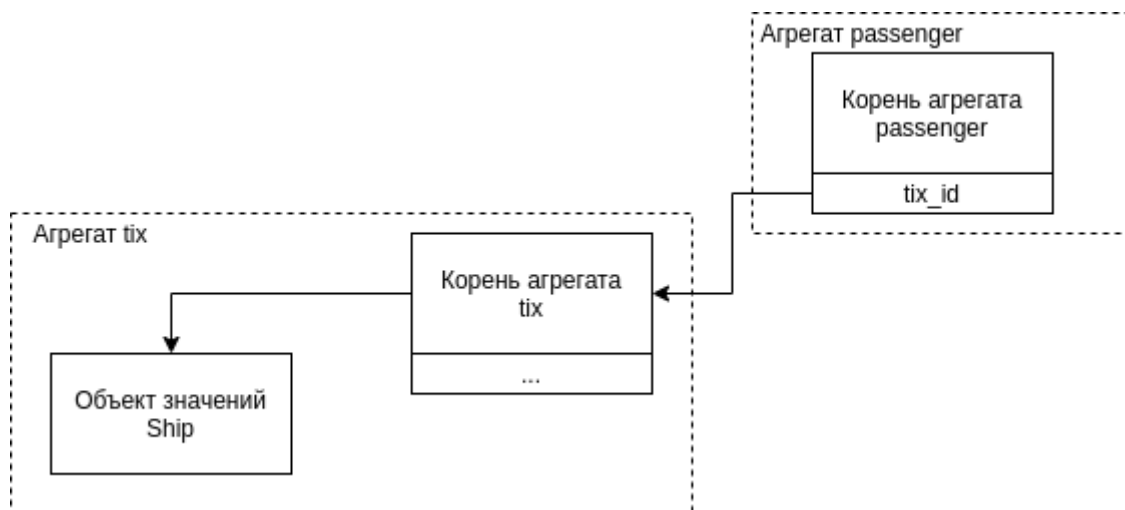
Вариант 11

1. **Цель работы:** познакомиться с тактическими шаблонами предметно-ориентированного проектирования.

Задание для выполнения: Реализуйте не менее двух агрегатов, сущностей, объектов значений домена Вашей гексагональной архитектуры из предыдущей работы, а также репозитории и не менее двух доменных событий для агрегатов. Один агрегат должен включать в себя сущность-корень агрегата, список дочерних сущностей, каждая сущность содержать наряду с обычными свойствами ещё и объекты-значения. Идентификаторы сущностей должны генерироваться: для корней агрегатов – репозиторием, для внутренних сущностей – самим корнем агрегата.

Предметная область: космический туризм.

3. Ход работы



Tix:

```
<?php
```

```
namespace App\Entity;
```

```
use Doctrine\ORM\Mapping as ORM;
```

```
/**
```

```
 * @ORM\Entity(repositoryClass="App\Repository\TravelRepository")
```

```
 */
```

```

class Travel
{
    /**
     * @ORM\Id()
     * @ORM\GeneratedValue()
     * @ORM\Column(type="integer")
     */
    private $id;

    /**
     * @ORM\Column(type="string", length=100)
     */
    private $where_from;

    /**
     * @ORM\Column(type="string", length=100)
     */
    private $where_to;

    /**
     * @ORM\Column(type="integer")
     */
    private $price;

    /**
     * @ORM\ManyToOne(targetEntity="App\Entity\Ship")

```

```

    * @ORM\JoinColumn(nullable=false)

    */

private $ship;

/**
    * @ORM\Column(type="integer")
    */

private $ship_id;

public function getId(): ?int
{
    return $this->id;
}

public function getWhereFrom(): ?string
{
    return $this->where_from;
}

public function setWhereFrom(string $where_from): self
{
    $this->where_from = $where_from;

    return $this;
}

```

```
public function getWhereTo(): ?string
{
    return $this->where_to;
}
```

```
public function setWhereTo(string $where_to): self
{
    $this->where_to = $where_to;

    return $this;
}
```

```
public function getPrice(): ?int
{
    return $this->price;
}
```

```
public function setPrice(int $price): self
{
    $this->price = $price;

    return $this;
}
```

```
public function getShip(): ?Ship
{

```

```
        return $this->ship;
    }

    public function setShip(?Ship $ship): self
    {
        $this->ship = $ship;

        return $this;
    }

    public function getShipId(): ?int
    {
        return $this->ship_id;
    }

    public function setShipId(int $ship_id): self
    {
        $this->ship_id = $ship_id;

        return $this;
    }
}
```

Passenger:

<?php

```

namespace App\Entity;

use Doctrine\ORM\Mapping as ORM;

/**
 *
 * @ORM\Entity(repositoryClass="App\Repository\PassengerRepository")
 */
class Passenger
{
    /**
     * @ORM\Id()
     * @ORM\GeneratedValue()
     * @ORM\Column(type="integer")
     */
    private $id;

    /**
     * @ORM\Column(type="string", length=50)
     */
    private $last_name;

    /**
     * @ORM\Column(type="integer")
     */
    private $tix;

```

```
public function getId(): ?int
```

```
{
```

```
    return $this->id;
```

```
}
```

```
public function getLastName(): ?string
```

```
{
```

```
    return $this->last_name;
```

```
}
```

```
public function setLastName(string $last_name): self
```

```
{
```

```
    $this->last_name = $last_name;
```

```
    return $this;
```

```
}
```

```
public function getTix(): ?int
```

```
{
```

```
    return $this->tix;
```

```
}
```

```
public function setTix(int $tix): self
```

```
{
```

```
    $this->tix = $tix;
```



```
        return $this;
    }
}
```

4. Выводы: в данной лабораторной работе я познакомился с тактическими шаблонами предметно-ориентированного проектирования.